# RISKEE: A Risk-Tree Based Method for Assessing Risk in Cyber Security

Michael Krisper[1](✉), Jürgen Dobaj[1], Georg Macher[1], and Christoph Schmittner[2]

[1] Graz University of Technology, 8010 Graz, Austria
{michael.krisper,juergen.dobaj,georg.macher}@tugraz.at
[2] AIT Austrian Institute of Technology, 1020 Vienna, Austria
christoph.schmittner@ait.ac.at

**Abstract.** In this paper, the RISKEE method for evaluating risk in cyber security is described. RISKEE is based on attack graphs and the Diamond model combined with the FAIR method for assessing and calculating risk. It can be used to determine the risks of cyber-security attacks as a basis for decision-making. It works by forwarding estimations of attack frequencies and probabilities over an attack graph, calculating the risk at impact nodes with Monte-Carlo simulation, and propagating the resulting risk backward again. The method can be applied throughout all development phases and even be refined at runtime of a system. It involves system analysts, cyber security experts as well as domain experts for judgement of the attack frequencies, system vulnerabilities, and loss magnitudes.

**Keywords:** Risk assessment · Risk propagation · Attack trees · FAIR method · Diamond model · Cyber physical security · IT-security

## 1 Introduction

In earlier work, we established the idea of combining existing methods for safety and security for the automotive and industrial domain in a quantitative way to come up with a fully integrated quantitative risk assessment [9]. During working on that topic, we stumbled upon several problems with existing methods and we are now on a pursuit of solving them.

Risk is the notion of an event which may occur in the future and which may have negative outcomes (for positive outcomes it is called *opportunity*). Classically this can be expressed in mathematical terms like this:

$$Risk = Probability \times Impact \tag{1}$$

*Probability* is a number between 0 and 1 (0% to 100%) and *Impact* is a number denoting a quantitative measure of the loss if the event occurs (e.g. 1000$). This impact could be actual monetary loss, where you have to pay some money or

A. Walker et al. (Eds.): EuroSPI 2019, CCIS 1060, pp. 45–56, 2019.
https://doi.org/10.1007/978-3-030-28005-5_4

replace some device, but it also could be loss in the form of reduced income or revenue. The calculation of risk helps to compare and evaluate the events and furthermore to make decisions based on that evaluation. By knowing the total risk of a project, one could reserve enough financial resources to overcome expected losses (recovery), or try to decrease high risk events down to a tolerable level by lowering the probability or decreasing the impact (prevention). It is important to note that risk represents only the *expected* amount of loss, which is an artificial value, because it is derived in a probabilistic way. This value represents the expected average which, due to the law of large numbers, is only realistically accurate in the case of high sample sizes. For small samples sizes this estimation could be completely wrong. This is due to the fact that traditionally risk estimates are just point estimates which do not take into account uncertainty, sample size and confidence. If we would judge the probability and impact in form of a range of values which also includes the uncertainty, we could depict the resulting plausible range for the risk, which even for small samples sizes could give us an estimate with high confidence. This is the basic idea of this papers' contribution, the RISKEE method for risk assessment using attack trees and probability distributions.

Beside neglection of uncertainty, another problem is the usage of ordinal scales [17,28], especially in areas which are difficult to quantify. For example, in safety, risk is not measured in monetary values, but instead with harm or danger to human life, which is much more difficult to measure. Because of that, often ordinal scales are used which define increasing levels of injury or harm. Methods which use such ordinal or even nominal scales are called qualitative or semi-quantitative methods. These levels and thresholds of ordinal scales have several drawbacks, e.g. they are often completely arbitrary, introduce quantization errors, or are ambiguous [6–8], but nevertheless they are commonly used because they seem simple to understand, to use, and to evaluate, although this may be just a perceived impression of benefits, which cannot be proven in reality [14,33]. But whats even worse: Ordinal scales don't allow arithmetic operations. They allow ordering relations like equal, smaller, or larger, but addition or multiplication are not defined. Think of multiplying two t-shirt sizes: x-large * small. Is this reasonable? No. The size of t-shirts is just one example for an ordinal scale, it allows for assessments of smaller or larger, but nothing more. Bizarrely enough, for risk assessment we have no reluctance of multiplying two ordinal scales together. For example in the failure mode and risk analysis (FMEA) [15], the input values for severity going from 1 (none) to 10 (hazardous without warning) are multiplied with the occurence going from 1 (<0.001% of cases) to 10 (>10% of cases). The result is called risk priority number (RPN) and the risks are prioritized according to this number. This has been proven to be wrong and inaccurate many times over [2,3,11,13,27].

Many existing methods for analysis of security and safety use such qualitative judgements to evaluate the risk of a security breach, or the risk of danger and harm to human life. These methods use expert judgements based on arbitrary quasi-quantitative ordinal scales to judge values like e.g. exposure, severity, knowledge, resources, criticality and so on. This results in an overly simplified and rough classification, which is to unprecise and error-prone. That is why we began working on a method which used real frequencies, probabilities and impacts to evaluate the risk of a system for its cyber-physical-security, but also safety. We call this method RISKEE, and our intention was to develop an easy to use method to evaluate risk in attack-trees in a quantitative way.

The remainder of this paper is structured as follow: In Sect. 2 the background of the work is presented and related work including its shortcomings are discussed. The contribution of this paper is shown in Sect. 3. Section 4 discusses the limitations and current challenges and concludes the paper.

## 2 Background and Related Work

In this section we describe the background as well as related work for the proposed method. First, we shortly summarize our earlier work on that topic, then we shortly dive into a comparison of existing methods. Afterwards we describe the Diamond model of Intrusion Analysis, and the FAIR method for risk assessment.

### 2.1 Towards Unified Quantitative Risk Assessment of Safety and Security

In our previous work [9], we connected established methods for safety and security assessment (namely SAHARA [19] and FMVEA [25]) to create an informed knowledge base in form of the Diamond model [4] for evaluating the risk using the FAIR method [30,32]. While this work was important for our understanding, it also opened up many questions and showed problems in the existing methods. One of the results was, that methods primarily focus on the attacker side, and neglect the victim, which is reasonable since those methods based on threat modelling, which emphasizes threats, not defenses. Due to focusing reasons, we will not tackle this in the current paper, but we have it on our todo list and will be solved in future work.

In the following paragraphs, we repeat the fundamentals of some established methods for risk assessment in safety and security for the following methods: SAHARA, FMVEA, and ATA. We considered them, because they are the proposed methods by an analysis of state-of-the-art methods for integrated security, safety and reliability engineering by Macher et al. [18].

The first method is SAHARA [19], which is based on HARA (Hazard and Risk Analysis [16]), and extends it by using STRIDE [22] to find the security attack vectors. The attack vectors are evaluated on an ordinal scale according to the required resources, know-how and threat criticality, which are combined to a security level according to an evaluation matrix. If the resulting security level exceeds a specific threshold, the attack vector is considered for further safety analysis in the HARA.

The second method is FMVEA [25], which is based on FMEA [15] and again extends it by using STRIDE [22] to find additional failure modes which are caused by security attacks. Through the description of vulnerability and threat agent, the threat probability, severity, and furthermore the criticality, can be determined on an ordinal scale. The criticality can be used for prioritization to find the most critical failure and threat modes which have to be mitigated.

The third method, ATA [1,34] or attack tree analysis, is of special interest for us because it is a graphical model based on attack trees [26], which are used in RISKEE. It has its origins in safety, especially fault-trees and fault-tree analysis (FTA) [20,35,37]. In ATA, the events are not a simple list, but they are arranged in a tree structure where the root node is the attacker's goal, and the leaves are the steps which are needed to reach this goal. With every layer of the tree the steps get more detailed and refined. The nodes in the attack tree can have specific attributes which are needed to analyze the tree for e.g. the most feasible or dangerous attack paths. The idea of graphical representations of attack paths to analyze cyber security attacks over some given infrastructure was extended over the years to cover whole attack graphs, bayesian networks, belief propagation, markov chains, and petri nets, and many others. For example, Poolsappasit et al. use an attack graph to implement bayesian belief propagation [23] and apply genetic optimization algorithms to calculate pareto-optimal combinations of mitigation techniques.

## 2.2 FAIR

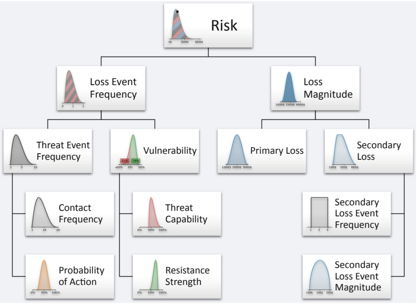


**Fig. 1.** The FAIR ontology with its subfactors (reused from [9]).

In FAIR, risk is decomposed into several subfactors, which can be evaluated more easily. It establishes a whole ontology of these subfactors which are mathematically related in a precise way. To estimate these subfactors, expert judgement and historical data is used, but always including the respective uncertainty or confidence in the data. Therefore, the judgements are given as value ranges, or probability distributions which represent the likelihood of values for the respective subfactor. Figure 1 shows the FAIR ontology and its subfactors as well as the according estimations in form of probability distributions. The modified PERT-beta distribution [24,36] is then used to model these expert judgements by using the parameters: minimum, maximum, most likely value and confidence.

By using monte carlo simulations [21] and the mathematical relations of the subfactors, FAIR can calculate the probabilities and likelihoods for a range of risk values. The result is a loss exceedance curve, which depicts the outcomes and respective probabilities thereof. This can be compared to the risk appetite curve, in order to decide if the risk is tolerable or not. For further information we politely guide you to the standards Risk Taxonomy [32] and Risk Analysis [31] by the Open Group, or the respective book by the creators Jones and Freund [12].

## 3 Contribution: The RISKEE Method

RISKEE (coined from Risk-Tree) is a method for risk assessment and evaluation, which is based on attack-trees/attack-graphs with special emphasis on risk. It works by building a graph of consecutive attack events, judging the frequencies, vulnerabilities and impact magnitudes of events, and calculating a distribution of risks based on that. The initial attack frequencies are carried forward over the node's vulnerabilities until the end nodes. There, the losses are realized and propagated backward again to calculate the respective *virtual risk* for all individual nodes (Fig. 2).

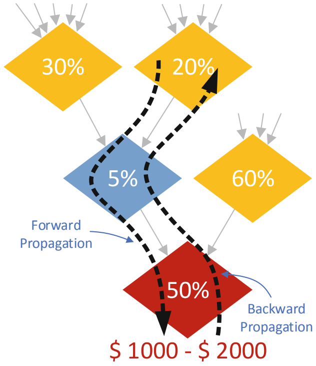


**Fig. 2.** A risk-tree showing forward propagation of attacks and backward propagation of risk in RISKEE.

The attack graph depicts different attack paths an attacker must take to reach some goal. The edges define the consecutive order of these events, one step after another. Such attack paths may intersect and split up again when attackers have several possibilities to choose from. Nodes have some necessary attributes which must be defined, to make risk evaluation possible. These values are frequency, vulnerability and magnitude, and can be defined via historical data or expert judgement. Important here is, that they should be given as distribution and ranges which consider the respective uncertainty, not only as single values. The first nodes on an attack path form the attack surface, which is subject to a permanent bombardment of attacks. Here the frequency of attacks and the respective vulnerability (probability of an attack going through) have to be defined. The attack continues then with the intermediate nodes, which only need a vulnerability rating, but already may involve impact. In the end, the last nodes represent the actual goal, involving the actual losses. When an attacker reaches them, the loss is realised.

### 3.1 Structure and Framework

A risk tree consists of three different types of nodes which are connected and form the individual attack paths: entry events, intermediate events, and goal events. The events are described using the attributes frequency, vulnerability, and impact.

**Types of Events.** All events in RISKEE represent attack events which form a path to a specific goal for the adversary. In the simplest case this is only one event representing attack surface and goal, but in real cases the attack graph consists of multiple events and paths.

- **Entry Events** represent the attack surface. Every attack starts with these nodes (regardless if it is an external or internal attack). They are also the only ones defining the frequency of attacks.
- **Intermediate Events** are events which have to be passed through in order to get to the goal events. To achieve their goal, adversaries have to go through these events. Intermediate events are described mainly by the vulnerability.
- **Goal Events** are events on the end of an attack path, which cause the most loss or harm to the victim. The goal events represent the last event of an attack path (a so called "sink"). The most important attribute for a goal event is the impact magnitude.

**Types of Attributes.** The events are described using three attributes: *frequency*, *vulnerability*, and *impact*, which are used to calculate the resulting *risk*:

- **Frequency:** The estimation of the number of events over a specific time span (e.g. 4–6 times per year, or 80–100 times a day).
- **Vulnerability:** Is the probability that a threat event will become a loss event. Or stated in other words: that an attack is successful. The vulnerability depicts the difference in strength, between the adversary and victim - like pulling on a rope from two sides. It is the difference of the respective estimations for the Threat Capability (adversary-side) and Resistance Strength (victim-side). To be comparable, both estimates must have the same scale within the event which gets estimated. A good proposal is to use the distribution of overall threat population for the scope of the event as a scale.
    - *Resistance Strength* is the rating of the defender. It defines how well the analysed system is protected against attacks. The scale for this should be the overall threat population to evaluate which portion of attacks the system can withstand.
    - *Threat Capability* is the rating of the attacker. It determines the capabilities, resources and knowledge of the assumed attackers compared against the overall threat population.
- **Impact:** The range of impacts an event can have, when it actually occurs (e.g. $1000–$2000, but most likely around $1100).
- **Risk:** Represents a whole range of estimated outcomes for future events, together with their likelihoods. This is not a single value, but a distribution over probable outcomes. It is the result of the calculations and is most accurately visualized with a loss exceedance curve (LEC).

### 3.2 Using RISKEE

RISKEE can be embedded into the development process already very early on, at architectural phase or at design phase and can be refined during development as well as later on during runtime. We propose to use the Diamond model [4] as a framework to define the attributes in order to judge frequency, vulnerability, and magnitude later on. While it would also be possible to judge them directly, it would not be as comprehensible as using a rigorous formal model like the Diamond model. If data is available from existing methods like SAHARA or FMVEA, it can be used to fill the Diamond attributes [9]. We use expert judgements to determine the attributes. The expert group should be composed of three to five people [10] from different domains [5] e.g. cyber security experts, infrastructure experts, domain/field experts, system analysts, or system architects. The judgements are combined via a linear opinion pool (arithmetic average) [29]. A risk tree is created by identifying the necessary steps for an attacker to achieve a goal and attributing them with frequency (how often does this attack occur), vulnerability[1] (how likely is the attack to be successful), and magnitude (what is the impact of an successful attack event). The nodes represent attack events and the edges resemble the order in which they can occur. The result is calculated via applying the RISKEE PROPAGATION ALGORITHM (Algorithm 1) and cumulated via a so called Loss-Exceedance-Curve (LEC). This curve depicts the probability of exceeding certain amounts of money over the whole range. With the LEC as basis, management can judge if the possible risks are tolerable, or still to high (which is called the risk appetite). Figure 3(b) in Sect. 3.5 shows an example for a loss-exceedance curve.

### 3.3 Process

To give a step-by-step guidance, here the complete process using the RISKEE method is described:

- Step 1: Create the attack graph.
- Step 2: Estimate the attributes for the events:
  - Entry Events: **Frequency**, *(Vulnerability, Magnitude)*
  - Intermediate Events: **Vulnerability**, *(Magnitude)*
  - Goal Events: **Magnitude**, *(Vulnerability)*
- Step 3: Calculate risk with the RISKEE Propagation Algorithm (see Algorithm 1).
- Step 4: Make decision based on the loss exceedance curve for the risk or enact further mitigation steps to reduce risk.

### 3.4 Calculation of Risk

The calculation of risk is done with the RISKEE PROPAGATION ALGORITHM (see Algorithm 1). All operations are done with probability distributions coming from the factor analysis of information risk FAIR analysis. It basically consists of the following steps:

[1] Vulnerability can also be judged indirectly in form of resistance strength and threat capability. These two estimations are subtracted by RISKEE to get the percentage of cases where an attack would be successful.

1. Split up the risk-graph into all distinct individual paths from all entry nodes to nodes with a defined loss magnitude (mostly goal nodes, but others could also have a defined magnitude).
2. For every path: take the attack frequencies of the input node and propagate it forward over the intermediate nodes (multiplied with the respective vulnerability) until the goal node is reached. Also accumulate any impacts on the way.
3. Calculate cumulative Risk in the goal node by multiplying the resulting frequency with the impact.
4. Apply this risk to all nodes and edges on the current path (sum up if they already contain existing risk-values).

**Algorithm 1.** Riskee Propagation Algorithm

```
procedure RiskeePropagate($G$)                                  ▷ G... Risk Graph
    for all $path \in$ Paths($G$) do
        $frequency \leftarrow frequency_{entry}$
        $magnitude \leftarrow 0$
        for all $node \in$ Nodes($path$) do                     ▷ Propagate Forward
            $frequency \leftarrow frequency * vulnerability_{node}$
            $magnitude \leftarrow magnitude + magnitude_{node}$
        end for
        $risk \leftarrow frequency * magnitude$
        for all $edge \in$ Edges($path$) do                     ▷ Propagate Backwards to Edges
            $risk_{edge} \leftarrow risk_{edge} + risk$
        end for
        for all $node \in$ Nodes($path$) do                     ▷ Propagate Backwards to Nodes
            $risk_{node} \leftarrow risk_{node} + risk$
        end for
    end for
end procedure
```

### 3.5 Computations on Probability Distributions

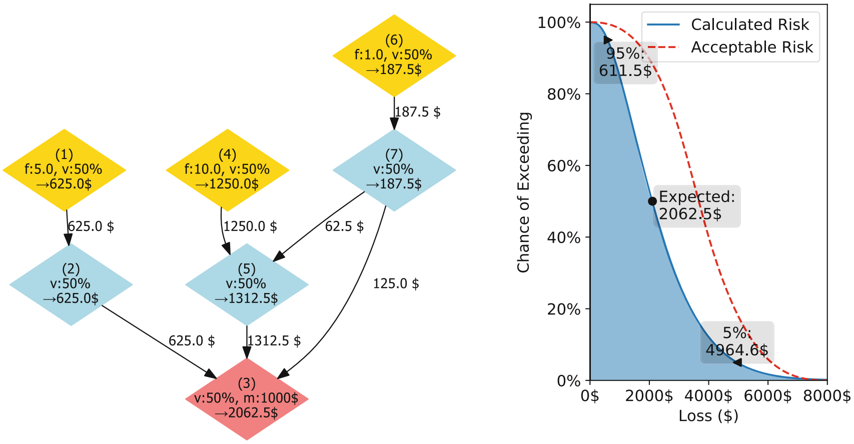
The Riskee Propagation Algorithm computes values based on probability distributions. We use monte-carlo simulation to calculate the mathematical operations on the probability distribution. This works by sampling many values from the distributions, executing the operations and in the end create a histogram over the results. We use PPS-sampling[2] via inverse-transformations of the cumulative probability density functions for unrelated values, and for related or conditional distributions we use simple random sampling. In both cases the calculation is finished by computing the histogram over the results and smoothing it with bounded kernel density estimation using gaussian kernels to get a continuous probability distribution function for further usage. We apply the smoothing to

[2] Probability-Proportional-Size-Sampling; A stratified sampling strategy.

avoid aggregation of quantification errors. In this way we can assume to have continuous functions for every further operation. For the PPS-sampling we use a fixed number of percentiles over the distributions (2500 percentiles), and for the random sampling we use an adaptive algorithm which stops on convergence ($\varepsilon < 0.05\%$), which happens mostly after about 20000 to 50000 samples.

**Example:** Here we feature an example for applying RISKEE on a simple attack-tree together with judgements of the input values and presenting the resulting risk. Figure 3 shows the risk tree and the result. In this example we showcase an attack tree consisting of seven events (enumerated from (1) to (7)). For easier demonstration it uses 50% vulnerability for all events, and only has one goal event with the magnitude of 1000$. The entry events were event (1) with a frequency of 5 times/year, event (4) with a frequency of 10 times/year, and event (6) which occurs once per year. The resulting paths in this example graph are as follows:

- 5 × [(1) → (2) → (3)] × 1000$, resulting in an expected risk of 625$
- 10 × [(4) → (5) → (3)] × 1000$, resulting in an expected risk of 1250$
- 1 × [(6) → (7) → (3)] × 1000$, resulting in an expected risk of 62.5$
- 1 × [(6) → (7) → (5) → (3)] × 1000$, resulting in an expected risk of 125$.



(a) An example for a risk tree, showing the different possible attack paths (f=frequency, v=vulnerability, m=magnitude, the resulting risk is displayed after the →).

(b) An example for a loss exceedance curve (LEC). It shows the probability of exceeding certain magnitudes of losses.

**Fig. 3.** Applied example for a risk assessment with RISKEE.

Summed up, the resulting expected risk 2062.5$. By using RISKEE utilizing the power of probability distributions we gain even more knowledge than the expected risk: We get the whole possibility space which can be presented as loss exceedance curve, shown in Fig. 3(b). It shows the probability of exceeding certain amounts of losses. To evaluate this, an overlay for the risk appetite is added which represent the acceptable risk. In this example we defined it as follows: with 70% chance we can afford to lose 3000$, with 50% chance we accept loosing 4000$, and with 10% we still can tolerate to loose 6000$. By interpolating and smoothing between these fixed points we get a curve which can be easily compared to the calculated risks. In this example the risk curve is below the acceptable risk, therefore we can accept it.

The advantages of this approach are visible in Fig. 3(b). RISKEE delivers not only an expected value, but much more information in the form of the distribution of all possible outcomes. For example, the resulting graph states that the expected range of outcomes will be between around 600$ and 5000$ with 90% confidence (range between the 5% percentile and 95% percentile). Furthermore, we can see what the extreme cases even go well beyond 6000$ (up until approximately 12400$, but the graph is cut off due to space saving reasons in this paper).

### 3.6 Threats to Validity and Limitations

In cyber-security we often have to deal with rare but catastrophic events, which cannot easily be judged and predicted. Therefore estimations of vulnerability could be off by magnitudes. Also the hacks are often so focused to one specific combination of technologies that experts have really quite some difficulties of predicting them. If there is a obvious hole in the protection line, it has to be protected anyways, therefore the risk assessment is most useful for the hard to estimate events which are unknown and infrequent. Nevertheless, since the method can also model other types of risk, it is not limited to the specific field of cyber-security, but could also be applied to other fields where the vulnerability can be judged in a more reliable way. In the current form the method only supports the modelling of monetary loss, but other types of loss metrics can easily be added in the future. Regarding Scaling: The method does scale very badly in its current form. Adding more nodes increases the calculation time manifolds. This limitation will be tackled in future research via usage of dynamic programming, and stochastic optimizations. Currently we have no defined file format for export and import of data, and no bindings to other languages. These are features which are on the agenda for future work in order to make RISKEE compatible and usable from multiple locations and environments.

## 4 Conclusion and Future Work

In this paper we took a step further into the direction of unified integrated quantitative risk assessment for safety and security. We connected to our previous work on this topic and mentioned some of the limitations established methods have. The contribution of this paper is the RISKEE method which is a method for risk assessment based on attack graphs and probability distributions. We described how to create such a graph and what the needed input values for determining the risk are. Furthermore, we showed the Riskee Propagation Algorithm to calculate risk by forward propagation of frequencies and backward propagation of risk. Finally, we discussed some aspects of computation with probability distributions, which we will follow on in future work. Also, in future work we want to investigate on mitigation possibilities and strategies, as well as enabling future predictions of risk by applying a decay on resistance over time and modelling dynamic evolving attackers.

## References

1. Ammann, P., Wijesekera, D., Kaushik, S.: Scalable, graph-based network vulnerability analysis. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, p. 217. ACM Press, Washington, DC (2002). https://doi.org/10.1145/586110.586140
2. Braband, J.: Risk assessment: as simple as possible (and no simpler). In: 1st IET International Conference on System Safety, vol. 2006, pp. 285–300. IEE, London (2006). https://doi.org/10.1049/cp:20060229
3. Braband, J.: Definition and analysis of a new risk priority number concept. In: Spitzer, C., Schmocker, U., Dang, V.N. (eds.) Probabilistic Safety Assessment and Management, pp. 2006–2011. Springer, London (2004). https://doi.org/10.1007/978-0-85729-410-4_322
4. Caltagirone, S., Pendergast, A., Betz, C.: The diamond model of intrusion analysis. Technical report. Center for Cyber Intelligence Analysis and Threat Research (2013)
5. Clemen, R.T., Winkler, R.L.: Combining probability distributions from experts in risk analysis. Risk Anal. **19**(2), 187–203 (1999)
6. Cox, A.L.: What's wrong with risk matrices? Risk Anal. **28**(2), 497–512 (2008). https://doi.org/10.1111/j.1539-6924.2008.01030.x
7. Cox, A.L., Babayev, D., Huber, W.: Some limitations of qualitative risk rating systems. Risk Anal. **25**(3), 651–662 (2005)
8. Cox, A.L., Popken, D.A.: Some limitations of aggregate exposure metrics. Risk Anal. **27**(2), 439–445 (2007)
9. Dobaj, J., Schmittner, C., Krisper, M., Macher, G.: Towards quantitative integrated security and safety risk assessment. In: Proceedings of the DECSOS Workshop for SAFECOMP 2019 Conference on Computer Safety, Reliability, and Security, p. 14 (2018)
10. Ferrell, W.R.: Combining individual judgments. In: Wright, G. (ed.) Behavioral Decision Making, pp. 111–145. Springer, Boston (1985). https://doi.org/10.1007/978-1-4613-2391-4_6
11. Franklin, B.D., Shebl, N.A., Barber, N.: Failure mode and effects analysis: too little for too much? BMJ Qual. Saf. **21**(7), 607–611 (2012). https://doi.org/10.1136/bmjqs-2011-000723
12. Freund, J.: Measuring and Managing Information Risk: A FAIR Approach. Butterworth-Heinemann, Oxford (2015)
13. Huang, M.S.: An approach for improvement of risk priority number in FMEA. DEStech Transactions on Computer Science and Engineering (itme), April 2017. https://doi.org/10.12783/dtcse/itme2017/8010

14. Hubbard, D.W.: The Failure of Risk Management: Why It's Broken and How to Fix It. Wiley, Hoboken (2009). oCLC: ocn268790760
15. IEC: IEC 60812: Analysis techniques for system reliability - Procedure for failure mode and effects analysis (FMEA) (2006)
16. ISO: ISO 26262 Road vehicles - Functional safety (2011)
17. Krisper, M., Dobaj, J., Macher, G.: Pitfalls, fallacies, and other problems in risk matrices using ordinal scales. Currently Under Review, p. 11 (2019, Submitted)
18. Macher, G., et al.: Integration of security in the development lifecycle of dependable automotive CPS (2017)
19. Macher, G., Sporer, H., Berlach, R., Armengaud, E., Kreiner, C.: SAHARA: a security-aware hazard and risk analysis method. In: Design, Automation & Test in Europe Conference & Exhibition (2015)
20. Messnarz, R., Sporer, H.: Functional safety case with FTA and FMEDA consistency approach. In: Larrucea, X., Santamaria, I., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2018. CCIS, vol. 896, pp. 387–397. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97925-0_32
21. Metropolis, N., Ulam, S.: The Monte Carlo method. J. Am. Stat. Assoc. **44**(247), 335–341 (1949). https://doi.org/10.1080/01621459.1949.10483310
22. Microsoft Corporation: Threat Modeling Tool. https://docs.microsoft.com/en-us/azure/security/azure-security-threat-modeling-tool-threats
23. Poolsappasit, N., Dewri, R., Ray, I.: Dynamic security risk management using Bayesian attack graphs. IEEE Trans. Dependable Secure Comput. **9**(1), 61–74 (2012). https://doi.org/10.1109/TDSC.2011.34
24. RiskAMP: The beta-PERT Distribution—RiskAMP (2010). https://www.riskamp.com/beta-pert
25. Schmittner, C., Gruber, T., Puschner, P., Schoitsch, E.: Security application of failure mode and effect analysis (FMEA). In: Bondavalli, A., Di Giandomenico, F. (eds.) SAFECOMP 2014. LNCS, vol. 8666, pp. 310–325. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10506-2_21
26. Schneier, B.: Attack trees - modeling security threats. Dr. Dobb's J. **24**(12), 21–29 (1999). http://www.schneier.com/attacktrees.pdf
27. Sellappan, N., Palanikumar, K.: Modified prioritization methodology for risk priority number in failure mode and effects. Analysis **3**(4), 10 (2013)
28. Stevens, S.S.: On the Theory of Scales of Measurement (1946)
29. Stone, M.: The opinion pool. In: Annals of Mathematical Statistics (1961)
30. The Open Group: FAIR - ISO/IEC 27005 cookbook (c103) (2010)
31. The Open Group: Risk Analysis (O-RA), October 2013. https://publications.opengroup.org/c13g
32. The Open Group: Risk Taxonomy (O-RT) 2.0, October 2013. https://publications.opengroup.org/c13k
33. Thomas, P., Bratvold, R.B., Bickel, E.: The risk of using risk matrices. In: SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers, New Orleans, Louisiana, USA (2013). https://doi.org/10.2118/166269-MS
34. Tidwell, T., Larson, R., Fitch, K., Hale, J.: Modeling internet attacks. In: Proceedings of the 2001 IEEE Workshop on Information Assurance and Security (2001)
35. Vesely, W.E., Goldberg, F.F., Roberts, N.H., Haasl, D.F.: Fault Tree Handbook (NUREG-0492). U.S. Nuclear Regulatory Commission (1981)
36. Vose, D.: Risk Analysis: A Quantitative Guide, 3rd edn. Wiley, Hoboken (2008)
37. Xing, L., Amari, S.V.: Fault tree analysis. In: Misra, K.B. (ed.) Handbook of Performability Engineering, pp. 595–620. Springer, London (2008). https://doi.org/10.1007/978-1-84800-131-2_38