A Model-Based Safety-Security Risk Analysis Framework for Interconnected Critical Infrastructures

Rajesh Kumar

▶ **To cite this version:**

Rajesh Kumar. A Model-Based Safety-Security Risk Analysis Framework for Interconnected Critical Infrastructures. 14th International Conference on Critical Infrastructure Protection (ICCIP), Mar 2020, Arlington, VA, United States. pp.283-306, 10.1007/978-3-030-62840-6_14 . hal-03794643

HAL Id: hal-03794643
https://inria.hal.science/hal-03794643v1

Submitted on 3 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Chapter 1

SECSAFE: A MODEL-BASED SAFETY-SECURITY RISK ANALYSIS FRAMEWORK FOR INTERCONNECTED INFRASTRUCTURES

Rajesh Kumar

Abstract Interconnected architectures are complex owning to several attributes: temporal evolution, dynamic dependencies between the architectures, component-level dependencies, presence of adversaries, etc. There exists a plethora of literature on safety and security risk assessment of isolated architectures. However, extending these techniques to interconnected architectures is usually infeasible due to the inherent complexity and lack of a generic modelling tool. This paper provides a framework SecSafe to model and analyze interconnected architectures. Secsafe is a layered framework with (a) a higher modelling layer representing the functional dependencies of the infrastructures. Each architecture is refined to component-level disruption and is represented using a novel combination of dynamic reliability block diagrams and attack-fault trees, and (b) a lower analysis layer based on stochastic timed automata (STA). The STAs serves as semantic framework for the elements in higher layer. While the high layer can graphically represent the complex dependencies, inter-dependencies, cascading and temporal disruption scenarios; the lower analysis layer provides a rigorous foundation to analyze these relationships using formal verification, in particular via statistical model checking. Furthermore, the lower layer provides a flexible way to incorporate the different quantitative attributes of the system like probability, time, costs, etc. We quantify the various metrics-of-interest such as reliability analysis, provide time-dynamic analysis and investigate the "what-if" scenarios. We demonstrate our approach using a real industrial incident of a disruption scenario. We argue that with Secsafe, an analyst can identify the weakest link, evaluate protective measures and take transparent decision on the investment strategies.

Keywords: Interconnected architectures, Risk analysis, Stochastic timed automata, Attack-fault tree, Dynamic reliability block diagram

1. Introduction

Modern-day society is built upon several critical infrastructures (CIs) such as telecom networks, water supply, banking systems, etc. Many of these systems are strategically interconnected with each other, exchanging logical and physical flows such as information, power, etc., resulting in a system-of-systems [**?**, 14]. Moreover, as these infrastructures have grown, expanded, networked and interconnected, they have become extremely vulnerable to a plethora of threats, ranging from natural hazards to terrorism and from operational failures to mechanically induced disruptions. Furthermore, due to the strong intertwining of CI services, a small malfunction in one architecture may quickly cascade to other infrastructures, resulting in devastating impact, sometimes bringing the entire industrial operations to a complete standstill. For example, the 2012 India blackouts [25], where the disruption of the northern sections of the Indian power grid on 30-31 July 2012, immediately impacted several services such as railways, metro, and emerge with longer effects on banking, telecommunication, etc. Thus, given the complex, multi-actor and the decentralized nature of interconnected infrastructures, CI owners need an active risk management framework to understand the disruption scenarios, both from the organizational point and from the system-of-systems level perspective. Important questions of interest to them are: What is the probability of a disruption? Or Will the countermeasures put at upstream infrastructure effect the downstream infrastructure? Or Where should the countermeasures be placed so that the overall reliability is improved? etc. Knowing the answers to the aforementioned questions, CI owners can make transparent decisions about risk-mitigating measures that can increase the resilience of the overall system.

To address the questions mentioned in the previous paragraph, in this paper, we present a generic model-based quantitative analysis framework Secsafe. Secsafe allows an analyst to decompose the complexity of interconnected architecture into smaller manageable parts. The analysis can be then done for such individual parts or holistically, taking into account all the different parts together. Technically, Secsafe combines two popular dependability models: the dynamic reliability block diagrams (DRBD, [**?**, 13]) and the attack fault trees (AFT, [23]). Quantitatively, to take into account the different functionally-related system attributes such as cost structures, time-dynamic attributes, we derive a stochastic timed automaton model (STA, [11]) of the combined DRBD+AFT model. We use the popular tool of Uppaal SMC [11] to perform statistical model-checking (SMC). SMC allows simulation of complex systems

where a simple closed-form solution does-not exist or a rigorous state-space search is infeasible, as in our case studies.

Related Work. In recent years, several attempts have been made to model the interdependent architectures providing related terminologies and discuss potential approaches [**?**, 29]. In [27], authors classify the modelling techniques into several broad categories, including empirical, agent-based, system dynamics based, economic theory-based and network-based approaches, among others. Based on an exhaustive survey of different analysis tools and frameworks, in [34] authors conclude that most literature use either agent-based or network-based analysis techniques to model interdependence in critical infrastructure. In [33], Stergiopoulos et. al. uses graph-based techniques for representing dependencies, concurrent cascading and common-cause failures. However, the aforementioned paper is restricted to high-level analysis while our paper permits both high level and component-level analysis. Other sector-specific analysis of interdependencies such as between the power and telecommunication sector is given in [5].

A popular strand of reliability and safety analysis is by using graphical models such as of the fault trees (FTs), reliability block diagrams (RBDs) and event trees (ETs) [**?**, 28]. Quantitative analysis of these models is well-understood using the simple laws of logic and probability theory. In the same spirit of model-based reliability analysis, model-based security analysis is an emerging area of interest. A survey of different FT variants and analysis techniques is given in [30].

Few popular frameworks such as of ATs [**?**,20], ADVISE framework [15], Boolean driven Markov process [7] exist. Though ATs and FTs have been used in many practical case studies such as of ATM security [16], none of them has been deployed for interconnected architectures. Similar to the several existing analysis techniques for FTs and RBDs, there has been several tools and techniques for AT analysis [22]. Few techniques translate ATs to transition diagrams such as of petri-nets, stochastic activity networks [31], priced timed automata [24], Markov chains [2] etc. Petri-net based formalisms such as stochastic petri-nets are very expressive with great modelling power. However, as noted in [7], there are too general, in the sense that sometimes the processing become intractable.

2. Proposed approach.

An overview of our approach is given in Figure 1. The first step is to decompose a given network of architectures into sub-architectures (for example, here the `Power network` and `Telecom Network`). These sub-architectures are further decomposed into different systems (for ex-

ample, here the `Power network` comprises of `Gen`, `Trans`, `Dist`). The second step is to construct a dynamic reliability block diagram (DRBD) based on one single service outputted by the synchronized operation of the architectures. If the analysis needs to be done for more services, we need to construct a separate DRBD for each service. DRBD consists of blocks, shown as boldface rectangular boxes in Figure 1. These blocks can be connected under different configurations such as series, parallel, etc. depending on the functional description of the system components. For example, in Figure 1, the block `Power Network` is connected in series with `Telecom network` (the DRBD configurations are detailed in next paragraphs). We extend the traditional DRBD formalism with additional dashed blocks, for example the DRBD block of `Switch` shown in the figure. These dashed blocks are similar to the traditional DRBD blocks and additionally embeds an attack-fault tree (AFT, [23]) models is described in details in the next paragraphs). An AFT is a succinct representation of all the disruptions scenarios that result in the disruption of the DRBD block. The third step is to derive a stochastic timed automaton model of the combined DRBD+AFT model. Stochastic timed automaton (STA) are transition diagrams with stochastic semantics. We construct the stochastic timed automata of the combined DRBD+AFT model in a compositional manner inspired by the compositional aggregation approach of Boudali et. al [6]. This automaton is then fed into

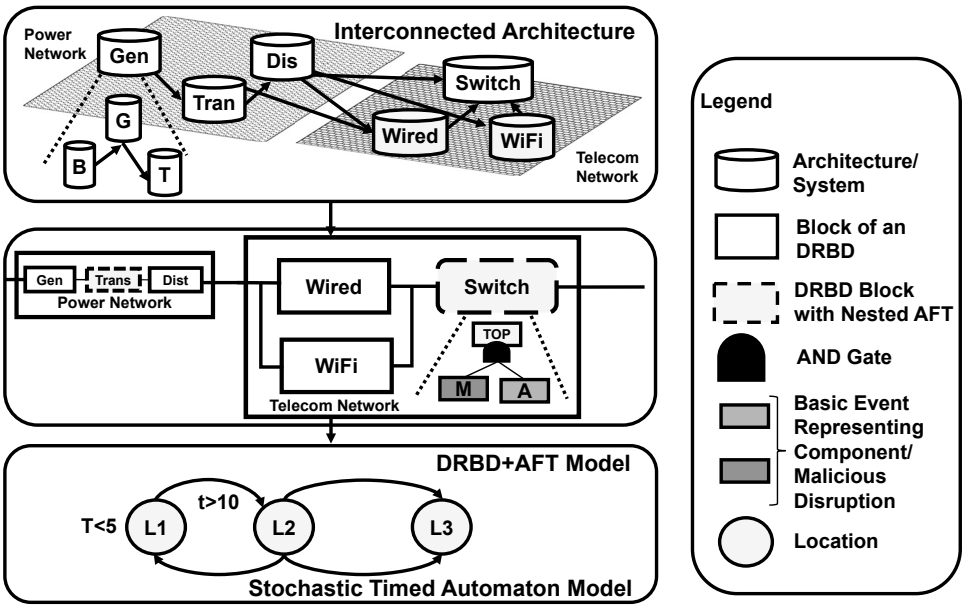


Figure 1. An overview of the proposed approach.

the model-checker along with the metrics of interest that are encoded in temporal logic. We obtain the results using the Uppaal SMC model-checker [11]. Below we briefly introduce the DRBDs and AFTs.

3. Building blocks.

Below we translate the DRBD+AFT diagram to stochastic timed automaton model that serves as one the input to the model-checker. The other input to the model checker is the encoded metrics-of-interest in temporal logic.

In this section, we briefly introduce the formalism of DRBDs and AFTs. Then, using DRBD and AFT syntactic constructs, we model the case study of interconnected architecture taken from the literature. We use this case study to showcase our methodology and results. Below we first briefly describe the DRBD different configurations.

DRBD configurations. Reliability block diagrams (RBD, [19]) consists of blocks which are connected in different configurations by edges (shown in Figure 2). Typically RBD analysis is traditionally time-invariant [28]. Moreover, traditional RBD diagrams assume independence of events which makes the analysis easy, however, is not very practical. DRBD extends RBD with dynamic features [35].

Figure 2 shows the RBD block and the standard configurations of series and parallel. In the series configuration, `N` blocks are placed in series, and the system is said to disrupted if any of the blocks is disrupted. In the parallel configuration `N` blocks are placed in parallel, and the system is said to disrupted if all the blocks are disrupted. In [35], authors extend the traditional RBD formalism to dynamic reliability block diagrams (DRBD), adding more syntactic constructs to model the dynamic

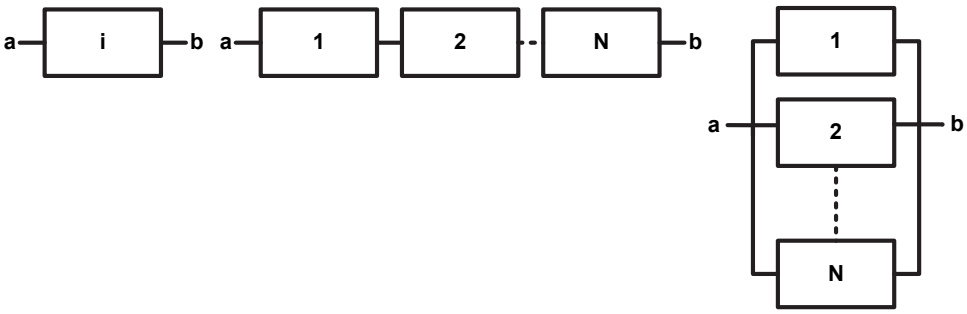


Figure 2. Standard RBD configurations (from left to right): a block representing the functional system component C_i with end points **a** as the input and **b** as the output, Series configuration with `N` number of blocks and Parallel configuration with `N` number of blocks.

dependencies and additional behaviour that cannot be modelled by standard RBDs, such as load sharing. Figure 3, shows the two DRBD blocks: SDEP and SPARE. These DRBD blocks extend the RBD formalism by characterizing each block with three states: active, standby and fail. An SDEP consists of a block "Trigger" and `N` number of blocks. When the event given by trigger occurs (`A`,`D`, or `F`), it will force any of the dependent events (`A`, `D`, or `F`) to occur. Here, `A` stand for the activation event that leads to an active state of the component, `D` stands for deactivation event occured on a component that leads to a standby state of that component, and `F` stands for disruption event occurred on a component that leads to a fail state of that component. Thus, SDEP allows to model nine types of dependency relationship among system components (`A`, `A`), (`D`, `D`), (`A`, `F`), (`D`, `A`), (`A`, `D`), (`D`, `F`), (`F`, `A`), (`F`, `D`), (`F`, `F`). The SPARE block models spare management. It consists of a primary and several spare components that can be cold, warm or hot. On deactivation or failure of the primary component, the second component is activated, on so on.

Attack-fault trees. An attack-fault tree [23] combines two popular formalism of attack trees [20] and dynamic fault trees [6].

An AFT consists of a top node which represents the unwanted event of a disruption. This event is refined using logical gates of AND, OR, SAND and SPARE until we reach to the child nodes. All these gates are shown in Figure 4. An AND gate model that all its children need to be disrupted for the disruption of its parent node, OR gate models that

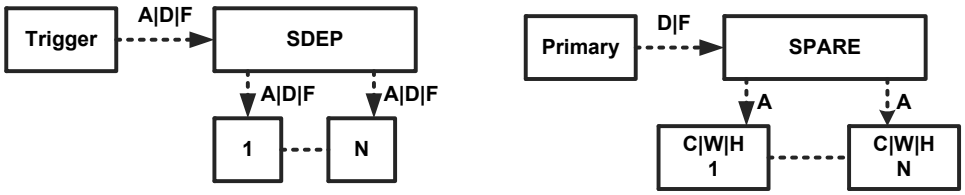


Figure 3. DRBD additional configurations (from left to right): SDEP block consisting of Trigger and `N` number of blocks, SPARE block consisting of a Primary component block and `N` number of blocks. Each of these block can be a cold/warm or hot spare. `A`, `D` and `F` refers to activation, deactivation and disruption signals respectively.

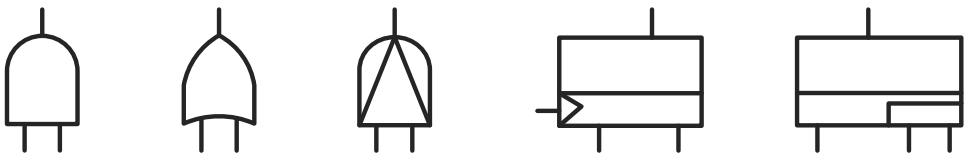


Figure 4. Standard and dynamic fault tree gates (from left to right): AND, OR, PAND, FDEP and SPARE gate

the disruption of any of the child node results in the disruption of the parent node, SAND gate is behaviorally similar to an AND gate, and models the constraint that the parent node is disrupted only when the disruption of its children are in an order from left to right, an FDEP gate models a disruption that can be attributed to a common cause. It consists of a trigger event and several dependent children. When the trigger event occurs, all dependent events are disrupted. A SPARE gate consists of a primary, the leftmost child of the SPARE gate, while all other children are SPARES. When the primary child fails, the SPARE gets attempts to a working SPARE child (in case of shared spare module, to the spare available).

Below, we provide a power outage incident, the so-called Rome scenario taken from the literature [4]. We use this real incident to build our model and perform quantitative analysis.

Running example. The "Rome scenario" refers to the outage of critical supervision, control and data acquisition (SCADA) communication links on January 2, 2004, in Rome, rendering many other infrastructures such as telecommunication, transport unavailable. We limit our attention to the interplay of two interconnected networks of electric infrastructure and industrial communication networks (SCADAs, communication links that connect SCADAs to Remote terminal units (RTUs)) that cooperate with each other to provide uninterrupted electric power. On one hand, the electric infrastructure relies on its SCADA network to perform remote diagnosis, telemetry, control, etc. On the other hand, the different telemetry resources such as radio communication, fibre-optics cables relies on an uninterrupted supply. Important to note that the different architectures mentioned here are operated by the different stakeholders. Hence, a quantitative analysis of such incident is important to fix the responsibilities in such shared assets by finding whose infrastructure/ system components are most vulnerable to disruptions. The different components involved in the Rome scenario are:

Electric power infrastructure. The electric power infrastructure comprises of two major subsystems: a) the power components that comprises of bus-bars, switches, circuit breakers, medium voltage (MV) transmission lines, high voltage (HV) transmission lines etc. and b) the control equipments which comprises of SCADAs, Remote terminal units (RTUs) and relays. The electric power infrastructure supplies power to the control equipments. The control equipments consists of primarily two SCADAs: the manned SCADA control (MSC) and the disaster recovery SCADA (DSC), both of which communicate over a redundant single-pair high-speed digital subscriber (SHDSL) based fibre optic link. The SCADAs observes different portions of the grid, however, the

DSC sends the observed parameters to the MSC for the control. Besides the SHDSL linking the two SCADA systems, the SCADAs are also connected with high voltage (HV) RTUs using default proprietary network (DPN) with public switched telephonic network (PSTN) serving as backup. Similarly, each SCADA is connected to medium voltage (MV) RTUs using global system mobile (GSM).

Power system spare. The power system spare is used as a back-up to power the industrial communication nodes in case of power outage of the electric infrastructure. Here, we consider that the spare power consists of diesel generator and battery.

Industrial communication network. The industrial communication network consist of several communication media linking the RTUs with SCADAs and the two SCADAs as mentioned in the previous paragraphs.

External infrastructures. The external infrastructure are the infrastructures that rely on the availability of power from the electric power infrastructure.

Figure 5 shows the high-level representation of "Rome scenario" using DRBD. Here the two blocks: the `Electric_power_infrastructure` and the `other_infrastructure` are put in series, which means that the unavailability of power due to disruption in electric power infrastructure will disrupt other infrastructures which depend on power. Next,

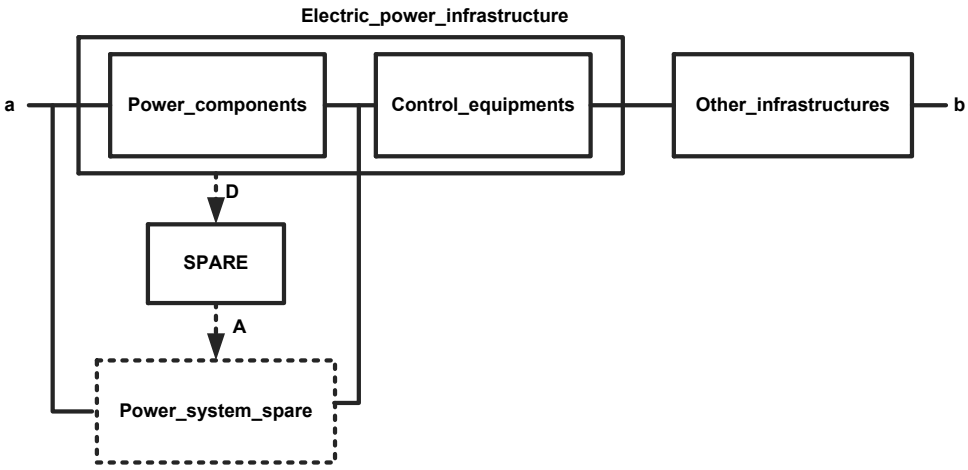


Figure 5. DRBD of Rome scenario. Here, the disruption of power renders other infrastructure stalled. Note, that the DRBD consists of two blocks: blocks indicated by bold lines and blocks indicated by dashed line. The block indicated by dashed line behaves similar to the box with bold line and additionally embeds an AFT. The `control_equipments` block is expanded in Figure 6, and the disruption of `Power_system_spare` embeds an AFT shown in Figure 7.

the electric power infrastructure consists of two sub-systems put as blocks: `Power_components` and `Control_equipments`, both put in se-

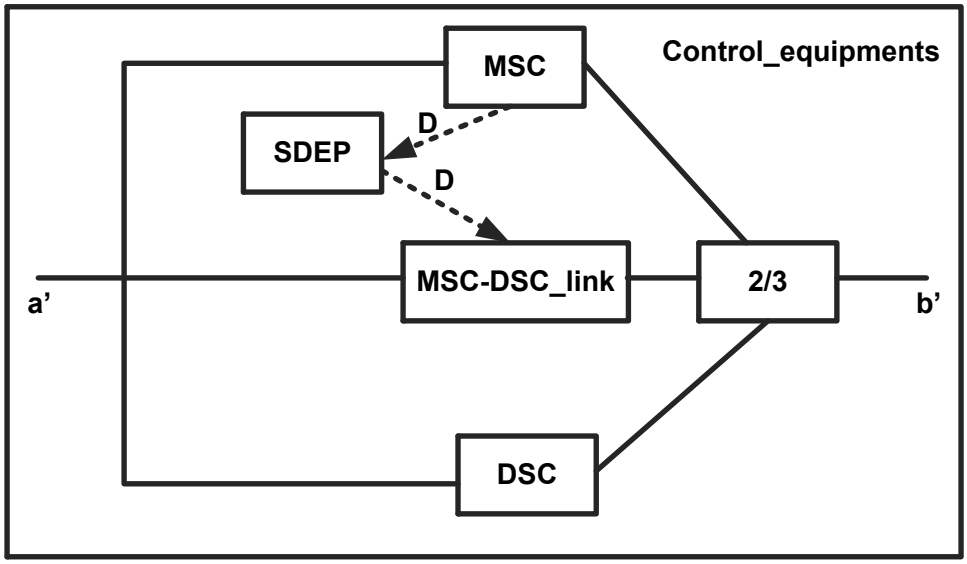


Figure 6. DRBD nested in the block of `Control_equipments`. The DRBD of the complete system is shown in Figure 5.

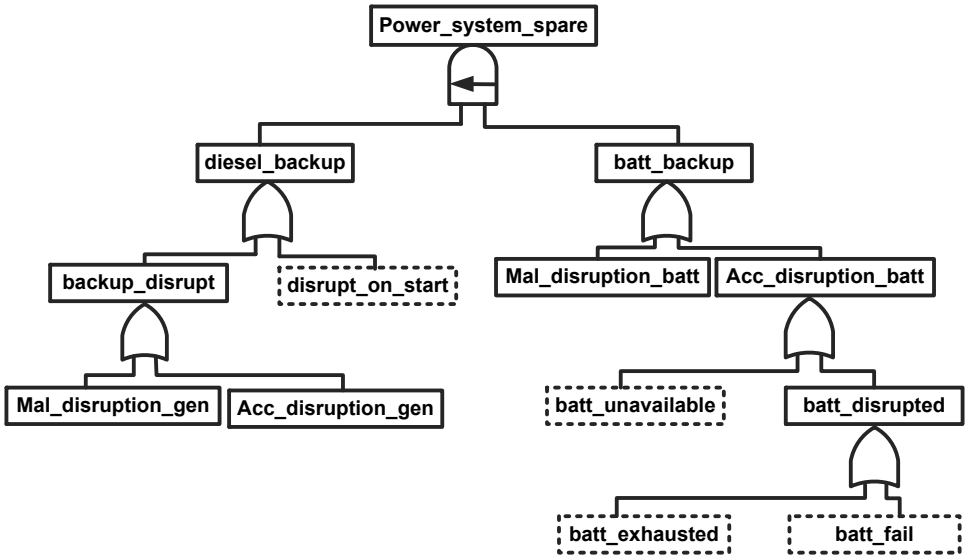


Figure 7. `Power_system_spare` AFT. The leaves drawn with black and bold lined rectangle represents a malicious failure. The leaves drawn with light black and dashed lined rectangle represents an accidental disruption.

ries to model that the disruption of any block will disrupt the electric power infrastructure. The block Power_components is placed in parallel with the block Power_system_spare, which means the for system disruption, both blocks of Power_components and Power_system_spare should be disrupted.

Figure 6 shows the control equipment functional block represented by DRBD. It consists of three basic blocks: MSC represents the subsystem MSC SCADA, DSC represents the subsystem DSC SCADA and MSC-DSC_link block represent the communication link connecting the two aforementioned SCADAs. Here, the blocks MSC, DSC and MSC-DSC_link are placed in parallel, further connected in series with 2/3 block, which means that if any of two blocks gets disrupted, the block Control_equipments gets disrupted. Furthermore, the disruption of blocks MSC immediately leads to the disruption of MSC-DSC_link. Thus, if MSC gets disrupted, the control_equipments DRBD basic block gets disrupted. However, if either the DSC or the link MSC-DSC_link gets disrupted, the Control_equipments block remains undisrupted. Each of the sub-systems represented by MSC, DSC and MSC-DSC_link can also get disrupted due to the disruptions of their system components. In this paper, we model the disruption of Power_system_spare using AFT (given in next paragraph). One can, in a similar fashion, construct the AFTs for MSC, DSC and MSC-DSC_link disruption if the user wants to perform more granular analysis.

Figure 7 starts with the top event of Power_system_spare that signify the disruption of spare power system. Here the top event can occur due to the disruption of diesel generator diesel_backup and the disruption of battery batt_backup, both serving as back-up option to supply power in case of main power supply outage. Here, we assume that as soon as the main power outage happens, first the battery back-up is activated. Once the battery back-up is disrupted, then the diesel back-up is activated. Hence, the event power_system_spare is refined using SAND gate. The event batt_backup can be disrupted due to any of the events of accidental disruption Acc_disruption_batt or malicious disruption Mal_disruption_batt. The accidental disruption of battery can happen due to any of the events of battery being unavailable batt_unavailable or battery being disrupted batt_disrupted. The event of accidental battery disrupted can occur due to the failure of battery batt_fail or battery upon exhaustion batt_exhausted. The event diesel_backup can occur due to any of the events of backup_disrupt or disrupt_on_start, hence refined by an OR gate. The event backup_disrupt can occur due to malicious disruption of diesel generator Mal_disruption_gen or Acc_disruption_gen.

4. Translation of DRBD+AFT into Stochastic timed automata

In this section, we provide the stochastic timed automaton templates for each element of our framework. Stochastic timed automata (STAs) are timed automata models (TA, [1]) with stochastic semantics. Consider the STA $\mathcal{S}_{\mathtt{v}}$ where the type of node `v` is DRBD basic block, as shown in the Figure 8. It consists of locations and transitions between these states. The locations represent the control states of the system, and transitions describe the behaviour when the system may move from one location to another. Additionally, real-valued variables clocks are used that keeps track of global time, which increase linearly over time but maybe reset when a transition is taken. One can specify constraints in terms of clocks to specify *invariants* that enforce deadlines or to specify enabling conditions as *guards* over the transitions. STAs, in contrast to TAs that allow only non-determinism, define a stochastic process where transition times are governed by probability distributions. Furthermore, one can compose multiple (S)TAs together using synchronisation signals on transitions. These signals take the form of `a?` which indicates that some transitions waiting for the signal `a` that can only be taken simultaneously with a transition in another STA emitting the corresponding signal `a!`. In Figure 9, the STA to start the entire system `Sys` is shown. Both the automaton of $\mathcal{S}_{\mathtt{v}}$ and $\mathcal{S}_{\mathtt{Sys}}$ communicate over broadcast signals (`act[id]`, `dis[id]`, `deact[id]`).

The STA $\mathcal{S}_{\mathtt{v}}$ consists of locations {`Initial`, `Activated`, `Disrupted`, `Deactivated`} and two types of transitions: the delay transition governed by the probability distributions (here put as invariant (`Rate_disr`) over the locations `Activated`. The STA $\mathcal{S}_{Sys}$ consists of the locations – {`Initial`, `wait_disrupt`, `Disrupted_system`}. This automaton initializes the system by emitting a broadcast signal `act[id]!` and then waits for a broadcast signal `dis[id]`?. After receiving that signal, it makes a transition to the "`Disrupted_system`" location, which indicates the disruption of the system.

Similarly, we provide the STA template for each DRBD + AFT element. Depending on actual DRBD+AFT model connections, these templates are instantiated appropriately. We utilize the compositional theory of timed input/output automata extended to stochastic timed automata [10] to compose the STAs together. We use the parallel composition operator || that allows one to construct a large STA from several smaller ones. If we denote by $S_{\mathtt{v}}$ the STA corresponding to the node `v` of DRBD+AFT $\mathcal{T}$, the complete automata model is given as a network of stochastic timed automata $\mathcal{N}_{\mathcal{T}} = \mathcal{S}_{\mathtt{V}_1} \,||\, \mathcal{S}_{\mathtt{V}_2} \,||\, \ldots \mathcal{S}_{\mathtt{V}_n} \,||\, \mathcal{S}_{\mathtt{sys}}$. This NSTA

serves as one of the input to our statistical model checker Uppaal SMC. The other input to the model checker is the encoded query of interest specified in Uppaal SMC specification language.

STA template for other DRBD elements. Below, we provide the STA template for series, parallel, SDEP and SPARE DRBD configurations.

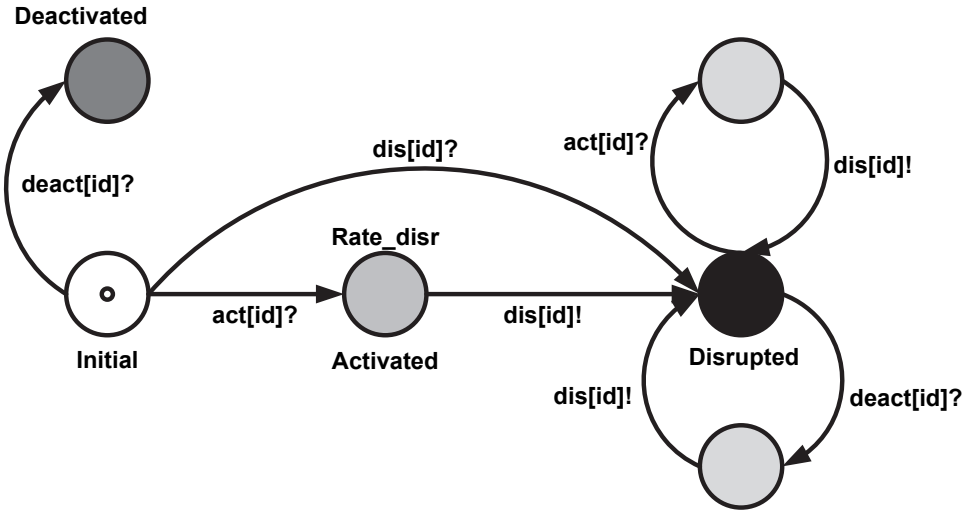


Figure 8. STA template for the DRBD basic block. Here `id` stands for DRBD block identifier. Here, we consider the disruption of the block is governed by exponential distribution with rate `Rate_disr`:$\lambda \in \mathbb{R}_+$.

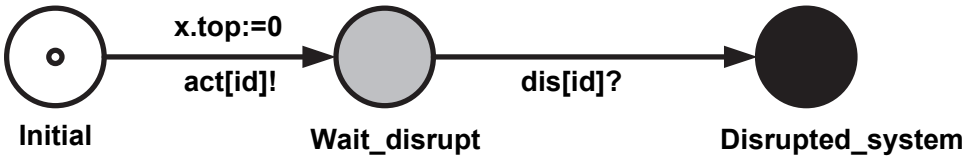


Figure 9. STA template for the system start. Here `x.top` is the global clock that keeps track of time.

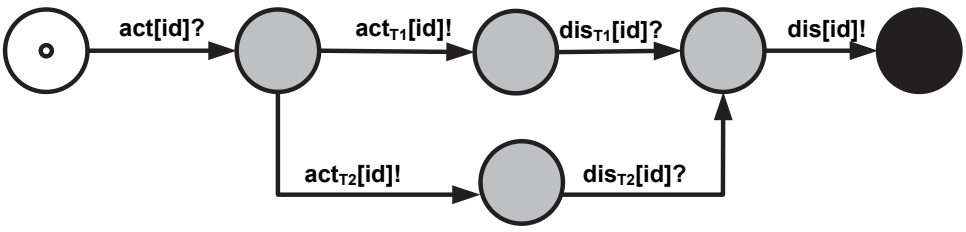


Figure 10. STA for DRBD series block configuration.

The STA template for the series RBD block configuration is shown in Figure 10. We consider that the DRBD series block configuration consists of two RBD basic blocks `T1` and `T2` connected in series. Once the STA receives the activation signal `act[id]?`, it activates both the blocks by sending $\texttt{act}_{\texttt{T1}}[\texttt{id}]!$ and $\texttt{act}_{\texttt{T2}}[\texttt{id}]!$. The DRBD series block configuration is disrupted if it receives `dis` from any of the two DRBD basic blocks. Then it sends the `dis[id]!`, signaling that the DRBD blocks in series is disrupted.

The STA template for the parallel RBD block configuration in Figure 11. We consider that the DRBD parallel block configuration consists of two DRBD basic blocks `T1` and `T2` connected in parallel. Once the STA receives the activation signal `act[id]`, it activates both the blocks. The DRBD parallel block configuration is disrupted if it receives `dis` from both the DRBD blocks.

Figure 12 give the STA template of SPARE block. Here, we consider that the spare block consists of a primary DRBD basic block given by `P` whose deactivation or disruption results in the activation of spare given by `T1`. The SPARE block gets disrupted if both `P` and `T1` gets disrupted. On either the deactivation or disruption of primary, the secondary components are activated.

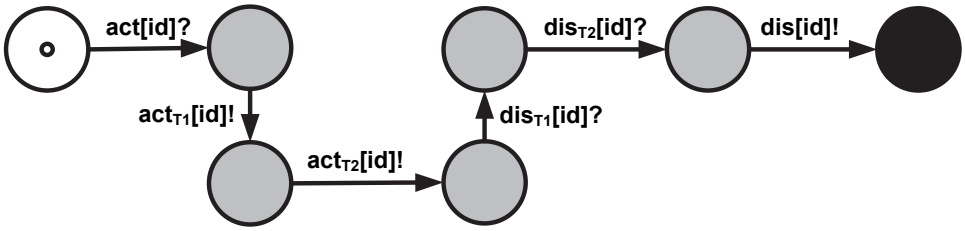


Figure 11. STA for DRBD parallel block configuration.

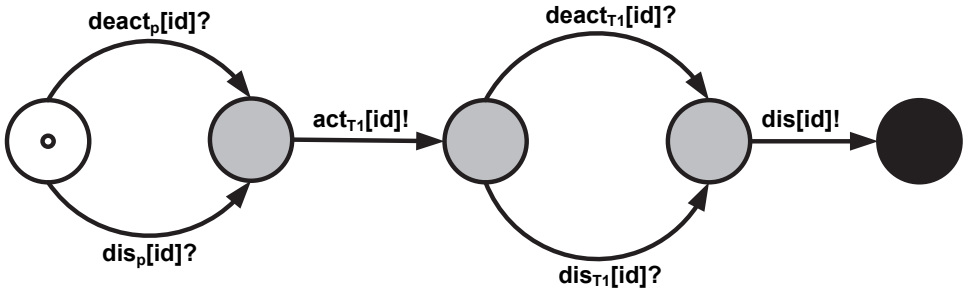


Figure 12. STA template for DRBD SPARE block.

Figure 13 give the STA template of SDEP block. The SDEP consist of a trigger and several dependent children. On activation of its trigger (similarly the deactivation and the disruption of the trigger), it can cause all its dependent children to activate, deactivate or disrupt, depending on the high level model, thus permitting several output signals. Here the SDEP consists of a trigger `T` and a child `P`. The SDEP block remains always activated and is ready to accept any of the signals of deactivation, activation or disruption of trigger. Depending on the system configuration it can then emit the activation/ deactivation or disruption of its child.

Since, in this paper, our semantics of DRBD using STAs does not alter the semantics of AFT elements given in [23] we do not repeat them in this paper. In the next paragraph, we encode the properties of interest which serves as one input to the model-checker, the other input being the NSTA model that we obtained by composing all DRBD+AFT automaton together.

4.1 Property verification using Uppaal SMC

In this paper, we perform the following scenario-analysis:

- **Sub-system view**: Here, we are interested in the probability of disruption of one subsystem. In our case, we are interested in obtaining the reliability over time of the subsystem of power system spare.
- **System-wide view**: Here, we are interested in the probability of disruption of the control system block. We also investigate here about the probability of disruption of electric power infrastructure

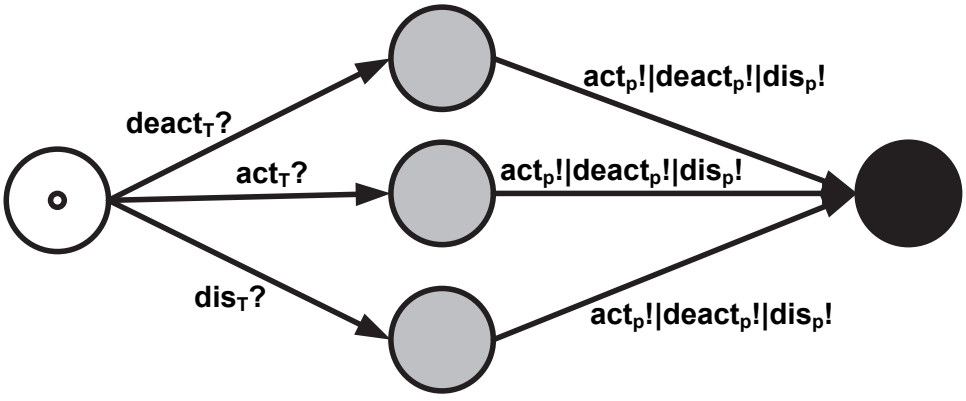


Figure 13. STA template for DRBD SDEP block.

without considering the spare power system.

- **System-of-System view**: Here, we are interested in the probability of disruption of electric power infrastructure taking along the spare power system. We also investigate here on how does the probability of disruption varies, if we disable all the nodes of the spare power system that can be disrupted maliciously.

To answer the aforementioned questions, we employ the UPPAAL model-checker. UPPAAL [3] is a platform for modelling, simulation, and verification (through model checking) of systems that can be modelled as networks of timed automata. In addition to the UPPAAL model checker for timed automata, the UPPAAL tool suite includes a *statistical model checker*, called the UPPAAL SMC, which extends the expressiveness of the modelling language of UPPAAL by supporting statistical model checking of the hybrid models.

MITL queries. In order to verify the properties of interest over the NSTA, we require to encode them into UPPAAL SMC specification language. The UPPAAL SMC specification language is a variant of Metric Interval Timed Logic (MITL) query language [8]. Formally, reliability $R(t)$ is the probability that a system remains undistributed in the interval from 0 to time t, i.e. $R(t) = \mathbb{P}(T > t)$. From practical point, we obtain unreliability in the NSTA $\mathcal{N}_{\mathcal{T}}$, which is given in MITL as: $\mathbb{P}_{\mathcal{N}_{\mathcal{T}}}(\diamond_{\leq t}\mathcal{S}_{Sys}.\texttt{Disrupted\_system})$ The aforementioned formula is a cost-bounded reachability query that asks for the probability that the random run [9] of the NSTA $\mathcal{N}_{\mathcal{T}}$ satisfies the state predicate $\mathcal{S}_{Sys}.\texttt{Disrupted\_system}$ within a cost given by t, where time t is the bound. In Uppaal SMC, this query is encoded as:

$$\texttt{P}[\leq \texttt{t}](<> \mathcal{S}_{Sys}.\texttt{Disrupted.System})$$

The mean time to successful disruption within time 't' is given as: $\mathbb{E}_{\mathcal{N}_{\mathcal{T}}}(T : \diamond_{\leq t}\mathcal{S}_{Sys}.\texttt{Disrupted\_system})$. Here, T is the accumulated time in a mission time t before disruption. In Uppaal SMC, we require two steps to obtain the aforementioned metric: a) Compute the expected time of disruption under a time bound t given by the query below; b) Divide the value obtained in the previous step with the probability of successful disruption within that time bound. In Uppaal SMC, the query to obtain the expected time under a time bound '$\texttt{t}$' is encoded as:

$$\mathbb{E}[\texttt{x\_top} \leq \texttt{t}, \texttt{N}](\texttt{max} : \texttt{x\_top} \times \mathcal{S}_{Sys}.\texttt{Disrupted\_system})$$

Here, $\texttt{N}$ is the number of simulation runs. Below we evaluate the "Rome scenario" over the aforementioned metrics.

DRBD block/AFT leaf	Disruption rate (1/MTTF)	Probability of instantaneous disruption
batt_fail	-	0.00023
batt_unavailable	-	0.000114
disrupt_on_start	-	0.00046
Mal_disruption_gen	0.0000114 (3600 days)	-
Acc_disruption_gen	0.000023 (1800 days)	-
batt_exhausted	0.00046 (90 days)	-
Mal_disruption_batt	0.00065 (64 days)	-
MSC	0.0000114 (3600 days)	-
DSC	0.000023 (1800 days)	-
MSC-DSC_link	0.00046 (90 days)	-
Power_components	0.0000114 (3600 days)	-

Table 1. Parameters used to decorate the DRBD blocks and the AFT leaves.

5. Evaluation of the "Rome scenario".

Experimental setup. All experiments were performed on an Intel Xeon CPU E5335 at 2.00GHz with 22GB RAM under Linux. As mentioned in previous paragraphs, we utilize the Uppaal SMC model-checker for verifying the properties. The statistical parameters chosen for the case studies are: the confidence interval $\alpha = 0.05$ and the probability uncertainty $\epsilon = 0.001$ (details on confidence interval and probability uncertainty are described in [11]). Furthermore, we also utilized another model-checker STORM [12] to compare the results.

Parameters used in the case study. As usual, in case of CIs, getting reliable data is difficult because of the confidentiality of data on disruptions and subsequently the unavailability of data repositories. The parameters used in the case study is given in Table 1. For each RBD block which is not further refined using AFT, we assign a disruption rate. Similarly, we decorate each leaf of the AFT with a disruption rate. The disruption rate is given as the inverse of "Mean Time to Failure (MTTF)". For few leaves in the AFT, which model instantaneous disruption such as `disrupt_on_start`, `batt_unavailable`, `batt_fail`, we do not assign the failure rate and put a disruption probability $P \in [0,1]$. This discrete probability models the instantaneous accidental disruption, for example when the component failed to start.

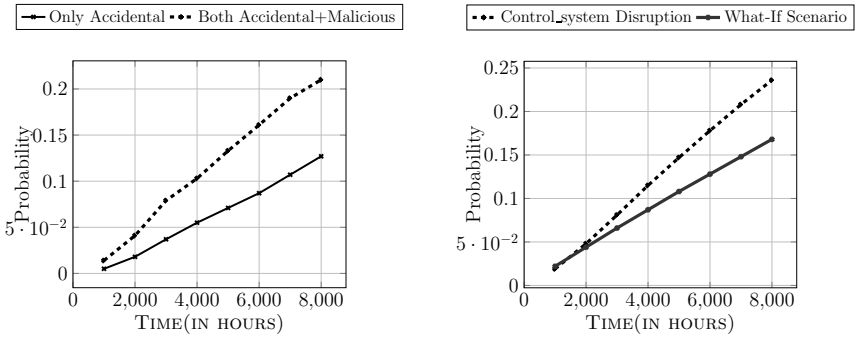
We assign the MTTF based on our knowledge to demonstrate our framework. In practical scenarios, it should be based on historical data. Note, that here though we have used only one parameter of MTTF in

RBDs and AFTs, this is not a limitation of our framework. Instead, one can decorate the leaves of the AFTs and RBDs with different cost structures that can stand for repair costs, costs incurred by disruptions, etc. Also, note that though we have used exponential distributions in our framework, which have been used in many reliability and security analysis, our framework supports acyclic phase-type distributions that can be used to approximate any probability distribution with arbitrary precision. Another important aspect is of model correctness, given that building models like ours is complex and can introduce subtle errors. We verify our model correctness at the design stage while drawing them in our tool Uppaal SMC. Here, Uppaal SMC tool has an in-built syntactic checker. Furthermore, it has an in-built simulator to check manually a run of the model and verify its correctness.

Results. We first analyze the disruption of small sub-system – power system spare, that was modelled by the AFT. In order to do so, we assign the parameters to the leaf nodes. Here, we consider the following two cases: A) Both the malicious and the accidental disruptions of battery and diesel generator. B) Only accidental disruptions. Concretely, we do this by enabling/disabling the leaves of our AFT. Figure 14(a) shows the cumulative ditribution function (CDF) for the unreliability of spare power system. In Uppaal SMC, we obtained the probability of disruption in a mission time of 1 year to be 0.23. In STORM model-checker, using the web interface of DFTCalc [18], we obtained the probability of disruption in the same mission time to be 0.226. This small difference in the values can be attributed to the difference in the model-checking techniques, with STORM based on model-checking and thus provides precise results while Uppaal-SMC is based on simulations. For Case B, the Probability of disruption in mission time of 1 year is 0.14 in Uppaal SMC, while the STORM model checker provides the result of 0.138. The results shows that malicious disruptions of components can result in overall higher probability of disruption of the system. The time for model checking in STORM model-checker took 87 seconds. The mean time to successful disruption in Case A is obtained to be 299 days. The mean time to successful disruption in Case B is obtained as 236 days.

Next, we consider the reliability analysis of the Control_components block, which was shown in Figure 6. In Figure 14(b), we plot the results. Based on the disruption parameters of `MSC`, `DSC` and `MSC-DSC` link, given in Table 1, we see that the probability of disruption in a mission time of 1 year is 0.236. One can similarly formulate the different "what-if" scenarios by configuring few sub-systems of the model to a disrupted state from the system start. For example, in our original model parameters, we have assumed that the link between the MSC and DSC is unreliable.

Suppose that the link between the MSC and DSC is 100 per cent reliable. In this case, both the MSC and DSC are functional, with the MSC taking control of the DSC functions in case of the DSC disruption. This modification results in a model of `Control_systems` DRBD to a modified DRBD with two blocks `MSC` and `DSC` in a parallel configuration. Running the query over the modified model yields the probability of disruption in a mission time of 1 year to 0.168 which is significantly lesser than the earlier disruption value of the `Control_systems` DRBD. This shows that the link between the SCADAs MSC and DSC is a considerable source of system disruption.

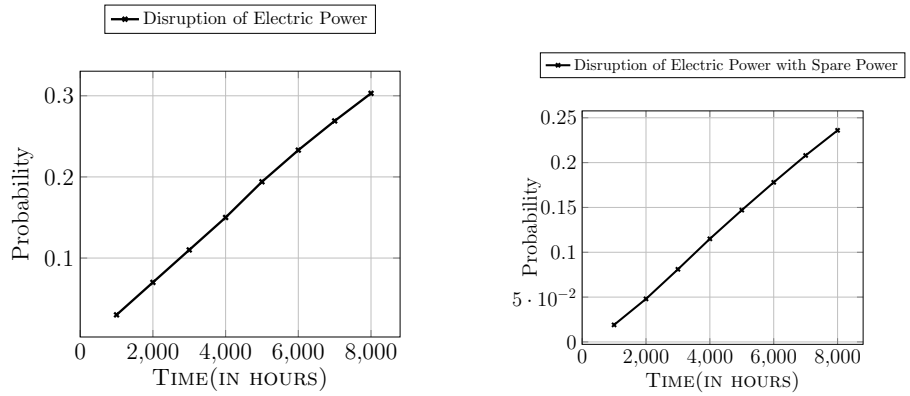


(a) Power System Spare (b) Control_systems

Figure 14. Probability of disruption over time of sub-systems of power system spare and control systems

In Figure 15(a), we plot the CDF representing the unreliability of the `Electric_power_infrastructure`. Here, we do not consider the spare power system to be the part of the `Electric_power_infrastructure`. The probability of disruption in thi case within a mission time of 1 year is 0.303. In Figure 15(b), we plot the unreliability of the `Electric_power_infrastructure` taking into account the spare power. The probability of disruption in this case within a mission time of 1 year is 0.303. We thus, see that having a spare power system reduces the unreliability of the system.

Similar to the above cases, one can run several what-if analyses. For example, one can perform sensitivity analysis to identify the most vulnerable components which have a maximum impact on the disruption values. To do this, we perform several runs of our model, wherein each run, we disable one component at a time. Furthermore, once we put the cost structures and repair rates into our DRBD blocks/ AFT leaves, we can ask more complex queries such as: What is the expected costs of a disruption? etc. With the aforementioned analysis results, an analyst



(a) Electric Power Infrastructure Without Spare Power

(b) Electric Power Infrastructure with Spare Power

Figure 15. Probability of Disruption Over Time of Electrical Power Infrastructure Under Different "what-if" Scenarios.

can identify the bottlenecks in the system-of-systems, thereby providing a way to implement measures, that can enhance the safety and security of the entire system.

6. Conclusion

Existing safety and security risk assessment works are not customized to the interconnected architectures. In this paper, we presented a generic framework of `Secsafe` to model the dynamic dependencies between the architectures using a new modelling approach that involves combining dynamic reliability block diagrams with a relatively recent formalism of attack-fault trees. With such a combination, we are able to capture the dynamic and complex aspects of different architectures such as redundancy policy management, load sharing, etc. using the high-level user-friendly syntactic notations of the reliability block diagrams. Furthermore, we used attack-fault trees to deductively depict the combination of component disruption that can lead to the sub-system disruptions. We presented the combined DRBD+AFT model for a real-case disruption scenario involving two interconnected architectures of "Power system" and "Telecom Network". We evaluate several different metrics and perform the "what-if" analysis over the aforementioned model. Our proof-of-concept work is adequate for the small case study discussed in this paper, however more work is needed to test for more complex models including repair policies, attack detection schemes, although the semantics of the combined DRBD+AFT models provide ample space for such extensions.

There are several research directions that follow our work. One is to how to automatically generate the DRBD+AFT models from the system specifications. In the present paper, we constructed the models manually which is a time-consuming and iterative task of finding all the disruption scenarios. Second, we believe that the validation of such a framework in practical scenarios is only possible when reliable input data is available. Third, we believe having a repository of reliability models for interconnected architectures can serve as a benchmark to compare and contrast the different results outputted by the several analysis tools.

References

[1] R. Alur and D. L. Dill, A theory of timed automata, *Theoretical Computer Science*, vol. 126(2), pp. 183–235, 1994.

[2] F. Arnold, D. Guck, R. Kumar and M. Stoelinga, Sequential and parallel attack tree modelling, in *Computer Safety, Reliability, and Security*, F. Koornneef and C. van Gulijk (Eds.), Springer, Cham, Switzerland, pp. 291–299, 2015.

[3] G. Behrmann, A. David and K. Larsen, A tutorial on UPPAAL, in *Formal Methods for the Design of Real-Time Systems*, M. Bernardo and F. Corradini (Eds.), Springer, Berlin Heidelberg, Germany, pp. 200–236, 2004.

[4] R. Bloomfield, P. Popov, K. Salako, V. Stankovic and D. Wright, Preliminary interdependency analysis: An approach to support critical-infrastructure risk-assessment, *Reliability Engineering and System Safety*, vol. 167, pp. 198–217, 2017.

[5] A. Bobbio, G. Bonanni, E. Ciancamerla, R. Clemente, A. Iacomini, M. Minichino, A. Scarlatti, R. Terruggia and E. Zendri, Unavailability of critical SCADA communication links interconnecting a power grid and a telco network, *Reliability Engineering and System Safety*, vol. 95(12), pp. 1345–1357, 2010.

[6] H. Boudali, P. Crouzen and M. Stoelinga, A compositional semantics for dynamic fault trees in terms of interactive Markov Chains, in *Automated Technology for Verification and Analysis*, K. Namjoshi, T. Yoneda, T. Higashino and Y. Okamura (Eds.), Springer, Berlin Heidelberg, Germany, pp. 441–456, 2007.

[7] M. Bouissou and J. Bon, A new formalism that combines advantages of fault-trees and Markov models: Boolean logic driven Markov processes, *Reliability Engineering and System Safety*, vol. 82(2), pp. 149–163, 2003.

[8] P. Bouyer, Model-checking timed temporal logics, *Electronic Notes in Theoretical Computer Science*, vol. 231, pp. 323–341, 2009.

[9] P. Bulychev, A. David, K. Larsen, A. Legay, G. Li and D. Poulsen, Rewrite-based statistical model checking of WMTL, in *Runtime Verification*, S. Qadeer and S. Tasiran (Eds.), Springer, Berlin Heidelberg, Germany, pp. 260–275, 2012.

[10] A. David, K. Larsen, A. Legay, M. Mikučionis, D. Poulsen, J. Vliet and Z. Wang, Stochastic semantics and statistical model checking for networks of priced timed automata, arXiv:1106.3961 (`arxiv.org/abs/1106.3961`), 2011.

[11] A. David, K. Larsen, A. Legay, M. Mikučionis and D. Poulsen, UPPAAL SMC tutorial, *International Journal on Software Tools for Technology Transfer*, vol. 17, pp. 397—415, 2015.

[12] C. Dehnert, S. Junges, J. Katoen and M. Volk, A STORM is coming: A modern probabilistic model checker, in *Computer Aided Verification*, R. Majumdar and V. Kunčak (Eds.), Springer, Cham, Switzerland, pp. 592–600, 2017.

[13] S. Distefano and A. Puliafito, Dependability evaluation with dynamic reliability block diagrams and dynamic fault trees, *IEEE Transaction on Dependable and Secure Computing*, vol. 6(1), pp. 4–17, 2009.

[14] I. Eusgeld, C. Nan and S. Dietz, System-of-systems approach for interdependent critical infrastructures , *Reliability Engineering and System Safety*, vol. 96(6), pp. 679–686, 2011.

[15] M. Ford, P. Buchholz and W. Sanders, State-based analysis in ADVISE, *Proceedings of the Ninth International Conference on Quantitative Evaluation of Systems*, pp. 148–157, 2012.

[16] M. Fraile, M. Ford, O. Gadyatskaya, R. Kumar, M. Stoelinga and R. Trujillo-Rasua, Using attack-defense trees to analyze threats and countermeasures in an ATM: A case study, in *The Practice of Enterprise Modeling*, J. Horkoff, M. Jeusfeld and A. Persson (Eds.), Springer, Cham, Switzerland, pp. 326–334, 2016.

[17] T. Gonschorek, M. Zeller, K. Höfig and F. Ortmeier, Fault trees vs. component fault trees: An empirical study, in *Revised Papers from the International Conference on Computer Safety, Reliability, and Security*, B. Gallina, A. Skavhaug, E. Schoitsch and F. Bitsch (Eds.), Springer, Cham, Switzerland, pp. 239–251, 2018.

[18] D. Guck, J. Spel and M. Stoelinga, DFTCalc: Reliability centered maintenance via fault tree analysis (Tool Paper), in *Formal Meth-*

ods and Software Engineering, M. Butler, S. Conchon and F. Zaïdi (Eds.), Springer, Cham, Switzerland, pp. 304–311, 2015.

[19] O. Hasan, W. Ahmed, S. Tahar and M. Hamdi, Reliability block diagrams based analysis: A survey, *AIP Conference Proceedings*, vol. 1648(1), 2015.

[20] B. Kordy, L. Cambacédès and P. Schweitzer, DAG-based attack and defense modeling: Don't miss the forest for the attack trees, *Computer Science Review*, vol. 13–14, 2014.

[21] R. Kumar, Truth or dare: Quantitative security risk analysis via attack trees, Ph.D Thesis, University of Twente, Netherlands, 2018.

[22] R. Kumar, S. Schivo, E. Ruijters, B. Yildiz, D. Huistra, J. Brandt, A. Rensink and M. Stoelinga, Effective analysis of attack trees: A model-driven approach, in *Fundamental Approaches to Software Engineering*, A. Russo and A. Schürr (Eds.), Springer, Cham, Switzerland, pp. 56–73, 2018.

[23] R. Kumar and M. Stoelinga, Quantitative security and safety analysis with attack-fault trees, *Proceedings of the Eighteenth IEEE International Symposium on High Assurance Systems Engineering*, pp. 25–32, 2017.

[24] R. Kumar, E. Ruijters and M. Stoelinga, Quantitative attack tree analysis via priced timed automata, in *Formal Modeling and Analysis of Timed Systems*, S. Sankaranarayanan and E. Vicario (Eds.), Springer, Cham, Switzerland, pp. 156–171, 2015.

[25] L. Lai, H. Zhang, C. Lai, F. Xu and S. Mishra, Investigation on July 2012 Indian blackout, *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 92–97, 2013.

[26] C. Nielsen, P. Larsen, J. Fitzgerald, J. Woodcock and J. Peleska, Systems of systems engineering: Basic concepts, model-based techniques, and research directions, *ACM Computation Surveys*, vol. 48(2), 2015.

[27] M. Ouyang, Review on modeling and simulation of interdependent critical infrastructure systems, *Reliability Engineering and System Safety*, vol. 121, pp. 43–60, 2014.

[28] M. Rausand and A. Høyland, *System Reliability Theory: Models, Statistical Methods, and Applications*, John Wiley and Sons, Heidelberg, Hoboken, New Jersey, 2009.

[29] S. Rinaldi, J. Peerenboom and T. Kelly, Identifying, understanding, and analyzing critical infrastructure interdependencies, *IEEE Control Systems Magazine*, vol. 21(6), pp. 11–25, 2001.

[30] E. Ruijters and M. Stoelinga, Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools, *Computer Science Review*, vol. 15–16, pp. 29–62, 2015.

[31] W. Sanders and J. Meyer, Stochastic activity networks: Formal definitions and concepts, in *Lectures on Formal Methods and Performance Analysis*, E. Brinksma, H. Hermanns, J. Katoen (Eds.), Springer, Berlin Heidelberg, Germany, pp. 315–343, 2000.

[32] R. Setola, and M. Theocharidou, Modelling dependencies between critical infrastructures, in *Managing the Complexity of Critical Infrastructures: A Modelling and Simulation Approach*, R. Setola, V. Rosato, E. Kyriakides and E. Rome (Eds.), Springer, Cham, Switzerland, pp. 19–41, 2016.

[33] G. Stergiopoulos, P. Kotzanikolaou, M. Theocharidou, G. Lykou and D. Gritzalis, Time-based critical infrastructure dependency analysis for large-scale and cross-sectoral failures, *International Journal of Critical Infrastructure Protection*, vol. 12, pp. 46–60, 2016.

[34] G. Stergiopoulos, E. Vasilellis, G. Lykou, P. Kotzanikolaou and D. Gritzalis, Classification and comparison of critical infrastructure protection tools, in *Critical Infrastructure Protection X*, M. Rice and S. Shenoi (Eds.), Springer, Cham, Switzerland, pp. 239–255, 2016.

[35] H. Xu, L. Xing and R. Robidoux, DRBD: Dynamic reliability block diagrams for system reliability modelling , *International Journal of Computers and Applications*, vol. 31(2), pp. 132–141, 2009.