# Stochastic modeling of safety and security of the e-Motor, an ASIL-D device

Peter T. Popov

Centre for Software Reliability, City University London, Northampton square, London, EC1V 0HB, United Kingdom
p.t.popov@city.ac.uk

**Abstract.** This paper offers a stochastic model and a combined analysis of safety and security of the e-Motor, an ASIL D (ISO 26262) compliant device designed for use with AUTOSAR CAN bus.

The paper argues that in the absence of credible data on the likelihood and payload of cyber attacks on newly developed devices a sensible approach would be to separate the concerns: i) the payloads that may affect the device's safety can be identified using standard hazard analysis techniques; ii) the difficulty with the parameterization of a stochastic model can be alleviated by applying sensitivity analysis for a plausible range of model parameter values.

**Keywords:** Stochastic modeling, adversary, safe state, cyber attack, ISO 26262.

## 1 Introduction

Cyber security of industrial electronics is becoming increasingly important as the Internet of things is becoming a reality. High profile vulnerabilities of embedded safety critical devices have been revealed, e.g. of insulin pumps, of pacemakers, etc. Researchers have developed exploits for these devices which demonstrate that patients' safety can be compromised remotely via cyber attacks.

Attention to cyber security has increased in automotive industry, too. Several demonstrations revealed that the CAN bus can be accessed remotely, e.g. via the entertainment (Bluetooth) system. CAN bus was designed as part of a "trusted" environment for which no authentication and authorization mechanisms were developed to protect the embedded devices connected to the CAN bus from malicious activities.

The good news is that despite the demonstrated vulnerabilities of automotive devices, the risk from exploiting these seems still relatively low and no significant accidents caused by cyber attacks have been recorded. The chip manufacturers, however, are taking proactive measures to create chips with built-in capabilities for enhanced security, e.g. the chips for automotive applications by Infineon and Freescale are manufactured with built-in *security processors*. Security analysis is increasingly taken seriously by the car manufacturers, too. Several research projects in the USA and in Europe are looking at the issue and attempt to lay down sound principles for future cyber security standardization. For example, the forthcoming extension of ISO 26262

is expected to at least acknowledge the importance of cyber security. Other safety standards, e.g. IEC/ISO 61508 (in v. 2010, clause 7.4.2.3), already acknowledge that cyber security must be an essential part of software safety analysis.

Cyber security for industrial control systems (ICS) has been discussed in a number of reports from various standardization bodies both in the US and in Europe. There is an essential difference between how cyber security is dealt within the ICT and in ICS in that the *reactive* approach, which dominates the ICT (patch as soon as a noteworthy vulnerability is discovered), may be inappropriate in the ICS. High availability and real-time requirements make patching difficult to implement and in many cases - simply inadequate.

Research effort has been allocated on demonstrating the benefits of proactive approaches to defending against cyber threats, e.g. using fault–tolerance, but this author is not aware of commercial solutions based on this approach.

## 2 Problem Statement and Related Research

Models of cyber attacks (including malicious software) have been proposed by many in the past. Cyber attacks broadly consist of a *delivery* mechanism, i.e. a mechanism of accessing the target, and a *payload*, the particular mechanism via which the attacker gains their rewards. These two can be seen as orthogonal: the same payload can be delivered via different mechanisms and the same delivery may be used to deploy different payloads. Conflicker worm, for examples used an aggressive delivery mechanism, but no particular payload for it has been identified [1]. At the other extreme, there are examples of a complex set of delivery mechanisms contained in a single malware. Stuxnet, for instance, is known to include among the delivery mechanisms 4 zero-day exploits and several different delivery mechanisms [1], some of which were known before Stuxnet, but have not been reported/fixed by the vendors.

Probabilistic models of cyber attacks have been used in the past ranging from very detailed models of a *particular attack* to models which operate at a relatively high level of abstraction suitable for exploratory analysis. An example of the first approach is [2] which models Stuxnet. An example of abstract models is the ADVISE formalism [3] and the various models of cyber economics. In between are models, which take a detailed look at the delivery mechanisms, e.g. [4] and make "pessimistic" assumptions about the payload. The same approach is quite common in ICT when getting an access to protected assets is seen as a "game over" event, as once the Adversary gains an access to the target they can do anything they please. Concentrating on delivery mechanisms and assuming the "worst" for the payload has been applied in ICS, too. For instance, [5] studied the impact of cyber attacks on a sub-station of a power transmission network and assumed that once the Adversary passes a substation firewall (s)he would switch off *all assets* controlled by the sub-station: generators, loads and bus-bars.

Given the wide range of delivery/payload combinations observed in cyber attacks and malware it is somewhat unclear how one should build *useful stochastic models* for security. My view is that attempts to model the delivery mechanisms in detail are

unlikely to be useful. Models which rely on detailed knowledge of a particular combination of delivery mechanisms, e.g. [2], provide a probabilistic *explanation* of what is already known, hardly a useful insight. One is usually, interested in studying the risk from attacks that have *not been seen before*. In my view modeling the delivery mechanisms should be done at a high level of abstraction, e.g. as a stochastic state machine with a small number of states which allow one to express the time (and effort) needed for a successful attack. The effect of the envisaged defense mechanisms on the likelihood of a successful attack must be modeled, too. A small number of parameters, characterizing the transitions between the states (that capture the possible multi-step paths of successful attacks) should allow a modeler to capture various scenarios and defense policies. This approach has been tried in the past [6].

While the delivery mechanisms should be modeled at relatively high level of abstraction, modeling the payload should be *as detailed as possible* and tailored to the specifics of the particular system. For safety critical systems enumerating the pertinent "failure modes" which can be caused by cyber attacks seems an essential starting point. Such an enumeration is typically a result of a *safety analysis*, which will produce the important hazards specific for the modeled safety critical system.

An essential problem for a useful probabilistic model is model parameterization. While with software reliability and safety putting in place a credible measurement program is usually sufficient for eliciting accurately the needed probabilistic parameters, for cyber security the feasibility of a measurement program is *questionable*. The international effort in this regard, e.g. with honey pots, provides plenty of evidence. Some colleagues seriously consider rethinking the concept of cyber risk in the light of this difficulty. The very idea of probabilistic security analysis is in doubt. Indeed if I cannot credibly parameterize a model how can I trust the findings from such a model about the risk from an unknown attack? The issue here is what we expect to learn from a probabilistic model? If the expectation is that we will be able to make predictions similar to those that we are able to make today about software reliability of a *specific* software system in its *specific operational environment* I do not believe that the problem can be solved! The issue is the Adversary profile, which can only be hypothesized, but is generally unknown and little can be learned from past observations about the next attack. Every new noteworthy attack looks like a "Black Swan" [7]. And yet, declaring that no meaningful probabilistic security analysis is possible will be in my opinion wrong, especially for those safety critical systems in which the primary concern is system availability and integrity. In such systems serious safety analysis is undertaken and the important hazards and failure modes are not merely identified but measures for error and failure detection, containment and recovery are provided. A good probabilistic security model will include two groups of parameters:

- *knowable*, i.e. with *low* epistemic uncertainty. Examples of such parameters would be the parameters related to accidental failures, their repairs, of the coverage of various detection mechanisms, etc., and
- *unknowable*, i.e. with *high* epistemic uncertainty. The parameters related to the behavior of an Adversary and even the Adversary models themselves fall into this category. For these parameters, however, one might be able to identify a range, possibly a large one, of *credible values*.

Epistemic uncertainty has been dealt with in the past, e.g. using the Bayesian analysis. Bayesian analysis asks for explicit quantification of epistemic uncertainty (in the form of a prior distribution), which may be problematic, and in the absence of a significant number of observations, the significant epistemic uncertainty may produce predictions, which are "too imprecise" to be useful. A reasonable alternative to a Bayesian assessment would be sensitivity analysis on the model parameters of the modeled safety critical system. Such an analysis would allow one to explore the space of *plausible* parameter values (without stating explicitly the epistemic uncertainty over this space) and to determine the range, for which the modeled system behaves *acceptably* and those for which the system behaves unacceptably (i.e. the cyber risk is too high). Sensitivity analysis can also be extended to the parameters of the mechanisms of cyber defense and help establish the parameter values, for which these mechanisms reduce the risk from cyber attacks to an acceptable level. Consider for example an intrusion detection/prevention system (IDS/IPS). In sensitivity analysis one can analyze the frequency of cyber attacks (subject to high epistemic uncertainty), for which cyber attacks pose too high a risk for the safety critical systems and "play" with IDS/IPS *coverage* to establish the coverage values, for which the risk from cyber attacks is reduced to a tolerable level. As a side effect "sensitivity analysis" may reveal that some (unknowable) parameters have low/negligible impact on system risk.

In this paper sensitivity analysis is demonstrated on a non-trivial case study.

## 3 The Model

The e-Motor case study was developed in the SESAMO (Artemis JU) project [8], [9] and is intended to be an ASIL-D[1] device. As a measure of safety assurance the device is designed to consist of *two diverse software channels*. The device is connected to the CAN bus of a car and is supplied by a torque request processed by the two channels independently. The control stimuli to the actual electric motor are provided by one of the two channels after adjudication of the outputs from the channels. The model presented in this paper is built under a number of assumptions about the architecture and the fault tolerant mechanisms used in the e-Motor, which are summarized below:

1. Each of the two channels is *self-checking*, capable to detect its own failures with some probability (coverage);
2. Each channel is provided with a *safe state*, e.g. a piece of code/data which can be used by the channel to move the device to a safe state, if instructed to do so by the "higher authority" (see below).
3. Each of the channels can be in one of the following states:

- working correctly;
- failed detected, which is further split into:
    - "safe failure state", if the *safe state* for the device has not been compromised. While in a safe failure state the channel can on its own complete a transition

[1] ASIL-D is the highest safety integrity level defined in ISO 26262 and requires the highest degree of rigor in development.

of the entire device to a safe state, i.e. irrespective of whether the other channel works correctly or has failed.
    - "Unsafe failure state 2" – if the safe state has been compromised. In this case, if the channel is instructed to move the device to a safe state, the device will find itself in an unsafe state.
- "Unsafe failure state" – if the channel has failed, but did not detect its failure.

4. The decision about which of the two channels controls the electric motor and executes a safe function (e.g. moves the device to a safe state) is taken by a "higher authority", which is outside the e-Motor itself. We assume that the higher authority is acting always *correctly*, i.e. if there is a channel that can to perform a control/safety function correctly (e.g. to move the device to a safe state) such as channel is always selected to do so. The e-Motor is essentially a 1-out-of-2 system with self-checking channels and a perfect adjudicator.
   The system state is established from the channel states as follows:
   - If at least one of the channels is OK, then the system is OK;
   - If none of the channels is OK, but at least one is in "safe failure state", then the system itself is in a safe failure state. If this situation occurs, the assumption is that the higher authority will instruct a channel in a "safe failure state" to move the e-Motor to a safe state. Further we assume that the transition to the safe system state occurs *instantaneously* and once it is started it is always completed successfully.
   - If both channels are neither in OK nor in "safe failure state", then the system enters (instantaneously) the unsafe system state.

The initial idea for combined analysis of safety and security of the e-Motor is shown in **Fig. 1** using the SAN formalism (Stochastic Activity Networks).

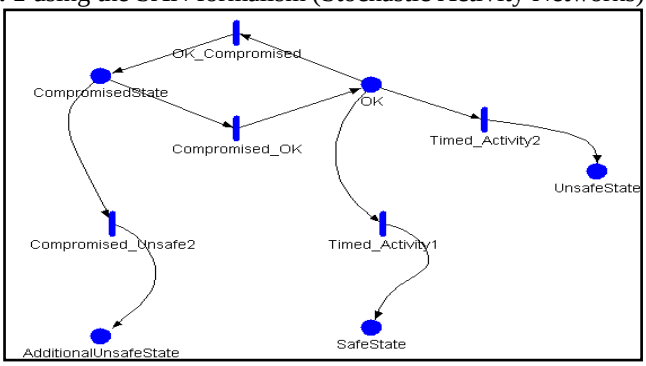



**Fig. 1.** A simplified SAN model of a channel of the E-motor

The "places" represent the states of the device: OK – represents the normal operation, CompromisedState – the state in which the safe state configuration is tampered with, SafeState – the state when proper reaction to a safety related event is taken, UnsafeState – represents a unsafe failure state (either the failure is not detected at all or it is detected but has been handled correctly) and the AdditionalUnsafeState (un-

safe operation due to calling upon the safe state, which in fact has become unsafe due to a successful tampering with of the safe state of the e-Motor device). The "stochastic activities" model the distribution of time a transition between two states takes. For instance, the activity "OK_Compromised" represents a transition (and the probability distribution of the duration of this transition) from OK state to the CompromisedState. Similarly, the activity "Compromised_OK" represents a transition to OK state from CompromisedState.

"CompromisedState" state models the fact that, due to a data integrity violation, the *safe state* of an e-Motor channel has been altered. The safe state definition (code and data) is stored in *safe state configuration file*. One of the configuration files is *active* at a time and should the e-Motor fail to perform correctly, the active safe state will be invoked by the higher authority logic built in the e-Motor. An alteration of the e-Motor's safe state will place the channel in a ***compromised state***, i.e. will create a ***new hazard*** and will have no consequence until the device needs to enter its safe state. Should the safe state be called upon while the device is in the "compromised" state, then the e-Motor instead of entering the designated safe state will enter an unsafe state (AdditionalUnsafeState), as shown in **Fig. 1.**

If the incorrect (e.g. the tampered with) safe state configuration file is detected before a transition to the "safe" state is called upon and there is a mechanism to restore the correct configuration (i.e. the compromised state is 'fixed' by a return to the state OK), then the "compromised" state will have no effect on device safety.

### 3.1 Simplifications Made

*Type of attacks.* The model considers three types of attacks as listed below of which only the first type is discussed in detail in the paper[2]:

1. A malicious alteration of the safe state configuration files.
2. A malicious modification of the requested torque (information coming to the e-Motor as an input, i.e. over the CAN Bus).
3. A malicious modification of the channels' control loop parameters. These do not affect immediately the device's safe operation, but may cause a channel failure.

*Common cause accidental failures are modeled simplistically.* It is based on the Marshall and Olkin model [10] of common stress.

*Limited knowledge about stochastic properties of the modeled cyber* attacks. As discussed in the introduction this limitation is not specific to the e-Motor case study and applies to most attempts to model stochastically cyber attacks.

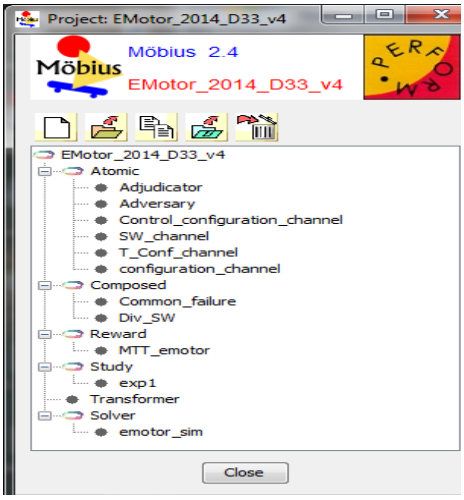
*Detection of attacks.* The SAN model includes several techniques of attack detection: i) a generic attack detection capability (represented simply by a probability of detecting an attack), ii) plausibility checks on the values of torque requests. Plausibility checks are among the mechanisms considered by the vendors for the real e-Motor device, and iii) timing checks, e.g. torque requests should not occur more frequently than a predefined timing constraints. This paper, however, uses only the first of the three detection techniques.

[2] This attack type was missed in the initial safety analysis.

### 3.2 The SAN Model

**Model Description.**

The Mobius (SAN) project is shown in **Fig. 2**.




The project consists of several atomic models: Adjudicator, Adversary, Control_configuration_channel, SW_channel, T_config_channel and configuration_channel.

The model also includes a composed model, Div_SW.

The Reward, Study and Solver components are parts of project, too.

The model is solved via Monte Carlo simulation.

**Fig. 2.** The SAN project, E-Motor_2014_D33.

**The Composed Model, Div_SW.**

**Fig. 3** presents the composed model of the e-Motor**.**

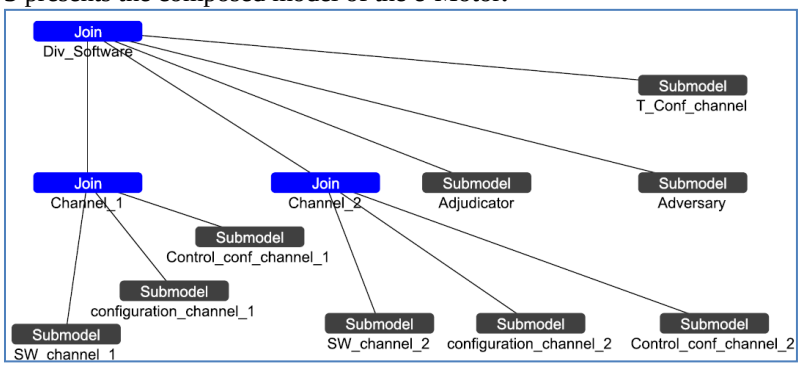



**Fig. 3.** The structure of the composed model, Div_SW

It consists of a number of other models:

- Channel 1 – represents the behavior of the first s/w channel;
- Channel 2 – represents the behavior of the second s/w channel;
- Adjudicator – offers the functionality, necessary to deal with the redundant channels. It implements a 1-out-of-2 architecture with two self-checking channels.
- Adversary – models the behavior of an adversary who might attempt one of 3 possible attacks described above. The adversary model allows for implementing a sin-

gle attack or multiple attacks.

Each of the software channels is represented by a Join in the composed model Div_SW. These are Channel_1 and Channel_2, respectively. Each of these, in turn, is a composition of three atomic models:

- SW channel – models the behavior of a software channel with respect to accidental failures. This model is substantially similar to the model shown in Fig. **1.** with various additions such as repair of the channel when it is in either safe or unsafe failure state provided the system is OK;
- Configuration_channel_1/2 – models the possible alterations of the safe state configurations (files) used by the respective s/w channel. The SW_channel and the respective configuration have *shared places* (one of the Mobius mechanism used for linking various atomic/composed models) defined, so that when the configuration file is altered and the sw_channel is instructed by the higher authority to enter a safe state, it will instead enter an unsafe state.
- The atomic models, T_Conf_channel and Control_conf_channel_1/2, model a torque attack and the attack on the control parameters of the respective s/w channels. Neither is discussed in detail in this paper.

The many models listed above communicate with each other via the mechanism of *shared places*. The interested reader can get further details about the model from the author (including the SAN model itself).

Using pairs of configuration files in the model (safe states and control_conf, respectively) – one per channel – is dictated by the requirement in ISO 26262 to eliminate any *single point of failure* in the e-Motor design.

**The Atomic Models**

*SW_channel.* The model is shown in **Fig. 4.**

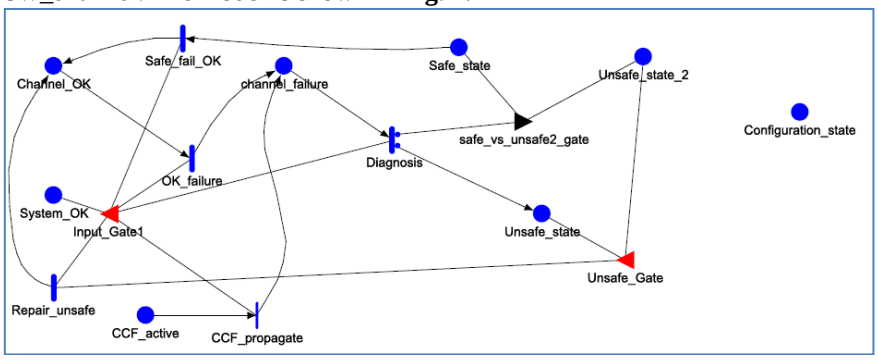



**Fig. 4.** Model of a SW channel

This model includes a model of common failures, captured by the place CCF_active (a shared place for both SW_channel atomic models) and the activity CCF_propagate. The common failure is enabled only when both channels are OK (Input_Gate). The place CCF_active is shared with the respective place in the atomic

model Adjudicator and triggers a common failure event. The intervals between the simultaneous failures are specified in the Adjudicator atomic model. More specifically, the model offers four distinct states for a software channel as defined in **Fig. 1** above: Channel_OK, Channel_failure, Safe_state – this is the state where the channel will end up, if the diagnosis determines successfully the need for a safe state AND the safe state at this moment in time is not compromised. If, however, the safe state is compromised, then the channel will end up in Unsafe_state_2; the decision logic which state to move to is captured by the output gate safe_vs_unsafe2_gate. The probabilities of successful diagnosis are defined explicitly as global variables.

From Safe_state a channel can be returned back to Channel_OK. Similarly, a channel after unsafe failure (Unsafe_state or Unsafe_state_2) can be repaired. Both repairs are conditional on the system being in OK state, i.e. at least one of the channels is OK, achieved via the place System_OK and the input gate, Input_Gate_1.

*Adversary*. This model is shown in **Fig. 5**.

It consists of several places: Start, where one can place a number of tokens to simulate attacks. The number of tokens determines how many attacks will be simulated. Each attack (a token being passed from Start place to either ConfAttack_in_progress, Torque_attack_in_progress or Control_Attack_in_progress will trigger a scenario of an attack of one of the three types. For the attacks on the safe state the scenarios may involve attacking a single channel or attacking both channels. Which of the options will be used in a study is controlled by the token in Single_both_enabler place (and the two input gates linked to this place). The other two attacks – on the torque and on the control loop parameters – are not discussed in this paper.

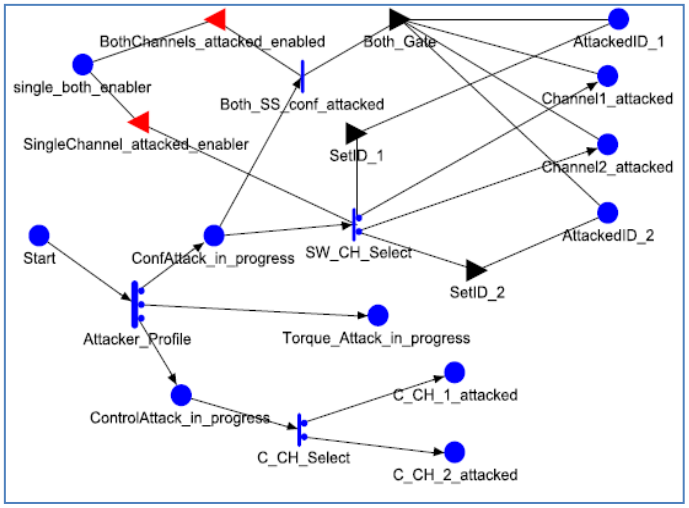



**Fig. 5.** A SAN model of the adversary

*Configuration_channel*. This model is shown in **Fig. 6.** A state model of a channel's safe state configuration file. This model defines the states associated with the

safe states as follows:

- Correct (default) – the safe state is not tampered with.
- Incorrect_1, Incorrect_2 – model two other states for the safe states, both valid as safe states for other contexts (i.e. modes of operation of a SW channel), but inadequate for the currently active mode of operation of the e-Motor. The use of two states to model the consequences of tampering with the safe state configuration file is motivated by the description of the device provided by the vendor in [8].

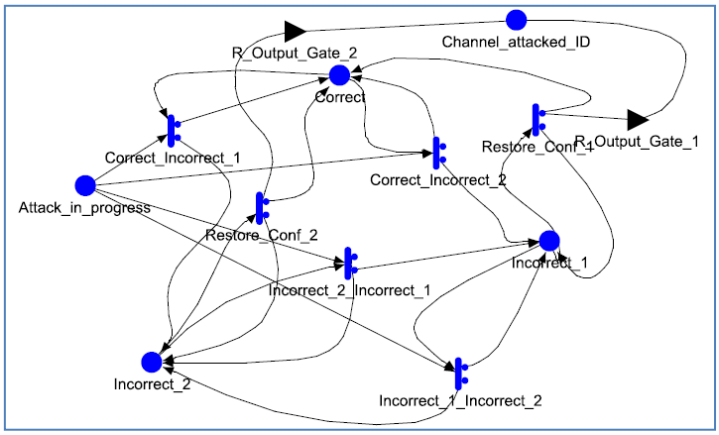



**Fig. 6.** A state model of a channel's safe state configuration file.

- The transitions between the states of the safe state configuration file are governed by the activities in this atomic model (Correct_Incorrect_1, etc. including restoring from incorrect states to the correct one, Restore_conf_1), which are enabled by the place Attack_in_progress enabler. The transitions between states (i.e. tokens between places) are only possible when Attack_in_progress place contains a token. This place is shared with the places, Channel1_attacked and Channel2_attacked, respectively, of the Adversary model. In other words, only when an attacker has chosen which channel to attack (the Adversary can choose to attack either a single channel or both simultaneously) is a transition of the respective configuration_channel enabled.
- Channel_attacked_ID place is shared with the places AttackedID_X of the Adversary model. The tokens in it are used in the predicates of the activities, which drive the transitions of tokens between the other places of this model.

*Adjudicator.* The model is omitted here for lack of space, but is described in [11]. It implements the logic summarized above and uses the mechanisms of shared places.

**Studies**

Sensitivity analysis was completed to establish how model behavior is affected by the values of model parameters documented in [11]. Here a small sample is presented to illustrate the work and comment on some outcomes which at first looked surprising.

Sensitivity analysis has been applied to a number of parameters. In the model these parameters are defined as *global variables*. Global variables can be used for defining the properties of activities (e.g. the parameters of the probability distributions used) and to define the initial marking of the places used in the SAN models. SAN allows for defining "studies". A study in SAN terms corresponds to a particular set of values assigned to global parameters. Further details are provided in [11].

## 4 Findings

A number of studies have been completed via Monte Carlo simulation. Full details are available from the author. The results consist of the probabilities of e-Motor (as a system) being in one of the 3 possible system states – "Success" (mission completed without a failure), "Safe failure" (device moved to safe state) and "Unsafe failure" (failure occurred, but the device failed to reach safe state) – at the end of *missions of fixed lengths*. The results are provided for missions of length in the range of 1000...12000 hours (slightly over a month to more than an year). These were derived from simulation traces of length 200,000 hours, i.e. 22 years of operation, likely to exceed the typical lifetime of a car.

An example of sensitivity analysis is provided in **Table 1**.

**Table 1.** A sensitivity analysis example.

| Global variable name | Exp 1 | Exp 2 | Exp 3 | Exp 4 |
|---|---|---|---|---|
| AttackRate | 0.001 | | | |
| CC_failure_rate | 1.00E-04 | | | |
| Config_repair_success | 0.6 | | | |
| SS_repair_rate | 36 | 360 | 3.6 | 36 |
| USF_repair_rate | 36 | 36 | 36 | 360 |
| attack_CH1_success_pr | 0.2 | | | |
| attack_CH2_success_pr | 0.1 | | | |
| attack_count | 10 | | | |
| channel_failure_rate | 0.001 | | | |
| failure_coverage | 0.8 | | | |

The four studies are parameterized identically except for the values of two parameters, **SS_repair_rate** and **USF_repair_rate**, the rates of repair of a channel from safe and unsafe failure, respectively. Common accidental failure is allowed with rate an order of magnitude lower than the rate of channel failure. Accidental failures are assumed to have the same rate as the attacks on the safe state. All probabilities of success (of attacks, of repairs, restoring the safe state and of failure detection) used in the studies are provided in **Table 1**.

The model behavior under the four parameterizations is summarized in **Table 2** in which the probabilities of success, safe failure and unsafe failure at the end of missions of fixed length are computed.

Note that the sum of the probabilities shown in **Table 2** per selected mission duration equal to 1 as each individual mission will end up in one and only one of the three

alternatives.

**Table 2.** Effect of repair time rates on the probabilities of how missions of fixed length will end (Success, Safe failure or Unsafe failure).

| | | Probability of mission Success/Safe failure/Unsafe failure | | | |
|---|---|---|---|---|---|
| **Mission duration [hours]** | **Probability of mission ending in** | **Exp 1** | **Exp 2** | **Exp 3** | **Exp 4** |
| 1000 | Success | 0.999120 | 0.999245 | 0.999660 | 0.999175 |
| | Safe failure | 0.000880 | 0.000755 | 0.000340 | 0.000825 |
| | Unsafe failure | 0 | 0 | 0 | 0 |
| 2000 | Success | 0.998290 | 0.998415 | 0.009525 | 0.998405 |
| | Safe failure | 0.001710 | 0.001585 | 0.990475 | 0.001595 |
| | Unsafe failure | 0 | 0 | 0 | 0 |
| 7000 | Success | 0.994180 | 0.995060 | 0.008180 | 0.995015 |
| | Safe failure | 0.005815 | 0.004940 | 0.991820 | 0.004985 |
| | Unsafe failure | 5.00E-06 | 0 | 0 | 0 |
| 8000 | Success | 0.993410 | 0.019340 | 0.007930 | 0.019285 |
| | Safe failure | 0.006585 | 0.980660 | 0.992070 | 0.980715 |
| | Unsafe failure | 5.00E-06 | 0 | 0 | 0 |
| 11,000 | Success | 0.991455 | 0.017560 | 0.007290 | 0.017650 |
| | Safe failure | 0.008540 | 0.982440 | 0.992710 | 0.982350 |
| | Unsafe failure | 5.00E-06 | 0 | 0 | 0 |
| 12,000 | Success | 0.020310 | 0.017005 | 0.007100 | 0.017070 |
| | Safe failure | 0.979685 | 0.982995 | 0.992900 | 0.982930 |
| | Unsafe failure | 5.00E-06 | 0 | 0 | 0 |

## 5 Discussion

The studies have been chosen so that one can systematically trace the impact of a single parameter on the behavior of the model. It is somewhat surprising that when the rates of repair from both safe state and unsafe state of a channel are the *same* (Exp 1) the mission is likely to survive longest without a failure. The probability of failure of the mission is very low until missions of 12,000 hours. Increasing one of the repair rates by an order of magnitude (Exp 2 or Exp 4, respectively) does not improve the chances of mission survival. For these two experiments the probability of a mission failure drops dramatically at around 8,000 hours. At first this may seem counterintuitive. Fast recovery must be a good thing. But somehow the improvement of the repair rates *asymmetrically* does not improve the mission chances to survive without a failure. The good news is that this drop is due to safe failures, which one would consider acceptable, although availability is reduced.

Looking at the probability of unsafe failure, we notice that for both Exp 2 and Exp 4 the probability of unsafe system failure the first 12,000 hours is 0. In fact after in-

specting the entire distribution (up to the simulated 200,000 hours per simulation run) we observed that no unsafe failure was recorded for Exp 4 at all. For Exp 2 the first mission length for which unsafe system failure was recorded was 24,000 hours[3]. For Exp 1, on the other hand, we did observed unsafe system failures starting from mission duration of 5,000 hours. In other words, the improvement of the repairs seems to indicate that the chances of unsafe failures are reduced, which is what one would expect/want.

Exp 3 is not surprising. Reducing the repair rate for channels in safe failed state does reduce the chances of a mission survival. Missions without a failure are very unlikely for missions longer than 2000 hours. On the other hand, unsafe failures were first recorded for missions of 13,000 hours. This is better than for Exp 1, but the confidence in the computed probabilities is very low for us to draw any conclusions.

The dependent accidental failures of the two channels are modeled using the model of common stress of Marshal and Olkin [10]. It is worth pointing out, that while the adequacy of the model for software failure may be questioned, it clearly models adequately hardware failures – both channels are executed on the same hardware. The model of common mode can be replaced by suitable alternatives, e.g. [12], [13] or [14] and may impact availability .

## 6 Conclusions and Future Work

In this paper an approach to stochastic modeling of a safety critical device is presented which accounts for both – accidental failures and cyber attacks affecting safety. The study is centered upon one particular attack type: an attack which may lead to eliminating the safe state of the device. The device being specified as an ASIL-D device must be built using design diversity and the model accounts for the two-channel architecture of the device.

The paper demonstrates that probabilistic modeling may be useful in quantifying the risk from cyber attacks. The approach advocated in the paper is that a probabilistic model should be detailed enough to account for *all hazards* identified in the safety analysis (in the presented model three hazards have been included and one of them studied in detail) while the epistemic uncertainty with model parameters (i.e. in deciding the values of the parameters related to the attacks) should be addressed by sensitivity analysis exploring the space of *plausible* parameters.

For lack of space the paper could not demonstrate sensitivity analysis more extensively, e.g. varying the rates and probabilities of success of the attacks; the impact of these on model behavior will be studied in the future:

- The model of accidental dependent failures is simplistic, possibly unrealistic and in the future will be replaced with alternatives;
- In the particular context of AUTOSAR an interesting design trade-off exists between protecting the individual devices against cyber attacks specific to a device

[3] For all probabilities lower that $10^{-4}$ the relative confidence in the particular number was lower than 10%. Thus, for small probabilities (including values of 0) the numbers should be treated as statistically insignificant.

vs. using a generic intrusion detection/protection system (IDS). Probabilistic modeling seems particularly suitable for addressing this problem and we intend to study the trade-off it in the future.

## Acknowledgement

The work was supported by the Artemis JU SESAMO project (grant agreement number 295354). The author would like to thank the anonymous reviewers and Dr Kizito Salako for their thorough reviews of earlier drafts of the paper.

## References

1. Zetter, K., Countdown to Zero Day: Stuxnet and the Lunch of the World's First Digital Weapon. 2014, New York: Crown Publishers.
2. Kriaa, S., M. Bouissou, and L. Pietre-Cambacedes. Modeling the Stuxnet Attack with BDMP: Towards More Formal Risk Assessments. in 7th International Conference on Risk and Security of Internet and Systems (CRiSIS). 2012 Cork, Ireland: IEEE.
3. Ford, M.D., et al. Implementing the ADVISE security modeling formalism in Möbius. in The 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). 2013. Budapest, Hungary: IEEE.
4. Wang, L., et al., *k-Zero Day Safety: A Network Security Metric for Measuring the Risk of Unknown Vulnerabilities.* IEEE Transactions on Dependable And Secure Computing, 2014. **11**(1): p. 30-44.
5. Ten, C.-W., C.-C. Liu, and G. Manimaran, *Vulnerability Assessment of Cybersecurity for SCADA Systems.* IEEE Transactions on Power Systems, 2008. **23**(4): p. 1836-1846.
6. Netkachov, O., P.T. Popov, and K. Salako. Model-based Evaluation of the Resilience of Critical Infrastructures under Cyber Attacks. in The 9th International Conference on Critical Information Infrastructures Security (CRITIS). 2014. Limassol, Cyprus: Springer.
7. Taleb, N.N., The Black Swan: The Impact of the Highly Improbable. 2008: Penguin. 394.
8. SESAMO, *Use Case Specification (Deliverable D1.2)*, 2013, SESAMO Consortium. p. 105.
9. SESAMO, *E-motor* 2014, http://sesamo-project.eu/content/e-motor.
10. Marshall, A.W. and I. Olkin, *A generalised bivariate exponential distribution.* Journal of Applied Probability, 1967. **4**: p. 291-302.
11. SESAMO, Integration of Safety and Security Analysis and Assessment Techniques (Deliverable D3.3), 2013, SESAMO Consortium. p. 250.
12. Ammann, P.E. and J.C. Knight, *Data Diversity: An Approach to Software Fault Tolerance.* IEEE Transactions on Computers, 1988. **C-37**(4): p. 418-425.
13. Bondavalli, A., et al., *Modelling the effects of input correlation in iterative software.* Reliability Engineering and System Safety, 1997. **57**(3): p. 189-202.
14. Popov, P. and G. Manno. The Effect of Correlated Failure Rates on the Reliability of Continuous Time 1-out-of-2 Software. in 30th International Conference on Computer Safety, Reliability, and Security (SAFECOMP 2011). 2011. Naples: Springer.