https://doi.org/10.52326/ic-ecco.2021/SEC.02

Task Completion Time Evaluation in Moving Target Defense Cloud Environment Based on Matrix SRN With Z-Fuzzy Parameters

Alexei Sclifos [1], Emilia Sclifos [1], Emilian Guțuleac [1]
[1] Technical University of Moldova, Bd. Ștefan cel Mare, 168, MD-2004, Chișinău, R. Moldova,
emilian.gutuleac@calc.utm.md, https://utm.md

***Abstract*—Moving Target Defense (MTD) has recently emerged as a game-changing technique in confronting cyber attacks and defending cloud computing systems (CCS) and computer networks (CN). A MTD technique randomly modifies the configurations of the attacked CCS, thus creating great uncertainties of the attack surface against cyber-intruders. In this paper, is proposed an analytical modeling approach based on the Matrix Stochastic Reward Nets (MSRN) with fuzzy parameters, that allows attacker's behaviors compact modeling and analysis of the impact in which the use of a MTD technique influences the completion processing time of a running task by CCS. A numerical case study is presented to analyze the impact of different parameters on the expected task completion time and other evaluation metrics.**

Keywords — moving target defense; fuzzy parameters; matrix stochastic reward net; task completion time

I. INTRODUCTION

Traditional computer networks (CN) and cloud computing systems (CCS) defense techniques such as IDS, firewalls, and antiviruses are increasingly ineffective in resisting new types of attacks due to the static properties of protected attack targets [1]. Intruders always have an advantage over time and can get enough information about the vulnerabilities of the CN's or CCS target system before launching an attack. To eliminate this disadvantage, the Moving Target Defense (MTD) approach [1] has emerged as a game changer for these interactions, as it *provides a proactive defense strategy* by creating asymmetric uncertainties in favor of defenders, constantly and variably changing CN configurations.

Designing CN and CCS security systems based on MTD techniques is a complex task, in which a large number of parameters must be taken into account. The influence of these parameters is often mutually opposite, often uncertain and weakly predictable. Depending on the content of the research associated with the use of MTD techniques, they focused on one or more aspects of modeling, determining the parameters and efficiency of the CN and CCS defense systems, using different mathematical formalisms, which reflect their various behavioral aspects: continuous time Markov chains (CTMC), mathematical game theory and Stochastic Reward Nets (SRNs) [2, 3].

Existing research on the evaluation of MTD techniques focuses on the analysis of the effectiveness of MTD to assess the level of security or loss of performance of the system caused by the use of MTD. To our knowledge, only in the paper [4], based on SRN models, the impact of using the MTD technique on the damage of the task processing time on such a protected system is investigated. Nevertheless, in this type of models, the fuzzy epistemic uncertainties of the attacker's and defense's behavior are not taken into account. Also, it is easy to confirm from experience that SRNs [4] are often difficult to use in practice due to the problem of rapidly increasing graphical size of the model to describe the behavior of a real system. In this context, it is necessary to enhance the SRN formalism in order to fully represent more compactly and flexibly the models that describe complex processes in CN and CCS.

Here we present only some content of SRNs that is necessary to describe our quantitative analytical modeling approach based on the Matrix SRNs (MSRN) [5] with Z-fuzzy parameters, called FMSRN. This is needed to analyze how the use of a defense MTD technique affects the task completion processing time in a CCS, considering the real execution time of the entire task becomes longer than needed. More details on MSRNs can be found in [5]. The FMSRN model, presented in this paper, describes both the processing of each task on several VMs, and the behavior of intruders in a CCS defended using MTD techniques. Also, the results presented in this paper can help defenders to choose a better MTD configuration and defense strategy to

https://doi.org/10.52326/ic-ecco.2021/SEC.02

complete the execution of the paper as soon as possible and evaluate some quantitative QoS (Quality of Service) metrics [2, 4,5].

II. MATRIX SRNS WITH Z-FUZZY PARAMETERS

A. *Elements of Z-Fuzzy Numbers*

In [6], L. Zadeh proposed the concept of Z - numbers, which also allows us to take into account the inaccuracy of our knowledge of the membership function using a joint approach from the standpoint of probability theory and theory of possibility. The Z-numbers have more capability to describe the uncertain information. A fuzzy Z -number is an ordered pair of fuzzy numbers, denoted as $Z=(\widetilde{A},\widetilde{R})$. The first component $\widetilde{A}$, plays the role of a fuzzy restriction on the values, is a real-valued uncertain variable X . The second component $\widetilde{R}$ is a measure of reliability for the first component. Computing with Z-numbers can be realized by directly using Zadeh expansion principle, which requires very cumbersome calculations and is extremely difficult when solving complex applied problems. In [7], it was proposed a method of converting Z-numbers *to generalized fuzzy numbers*. Assume $Z=(\widetilde{A},\widetilde{R})$ is a Z-number. The left of Z is the part of restriction, and the right of Z is the part of reliability. Let $\widetilde{A}=\{\langle x,\mu_{\widetilde{A}}(x)\rangle \mid x\in[0,1]\}$ and $\widetilde{R}=\{\langle x,\mu_{\widetilde{R}}(x)\rangle \mid x\in[0,1]\}$, where $\mu_{\widetilde{A}}(x)$ is a *trapezoidal* membership function and $\mu_{\widetilde{R}}(x)$ is a *triangular* membership function. In fact, the conversion of a $Z=(\widetilde{A},\widetilde{R})$ into a *regular* fuzzy number $\widetilde{Z}'$ is performed as follows:

(1) Convert the $\widetilde{R}$ into a crisp number $\delta=(\int x\cdot\mu_{\widetilde{R}}(x)dx)/(\int\mu_{\widetilde{R}}(x)dx)$; (2) Add the weight δ of the $\widetilde{R}$ to $\widetilde{A}$. The weighted Z-number can be denoted as $\widetilde{Z}^{\delta}=\{\langle x,\mu_{\widetilde{A}^{\delta}}(x)\rangle \mid \mu_{\widetilde{A}^{\delta}}(x)=\delta\cdot\mu_{\widetilde{A}}(x),\ x\in[0,1]\}$;

(3) Convert the *irregular* fuzzy number $\widetilde{Z}^{\alpha}$ (weighted restriction) to *regular* fuzzy number :

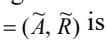
$$\widetilde{Z}'=\{\langle x,\mu_{\widetilde{Z}'}(x)\rangle \mid \mu_{\widetilde{Z}'}(x)=\mu_{\widetilde{A}}(x/\sqrt{\delta}),\ x\in[0,1]\}.$$

Example. Let $\widetilde{A}=(a_1,a_2,a_3,a_4;1)$ be the trapezoidal regular fuzzy number (TrFN) and δ be the weight of $\widetilde{R}$ in $Z=(\widetilde{A},\widetilde{R})$, then $\widetilde{Z}^{\alpha}=(a_1,a_2,a_3,a_4;\alpha)$ and $\widetilde{Z}'=(a_1\cdot\sqrt{\delta},\ a_2\cdot\sqrt{\delta},\ a_3\cdot\sqrt{\delta},\ a_4\cdot\sqrt{\delta};1)$ is TrFN.

According to [8], the average credibility value $\bar{z}'=E[z']$ of TrFN is determined by the expression $\bar{z}'=E[z']=(\sqrt{\alpha})\cdot(a_1+a_2+a_3+a_4)/4$, will further be used to determine the *credible parameters* of a FMSRN model.

B. *Definition* of MSRN *with Z-fuzzy parameters*

The definition and behavior of MSRN (SRN with matrix attributes) is presented in [5]. The MSRN inherits most of the SRN [3] characteristics. In a MSRN model, the matrix attributes of objects (arcs, places capacities, guard functions and transition priorities, rewriting rules, firing rates of transitions, etc.) of the specified type z , depending on the current state of the network, are defined by a set of matrix $A^z=[a_{ij}^z(s)]_{k\times n}\in\mathbf{A}$. The values of the network attributes may be constant, variable or functions of the specified type and may depend on the current state of the MSRN model. The location of the current element $a_{ij}^z(s)$ of the matrix A^z is specified by a set $P_A^z\subset P$ of net *control places*. For example, for the selection of current elements position in A^z , two control places should be specified. Therefore, the current number of tokens $i=m_l=M(p_l)$ and $j=m_v=M(p_v)$ in control places p_l and p_v shows the position of the respective element of matrix A^z , and its values must be imported and taken into account when executing and analyzing the model. Furthermore, should be specified the capacity of the control places $p_l\in P_A^z$ and place $p_v\in P_A^z$ respectively.

Graphically, a matrix attribute of MSRN will be presented in a way that it will contain the matrix name in square brackets. So, for example, a direct arc matrix cardinality, denoted by  , can take values that are contained in a specified matrix **A**.

In order to model more realistically the uncertainty of the attackers' behavior and the defense reaction of the security system of CCS, it is necessary to consider the fuzzy aspects in MSRN.

Thus, the FMSRN, denoted $FM\Gamma$, is specified as a 3-tuple system so that $FM\Gamma=<MSRN,\ \widetilde{\Lambda},\ \widetilde{W}>$, where: $MSRN$ is the underlying of $FM\Gamma$; $\widetilde{\Lambda}: E_\tau\times IN_+^{|P|}\to IR^+$ is the matrix function that determines the Z-fuzzy firing rate $0<\widetilde{\lambda}(t_j,M)<+\infty$ (the parameters of exponential-negative law) of timed transition t_j , that is enabled by current marking M; $\widetilde{\omega}: E_0\times IN_+^{|P|}\to IR^+$ is the matrix Z-fuzzy weight function $0\le\widetilde{\omega}(e,M)<+\infty$ which determines the firing probability $q(t_l,M)$ of immediate transition t_l , enabled by current marking M, that describes a probabilistic selector.

III. INFORMAL DESCRIPTION OF ATTACKED SYSTEM

We consider a scenario in which it is necessary for a critical task to be processed in a certain limited time τ_{Ask} on a CCS [4]. However, given the existence of attacks, the actual time τ_R taken to complete the processing of a task is longer than necessary. To eliminate security concerns and complete the processing of each task as soon as possible, a MTD technique is adopted to protect the processing of this task. As an example of MTD, we consider techniques based on the use of a dynamic computing platform (DCP) on the cloud computing, which allows, by dynamically changing the structure of

https://doi.org/10.52326/ic-ecco.2021/SEC.02

this platform, to considerably complicate the success of attacks [1, 2, 3].

The considered DCP consists of γ heterogeneous virtual machines (VMs) that can have the same or different system hardware / software environments. Each VM has some vulnerabilities and they can be exploited by intruders. To withstand attacks, the processing of each task is divided into L stages. In each stage, MTD randomly chooses a VM to perform (continue) the task processing. There are only two cases where a new VM needs to be elected. One is that the task at the current stage has been successfully processed. The other is that task processing is interrupted due to attacks at the current stage. In both cases the task will not be processed on the same VM if the next stage starts or the current stage will be reprocessed.

We consider a single determined and experienced attacker whose goal is to prevent the normal processing of the task in this CCS. Only one VM of the DCP can be used to process the task at a current stage. We assume that the attacker, before successfully attacking a VM and achieving the desired effect, cannot distinguish the process whether or not this VM is processing the target task during the attack. At the same time, we assume that the attacker can exploit all CCS vulnerabilities in order to have access to all VMs, but he can attack only one VM at a time. Based on these assumptions, typical attack behaviors can be summarized by the following steps. First, the attacker randomly selects a VM to attack. After a successful attack, the attacker gains access to a VM and he can determine if the target task is being processed by that VM. If so, the attacker will destroy the processing of this task and immediately reselect the next target VM. If not, the intruder waits a long time to determine if a task will arrive on this VM. This waiting time has an upper limit, noted $\bar{\tau}_{\max}$. As soon as this limit is reached, the attacker will abandon the current VM and return to selecting the next VM to be attacked. Regardless of how the VM is selected by the task, the process of choosing the target VM by the attacker is the one without memory. In the next attack stage, the same VM can be selected.

To form the modeling of the behavior of this attacked system by FMSRN, we adopt the following hypotheses: (i) the durations (delay) of the activities of the attacker's behavior and those of the task migration on DCP are exponentially distributed random variables; (ii) the correct processing time of the task at each phase by a VM, being a non-exponentially distributed random variable (eg deterministic), is approximated by the Cox-2 distribution [9]. This means that we have the values of the mathematical average $\bar{u}$ and that of the coefficient of variation K^v, $0.5 \le K^v < +\infty$ of the non-exponentially distributed random variable. Based on these two values, the parameters of the Cox-2 distribution are identified as follows [9]: 1) if $K^v < 1$ then $\lambda' = 1/(\bar{u} \cdot K^v)$, $\lambda'' = 2/\bar{u}$ and $q = 2(1 - K^v)$; 2) if $K^v > 1$ then $\lambda' = 2/\bar{u}$, $\lambda'' = 1/(\bar{u} \cdot K^v)$ and $q = 1/(2 \cdot K^v)$. Here λ' (resp. λ'') is the processing rate at the first (resp. at the second) phase, and q (resp. $1-q$) is the probability that after the end of the first phase the second phase will performe (resp. not perform) the task.

IV. FMSRN Model of Task Processing in MTD Environment

In this section, we present the developed FMSRN model, which formally describes the processing of an attacked task in CCS, presented in the previous section. The places (resp. transitions) correspond to the local states (resp. activities) of the attacker and of the MTD security system.

A. Attack Sub-model of DCP

In Fig. 1 is presented the FMSRN1 sub-model, noted $FM\Gamma 1$, *which describes* the behavior of the attacker, who repeatedly randomly selects a VM to attack.

The meanings of places and transitions in attack $FM\Gamma 1$ *sub-model*:

- **Places**. $pa1$ - the initial marking $M_0(pa1)$ describes the potential number of VMs in DCP; $pa2$ - the control place indicating the number, $j = M(pa2)$, of the VM_j that is or can be selected to be attacked; $pa3$ - passive attacker; $pa4$ - intruder started to select target VM; $pa5$ - the target VM_j is attacked; $pa6$ - the target VM_j has successfully attacked;
- ***Timed transitions***. $ta1$, $(ta2)$ - incrementing (decrementing) the target number of VM_j to be attacked; $ta3$ - attacking the CCS; $ta5$ - successfully attacking the target VM_j; $ta8$ -indicates the maximum waiting time for the arrival of the task on the attacked VM;

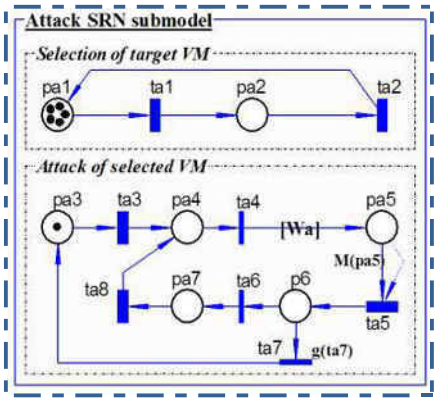


Figure 1. Attack $FM\Gamma 1$ sub-model of DCP

https://doi.org/10.52326/ic-ecco.2021/SEC.02

- **Immediate transitions.** $ta4$ - target selection; $ta6$ - switching to waiting for the arrival of the task on the attacked VM; $ta7$ - failure destruction of the current task processing process, complete the progress of the attack.

The weight $W(ta4, pa5) = [\mathbf{Wa}]$ of the arc $(ta4, pa5)$ is a line vector **Wa** whose components are the serial number $j = 1 + M(pa2)$ of selected target VM_j, i.e. **Wa**=[1, 2, ⋯, N–1, N], and the selection of the coordinate of an element Wa_n of this vector depends on $n = M(pa2) = 0, 1, \ldots, K(pa2)$. Here $K(pa2) = \gamma - 1$ is the capacity of the place $pa2$ and $2 \le \gamma \le N$.

The guard functions $g(ta6) := (M(pm1) \neq N)$ and $g(ta7) := (M(pm2) = 1)$ of attack $FM\Gamma1$ sub-model are evaluated based on the current marking of the FMSRN2 sub-model, noted $FM\Gamma2$ (see Fig. 2), which describes the processing and migration of a task among VMs on the DCP.

The "*Selection of target VM*" subnet of the $FM\,\Gamma1$ sub-model (consisting of $pa1$, $pa2$ places, and transitions $ta1$, $ta2$) models the randomly attack selection of a VM_j. If the firing rates $\lambda_{ta1} = \lambda_{ta2}$ of these transitions are equal to each other, the probability of attack of each VM is the same: $q_{VM_j}^{attack} = 1/\gamma$, where $\gamma = K(pa2)$.

B. Task migration sub-model in DCP

The $FM\Gamma2$ sub-model, shown in Fig. 2, describes the processing and migration behaviors of a task between heterogeneous VMs of DCP.

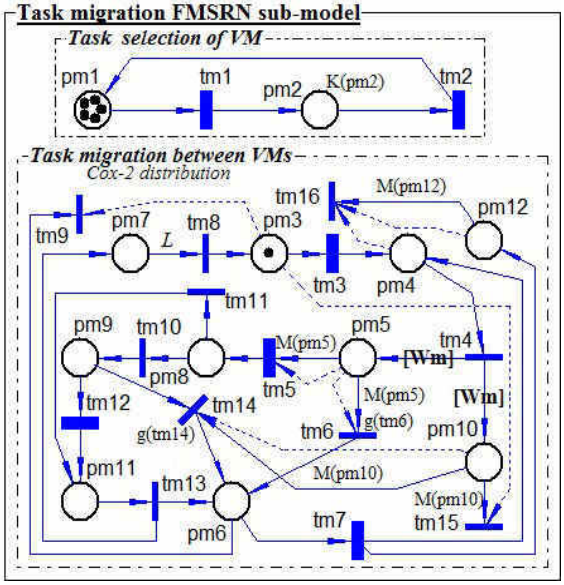


Figure 2. $FM\Gamma2$ sub-model of task migration in DCP

The meanings of places and transitions in task migration $FM\Gamma2$ *sub-model*:

- **Places**. $pm1$ - the $M_0(pm1)$ initial marking exhibit the potential number of VMs in the DCP; $pm2$ - the control location indicating the number $j = M(pa2)$, of VM_j, which is selected to migrate the task and process a respective phase; $pm3$ - the task is ready to be processed on the DCP; $pm4$ - task started to select one VM; $pm5$ - setting the processing of a stage of the task by VM_j; $pm6$ - wait migration task to another VM; $pm7$ - the current marking shows the number of task stages successfully processed by VMs in DCP, $M(pm7) \le L$, L -numbers of stages; $pm8$ - selecting phase 2 or finishing processing at phase 1 of the task according to the Cox-2 distribution by VM_j; $pm9$ - processing the task at stage 2 by VM_j; $pm10$ - the current serial number $j = M(pm10)$ of VM_j which the current stage of the task is processing; $pm11$ - task processing completion time indicator according to Cox-2 distribution; $pm12$ - counter the actual number of migrations.
- **Timed transitions**. $tm1$, ($tm2$) - incrementing (decrementing) the target number of VM_j to be attacked; $tm3$ - arrival of the task processing request on DCP; $tm5$ - task processing time at phase 1 of the Cox-2 distribution; $tm7$ - the delay of the task migration to another VM; $tm12$ - task processing time at phase 2 of the Cox-2 distribution.
- **Immediate transitions.** $tm4$ - the selection of VM_j on which the task is processed; $tm6, tm14$ - disables the current task processing, if VM_j is attacked, respectively; $tm8$ - DCP reset; $tm9$ and $tm15$ - the current task has completed the migration and processing process on DCP; $tm10$ - moving to phase 2 of the Cox-2 distribution processing task; $tm11$ and $tm13$ - successfully complete the processing of a task stage on VM_j; $tm16$ - resetting of $pm12$.

The weights $W(tm4, pm5) = W(tm4, pm10) = [\mathbf{W}m]$ of the arcs $(tm4, pm5)$ and $(tm4, pm10)$ are respectively rendered by a line vector $\mathbf{W}m$, whose components determine the serial number $j = 1 + M(pm2)$ of the current VM_j on which the task is processed, i.e. $\mathbf{W}m$=[1, 2, ⋯, N–1, N], and the selection of the coordinate of an element Wm_k of this vector depends on the current marking of place $pm2$, i.e. $k = M(pm2) =$

https://doi.org/10.52326/ic-ecco.2021/SEC.02

$0, 1, \ldots, K(pm2)$. Here $K(pm2) = K(pa2) = \gamma - 1$ is the capacity of the place $pm2$.

The guard functions $g(tm6) := (M(pm5) = M(pa5))$ and $g(tm14) := (M(pm9) = M(pa5))$ of the $FM\Gamma 2$ sub-model (see Fig.2), describing the processing and migration of a task among VMs, are evaluated based on the current marking of the $FM\Gamma 1$ shown in Fig. 1.

The "*Task migration between VMs*" subnet of the $FM\Gamma 2$ models the probabilistic selector of task migration between VMs, and its meaning is similar to that of the submodel $FM\Gamma 1$. The difference between them is that $\lambda_{ta1} = \lambda_{ta2} \neq \lambda_{tm1} = \lambda_{tm2}$. Also, the probability of task migration between VMs is the same: $q_{VM_j}^{migr} = 1/\gamma$, where $\gamma = K(pa2) = K(pm2)$.

C. *Unfolding of FMSRN sub- models*

It can be demonstrated that any model $FM\Gamma$ type FMSRN can be unfolded in a SRN with fuzzy parameters (FSRN) model with the same attributes and behavioral properties. This allows to use the available Petri Nets Software Tools [10] to perform their analysis. In order to better understand the logic of the elaborated submodels, we present in Fig. 3 and Fig. 4, respectively the sub-models $F\Gamma 1$ and $F\Gamma 2$, obtained by unfolding of sub-models $FM\Gamma 1$ and $FM\Gamma 2$, respectively for the case $\gamma = 3$, i.e. the DCP has three available VMs.

The meaning of places and transitions in $F\Gamma 1$ (resp. $F\Gamma 2$) is clarified by those of $FM\Gamma 1$ (resp. $FM\Gamma 2$) sub-model as follows: $pa0,1 := pa3$, $pa0,2 := pa4$;

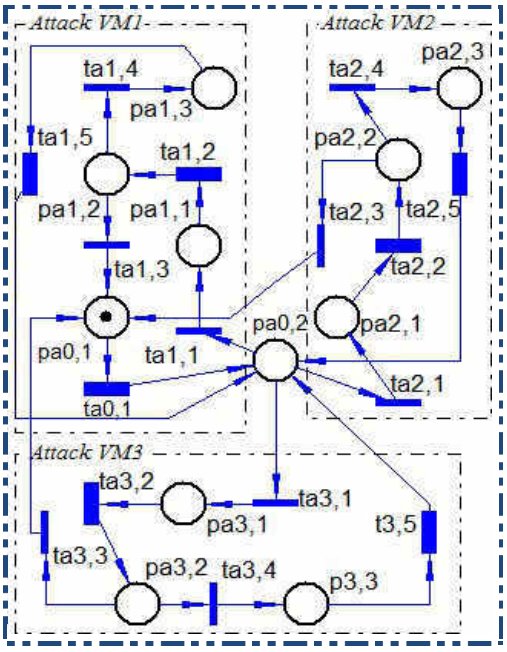


Figure 3. Attack $F\Gamma 1$ sub-model of 3 VMs

$pai,2 := pa6$, $ta0,1 := ta3$; $pai,1 := pa5$, $pai,3 := pa7$ and $tai,1 := ta4$, $tai,2 := ta5$, $tai,3 := ta7$, $tai,4 := ta6$, $tai,5 := ta8$; $p0,1 := pm7$, $p0,2 := pm3$; $p0,3 := pm4$, $p0,4 := pm12$, $pi,1 := pm5$, $pi,2 := pm8$, $pi,3 := pm9$, $pi,4 := pm11$, $pi,5 := pm6$ and $t0,1 := tm8$, $t0,2 := tm3$, $ti,1 := tm4$, $ti,2 := tm5$, $ti,3 := tm6$, $ti,4 := tm11$, $ti,5 := tm10$, $ti,6 := tm12$, $ti,7 := tm14$, $tmi,8 := tm13$, $ti,9 := tm7$, $ti,10 := tm9$, where $i = 1, 2, \ldots, \gamma$ represents the serial number of the attacked (resp. task migration) VM_i (here $\gamma = 3$) which, in the $FM\Gamma 1$ and $FM\Gamma 2$ sub-models, is determined by the dynamic weight vector parameter $W(ta4, pa5) = [\mathbf{Wa}]$ and $W(tm4, pm5) = [\mathbf{Wm}]$, respectively.

Thus, the number of graphical elements of the $F\Gamma 1$ and $F\Gamma 2$) *unfolded* sub-models depends on the number γ of VMs in DCP: $N_1^P = 3 \cdot \gamma + 2$ and $N_2^P = 6 \cdot \gamma + 4$ places, $N_1^T = 5 \cdot \gamma + 1$ and $N_2^T = 10 \cdot \gamma + 3$ transitions, $N_1^A = 10 \cdot \gamma + 2$ and $N_2^A = 21 \cdot \gamma + 7$ arcs, respectively, which leads to huge dimension and cumbersome FSRN models and thus complicates their analysis presentation.

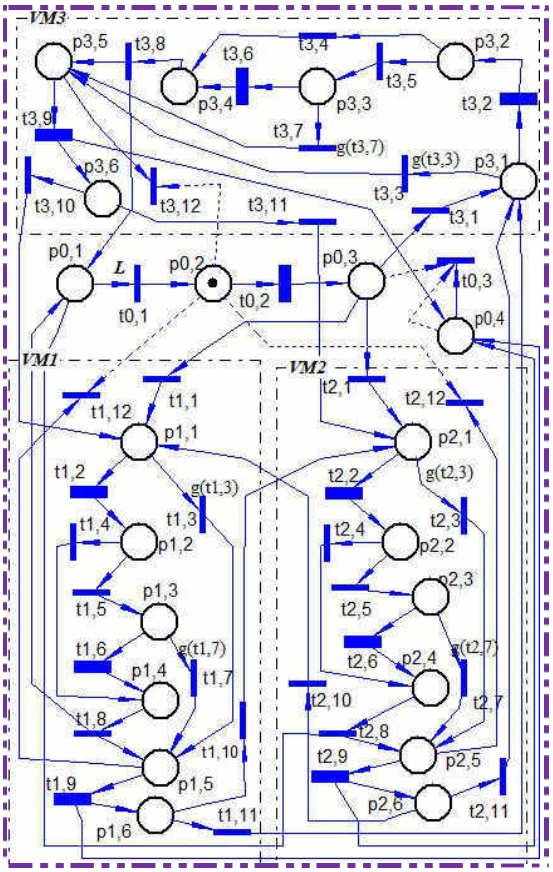


Figure 4. $F\Gamma 2$ sub-model of task migration between 3 VMs in DCP

https://doi.org/10.52326/ic-ecco.2021/SEC.02

V. NUMERICAL CASE STUDY

Next we will present a numerical case study to show the use of the approach presented in this paper. The numerical analysis of some QoS metrics of this MTD security system is based on the unfolded $FMT1$ and $FMT2$ models, using the knowledge of the experts in the field [4]. As an example, we establish the parameters:

1) *crisp values*: $\tau_{Ask} = 20\ days$ - task completion exection time without attack; $L = 10$ - number of stages; $\bar{u} = 2\ days$ - mean time to execute the task (with coefficient of variation $K^v = 0.6$) by VM at a every stage, based on which we identify the firing rates $[\lambda_{tm5}]_{1\times\gamma} = [0.833]_{1\times\gamma}$, $[\lambda_{tm12}]_{1\times\gamma} = [1]_{1\times\gamma}$, $[q_{tm10}]_{1\times\gamma} = [0.8]_{1\times\gamma}$; $\lambda_{ta1} = \lambda_{ta2} = 1$, $\lambda_{tm1} = \lambda_{tm2} = 2.5$, $\lambda_{tm7} = \lambda_{ta8} = 48$, $\lambda_{tm3} = 2$.

2) Z-TrNFs *values*: $[\tilde{\lambda}^{Z_j}_{ta3}]_{1\times\gamma} = [(\tilde{\lambda}^{A_j}_{ta3}, \tilde{\lambda}^{R_j}_{ta3})]_{1\times\gamma}$ and $[\tilde{\lambda}^{Z_j}_{ta5}]_{1\times\gamma} = [(\tilde{\lambda}^{A_j}_{ta4}, \tilde{\lambda}^{R_j}_{ta5})]_{1\times\gamma}$, where $[(\tilde{\lambda}^{A_j}_{ta3}]_{1\times\gamma} = [(0.05, 0.08, 0.12, 0.20; 1)]_{1\times\gamma}$, $[\tilde{\lambda}^{A_1}_{ta5}]_{1\times\gamma} = [(0.1, 0.2, 0.5, 0.8; 1)]_{1\times\gamma}$, $[\tilde{\lambda}^{A_2}_{ta5}]_{1\times\gamma} = [(0.6, 1.0, 1.5, 2.5; 1)]_{1\times\gamma}$ and $[\tilde{\lambda}^{A_3}_{ta5}]_{1\times\gamma} = [(3.0, 5.45, 7.0, 9.0; 1)]_{1\times\gamma}$ with $[\tilde{\lambda}^{R_j}_{ta3}]_{1\times\gamma} = [\tilde{\lambda}^{R_j}_{ta5}]_{1\times\gamma} = [(0.8, 0.9, 1; 1)]_{1\times\gamma}$ fo $j = 1, 2, 3$, respectively. For these Z -TrNFs parameters we obtain the weight $\delta = 0.9$ of the $[\tilde{\lambda}^{R_j}_{ta3}]_{1\times\gamma} = [\tilde{\lambda}^{R_j}_{ta5}]_{1\times\gamma}$.

Conform [7], the average credibility values of Z -TrNFs parameters are: $[\bar{\lambda}_{ta3}]_{1\times\gamma} = [0.1]_{1\times\gamma}$ and $[\bar{\lambda}^1_{ta5}]_{1\times\gamma} = [5.50]_{1\times\gamma}$, $[\bar{\lambda}^2_{ta5}]_{1\times\gamma} = [1.25]_{1\times\gamma}$, $[\bar{\lambda}^3_{ta5}]_{1\times\gamma} = [0.35]_{1\times\gamma}$ which indicates the *strong*, *medium* and *weak* attack, respectively.

We used the VPNPtool [10] and PIPEtool [11] to evaluate the specified QoS metrics of given models.

Due to space limitations, in Fig. 5 are presented only the evolution of the *task expected completion time* $\bar{\tau}_R$ curves under different numbers of VMs and attack rates.

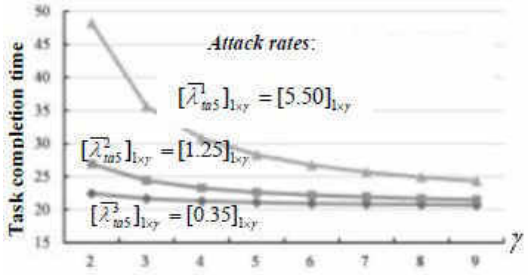


Figure 5. Expected task completion time under different γ numbers of VMs and attack rates

As we can see from the curve with triangular signs, $[\bar{\lambda}^1_{ta5}]_{1\times\gamma} = [5.50]_{1\times\gamma}$, when the number of VMs increases from 2 to 9, the task completion time decreases from 49.6 days to 23.5 days. The main reason is that as the number of VMs increases, the task execution process is less likely to be destroyed by the attacker.

Also, it should be noted that the use of a MTD technique introduces additional costs of calculation, the total processing time of a task will be affected and the performance of the defended system will be reduced. Therefore, users choose a cloud service provider and pay for the VMs for the task execution. According to the service level agreement(SLA), there is a slowest time threshold τ_{SLA}, and the defender must complete the task execution before τ_{SLA}. Thus, the sooner the task is completed, the more benefits the defender will get. Let c_B indicate the benefits the defender earns each day before the threshold. Also, the user needs to pay for the cost of the VM. Let c_{VM} *dollars/hour* to indicate the cost of one VM .

Given these parameters, the defender's profit P_{rofit} will be calculated as follows: $P_{rofit} = c_B \cdot \tau_{SLA} - (c_B - \gamma \cdot c_{VM}) \cdot \bar{\tau}_R$.

We mention that our modeling approach is flexible and can be used on other types of MTD.

In a future work, we will focus on developing a dynamic reconfigurable FMSRN models to evaluate the trade-offs between security and performability of MTD environments based on different VMs policies placement and task migration scheduling algorithms.

REFERENCES

[1] J. Zheng, and A. S. Namin, "A survey on the moving target defense strategies: An architectural perspective," *Journal of Computer Science and Technology*, 34(1), pp. 207–233, 2019.

[2] M. Torquato, and M. Vieira, "Towards models for availability and security evaluation of cloud computing with moving target defense," 2019, *arXiv:1909.01392*. [Online]. Available: https://arxiv.org/abs/1909.01392

[3] J. Muppala, G. Ciardo, and K. S. Trivedi, " Stochastic reward nets for reliability prediction," *Commun. Reliab. Maintainab. Serv.*, vol. 1(2), pp. 9–20, 1994.

[4] Z. Chen, *et al.* , " Numerical Evaluation of Job Finish Time Under MTD Environment," ***IEEE****Access*, vol. 8, pp. 11437-11446, 2020. DOI: 10.1109/ACCESS.2020.2965090.

[5] L. A. Zadeh, "A note on Z-numbers," *Information Science*, no. 181, pp. 2923–2932, 2011.

[6] B. Kang, D. Wei, Y. Li, and Y. Deng, " A Method of Converting Z-number to Classical Fuzzy Number," *Journal of Information & Computational Science*, 9: 3, pp. 703-709, 2012.

[7] B. Liu, and Y. Liu, "Expected value of fuzzy variable and fuzzy expected value model," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 4, pp.445–450, 2002.

[8] E. Guţuleac, S. Zaporojan, I. Gîrleanu and V. Cărbune, "Hybrid stochastic Petri nets with matrix attributes for modelling of discrete-continuous process," *Meridian Ingineresc, no.* 2, pp. 34–40, 2016.

[9] E. Gelenbe, et all., *Réseaux de files d'attente : modélisation et traitement numérique.* Ed. Hommes et Techniques. Monographies d'Informatique de L'AFCET, 1980.

[10] E. Guţuleac, C. Boşneaga, A. Reilean,"VPNP-Software tool for modeling and performance evaluation using generalized stochastic Petri nets," In: *Proceedings of 6-th International Conference on D&AS-2002*, Suceava, România, 2002, pp. 243-248.

[11] Petri Nets Tool Database. May. 11, 2021. [Online]. Available: https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db.html