Proceedings of the Annual General Donald R. Keith Memorial Conference
West Point, New York, USA
May 2, 2024
A Regional Conference of the Society for Industrial and Systems Engineering

Developing a Model-Based Systems Engineering Tool for Cybersecurity Risk Management of Micro-Electronic Devices

James Leland IV[1], Brett Schraeder[1], Collin Chilton[1], James Walliser[1], and Sathyamurthi Sadhasivan[2]

[1]Department of Mechanical Engineering,
United States Air Force Academy,
Colorado Springs, CO 80841

[2]The Aerospace Corporation,
El Segundo, CA 90245

Corresponding author's Email: james.walliser@afacdemy.af.edu

Author Note: Cadets Leland, Schraeder, and Chilton are senior systems engineering students at the United States Air Force Academy. Lieutenant Colonel James Walliser, Director of the Systems Engineering program, is the group's advisor.

Abstract: Cyber-security threats to micro-electronic components can drive significant cost into a program over its lifecycle. Cost savings can be achieved by selecting an appropriate mitigation strategy, but this requires a method for quantifying risks and countermeasures. This project developed a mathematical approach to quantify cybersecurity risk and implemented the solution in a model-based systems engineering product, called the Cyber-security Risk Assessment and Mitigation (CRAM) Tool. Users of the CRAM tool can select a set of cyber-security threats and visualize them in a 5x5 risk matrix, then explore the effectiveness of various countermeasures in reducing overall risk. The CRAM Tool produces the residual risk for a specific micro-electronic component that can be used to compare the effectiveness of threat-countermeasure combinations, allowing the user to develop a cost-effective mitigation strategy. Application of this mathematical risk quantification method and the CRAM Tool is demonstrated for hardware-trojan horse threats to a field-programmable gate array.

Keywords: Model-based systems engineering, risk analysis, micro-electronics, mitigation, optimization

1. Introduction

This paper addresses the challenge of managing cyber security risks for advanced micro-electronics. Contemporary digital systems integrate several micro-electronic components that enable system function, including hardwired Application-Specific Integrated Circuits (ASICs) and Field Programmable Gate Arrays (FPGAs). The risks associated with micro-electronic use span the life cycle of every digital system; however, the tools available to identify and mitigate these risks are limited. The goals of the project were to 1. develop a method that quantifies the cyber-security risks associated with micro-electronic use throughout the life cycle 2. Create a tool that enables cyber security experts to assess the risk to a specific system and develop a mitigation strategy to reduce both the severity and likelihood of that risk negatively affecting the system 3. Enable optimization of mitigation strategies to provide users with the most efficient and economical solution for their specific problem set.

The threat landscape of cyberattacks is rapidly changing and the potential impact of such attacks is uncertain, as there is a lack of effective metrics, tools, and frameworks to understand and assess the harm organizations face from cyber-attacks (Agrafiotis, Creese, Goldsmith & Upton, 2018). There are many different types of threats developers must account for when designing a system that utilizes micro-electronics. Counterfeiting devices, hardware trojans, reverse engineering hardware designs via de-composing or decrypting bitstream files, and side-channel analysis attacks all can negatively affect system performance. As more cloud-based providers, third-party accelerator suppliers, and open-source programmable design tools are available for prototyping, hardware acceleration, and high-performance computing, new threat vectors emerge that widen the threat profile of micro-electronic utilization in new, cloud-capable digital systems (Sunkavilli, Yu & Zhang, 2021). To provide adequate security; attack methodologies, vulnerability concepts and defense strategies should be thoroughly identified and employed by system developers (Aslan & Samet, 2017). The decisions of systems engineers and program managers on how to address cyber threats can have a tremendous impact on program cost, schedule, and performance.

There's no existing standard for quantifying the effects of these cybersecurity risks and the potential countermeasures

that can be taken to mitigate them. Traditionally, security risk management is an asset-based, qualitative process dependent on expert opinion and information at hand for a particular organization and development process. A group of experts apply their knowledge to a set of applicable risks to determine risk severity on a system, providing stakeholders with a qualitative evaluation of the system risk profile (Langenkamp, Jongsma, Phillipson & Wolthuis, 2021). Recently, cyber risk management techniques included some risk quantile-based measures, such as Cyber Value at Risk (Cy-VaR), that are widely employed in the financial domain but still lack detailed quantification of individual risk vectors and potential countermeasure effects. It is essential for project managers to accurately quantify individual risks associated with system security and operation, as accurate quantification facilitates efficient security in-vestment valuation and processes (Orlando 2021). To plan the size of security investments and estimate the consequent risk reduction, managers must quantify accurately. The lack of specialized methods and tools for quantifying risk along with the vast number of threats and mitigations makes it very difficult for program managers, developers, and engineers to select the right mitigation strategy to secure their system.

2. Project Scope

This project demonstrates the utility of Model-Based Systems Engineering (MBSE) through the application of a risk analysis tool scoped to focus on the use of FPGAs and Hardware-Trojan Horse (HTH) risks and countermeasures. An FPGA is an integrated circuit designed to be configured by a customer or a designer after manufacturing. Unlike traditional ASICs, FPGAs are built and distributed to be programmed and re-programmed after manufacturing. In an FPGA, internal hardware block components are connected via user-programmable interconnects that enable the circuit to customize operation for specific applications. In contrast to an ASIC, an FPGA can be manufactured and bought commercially for a wide range of applications. Customers can purchase FPGAs to emulate their ASICs before committing them to a mask and sending them out to the factory to be manufactured. Intel, AMD, and many other companies use FPGAs to emulate their chips before manufacturing them. FPGAs facilitate rapid prototyping and allow research in new architectures and communication techniques without the problem of ASIC production (Astarloa, Bidarte, Dorta, Jiménez & Martín, 2009).

Due to the frequent turnover of design iteration and prototyping, FPGAs are used extensively during the development stages of complex digital systems, for example vehicles and weapon systems. FPGAs allow engineers to quickly iterate through design options without the associated development cost and timeline of traditional ASIC production. The integration of FPGA reprogramming into the system development process is accompanied by an inherent increased risk to the system during both development and operation. Developers are tasked with the coding and implementation of FPGAs into system prototypes, introducing unique coding solutions from third party device software. This increases the avenues by which malicious actors and hackers can disrupt the development and procurement process.

The eventual transfer of FPGA solutions into a hardwired ASIC poses another significant risk to system integrity: HTH attacks. HTH attacks are malicious modifications of circuits designed to wreak havoc by altering the intended behavior of the system. When triggered, HTHs adversely affect electronics leading to reduced reliability, system failure, remote access into hardware, and sensitive information leakage. HTH attacks are specifically designed to be rarely activated and undetectable to conventional testing practices and verification methodologies. HTH attacks can be hidden in many micro-electronic components of integrated circuits, ASICs, and FPGAs. They can be inserted by adversarial entities including untrusted foundries, designers, vendors, as well as electronic design automation and computer-aided design software tool suites (Vosatka 2017). Manufactures that utilize FPGAs must be cognizant of such risks, taking careful consideration in security measures to ensure the integrity and safety of their engineered systems. Engineers and developers that utilize a combination of FPGA and ASIC components in design must identify and understand these risks early in the system life cycle, as preventative actions and countermeasures can be implemented to mitigate the risks associated with FPGA utilization. The Cyber-security Risk Assessment and Mitigation (CRAM) tool focuses on these HTH threats, as well as the potential countermeasures that can be taken to mitigate them.

3. Methodology

MBSE is the “formalized application of modeling to support system requirements, design, analysis, and verification and validation activities throughout the life cycle of a system” (Friedenthal, Griego & Sampson, 2007). MBSE extends beyond traditional engineering activities to support complex predictive and affects-based modeling. Our team harnessed MBSE tools to address the cyber-security challenge outlined above: We created a model to identify and quantify the effects of cyber-security vulnerabilities as well as the potential countermeasures to mitigate those risks, MATLAB and Microsoft Excel plug-ins enabled the quantification of individual risk vectors, providing stakeholders with the quantitative data necessary to make informed decisions on cyber-security risk management and security investment for their specific system. The flexibility of MBSE tools

enables engineers to easily adapt this method to analyze a range of systems and applications.

3.1 Mathematical Quantification of Threats and Countermeasures

To inform model output and results, we created a mathematical method for quantifying threats and countermeasures. The Aerospace Corporation developed an initial quantification method, and we worked with corporation employees to modify and adjust the method for our specific MBSE application. The method culminates in the definition of a single value that quantifies the residual risk of a selected group of threats on a system after the application of countermeasures. This enables subject matter experts to compare the effects of different threat profiles and mitigation strategies on system operation. The process for calculating Individual Risk Potential (IRP), Device Risk Potential (DRP), and Cumulative Mitigation Effect (CME) is defined and explained below.

3.1.1 Individual Risk Potential and Device Risk Potential

A device's risk potential is the aggregate of each threat's Individual Risk Potential (IRP_j). An IRP is a quantification of a threat's significance, with a higher IRP score indicating a more serious threat. The Method defines IRP as the product of Severity, Ease, and Likelihood. Severity (S_j) is the impact from the occurrence of a threat, ease (E_j) is the effort required to execute an attack, and Likelihood (L_j) is the probability of an adversary performing the attack. We assessed severity on a scale from 1-5, and ease with values of {0, 1, 3, 9}. Likelihood considers the effect of five risk factors, with each factor assigned a value of {1, 3, 5}. These risk factors are described in Table 1 below.

Table 1. Baseline Likelihood Risk Factor Definition

Risk Factor		
Ease of Obfuscation	*(EO)*	Difficulty for a malicious actor to mask the HTH in FPGA hardware
Willingness of the Adversary	*(WTR)*	Adversarial Willingness to invest time and resources for attack development
Technical Knowledge Gap	*(TKG)*	Technological capabilities of the adversary in comparison to system capabilities
Perceived Impact due to Time Frame	*(PIT)*	The adversary's perceived impact of the threat on the system given its' current state and life cycle phase
Perceived Impact on Resources	*(PIR)*	The adversary's perceived impact on the number of resources needed to counter the threat

An average of the five risk factors from Table 1 results in (L_j), a value for the expected occurrence probability for each threat.

$$\boldsymbol{L_j = \frac{\sum(EO,WTR,TKG,PIT,PIR)}{5}} \quad (1)$$

To determine an Individual Risk Potential (IRP_j) for each threat, the S_j, E_j, and L_j values are related according to Equation 2 below. Severity is left alone on the right side of the equation to account for the high impact threat severity has on system operation. While ease and likelihood for a specific threat might be low, a high severity value should still weigh heavily to alert stakeholders of the potential severe degradation in capability.

$$\boldsymbol{IRP_j = \frac{(E_j x L_j)}{2} x\, S_j} \quad (2)$$

A single micro-electronic device faces numerous HTH threats, and these threats must be considered cumulatively to determine an overall risk value. To do this, IRPs are aggregated for each threat in the threat profile. The geometric mean of the IRPs represents the Device Risk Potential (DRP), which communicates the level of peril for a device accounting for all possible threats. The geometric mean method is advantageous (as opposed to an arithmetic mean), as it is more appropriate for dependent values, and some of the threat vectors defined in the model share certain characteristics.

$$\boldsymbol{DRP = \left(\prod_{i=1}^{n} IRP_j \right)^{\frac{1}{n}}} \quad (3)$$

A complete understanding of a specific threat profile must also account for the complexity of addressing a high number of threats. As threat profile size increases, it becomes more difficult to adequately apply countermeasures to address each threat. To account for this, we integrated a Threat Complexity Value (TCV) into the calculus. TCV is a ratio of the active threats in the current threat profile (n) to the total number of possible threats a system could face over its' life cycle (including non-active threats). This relationship is shown below in Equation 4.

$$\boldsymbol{TCV} = \frac{\boldsymbol{n}}{(\boldsymbol{total\ number\ of threats})} \tag{4}$$

The TCV must be applied exponentially to the DRP value to produce an Adjusted Device Risk Potential (DRP_{adj}). The exponential relationship accurately accounts for the increasing difficulty in addressing threat profiles as they get larger.

$$\boldsymbol{DRP_{adj}} = (\boldsymbol{DRP})^{(\boldsymbol{1+TCV})} \tag{5}$$

3.1.2 Mitigations and Residual Risk

The next step in the quantification process is to account for the effects of mitigating countermeasures. The mitigation effect of each individual countermeasure (M) is calculated on a scale with the domain of {0,1,3,9}. Multiple countermeasures can be applied together to mitigate a single threat, so it is necessary to determine the average mitigation effect (M_{Avg}) of all applied mitigations. We sum the M values for each countermeasure (M_{used}) and divide by the highest possible effect the number of countermeasures could provide (multiply the number of countermeasures by 9 to reflect the highest possible value of total effectiveness). The result (M_{avg}) is percentage that relates the effectiveness of mitigations against a certain threat. Stacking multiple countermeasures has varying effectiveness on threat mitigation and may result in diminishing returns. Even with all applicable countermeasures applied, the threat may persist.

$$\boldsymbol{M_{Avg}} = \sum \frac{\boldsymbol{M_{Used}}}{(\boldsymbol{9x\ number\ of\ all\ recomended\ countermeasures})} \tag{6}$$

The application of countermeasures will lower a devices risk potential, but there will always be a residual risk to consider. Equation 7 provides the risk that remains for an individual threat after countermeasures have been applied. Residual Individual Risk Potential (IRP_{res}) represents how much risk is left after mitigation application.

$$\boldsymbol{IRP_{res} = IRP_j\ x\ (1 - M_{Avg})} \tag{7}$$

The final necessary calculation to inform the model results was the Adjusted Residual Device Risk Potential ($DRP_{adj\text{-}res}$). $DRP_{adj\text{-}res}$ is the remaining empirical risk value of a threat after the application of countermeasures, adjusted for threat complexity.

$$\boldsymbol{DRP_{adj-res}} = (\boldsymbol{DRP_{res}})^{(\boldsymbol{1+TCV})} \tag{8}$$

3.2 Modeling Process

We utilized CATIA Magic System of Systems Architect software and a SysML viewpoint to facilitate requirements definition, behavioral modeling, parametric constraint modeling, simulation activities, and a graphical user interface. The model integrates Block Definition Diagrams (BDD) and Internal Block Diagrams (IBD) which inform behavioral, parametric, and constraint diagrams. A separate Excel file, integrated via the Excel plug-in, provides model users with a single location to input the initial scale values for S_j, E_j, and L_j. as well as the individual mitigation effects for each threat-countermeasure pair. This information runs through a sequence of scripted MATLAB code and CAMEO parametric equations to generate values for equations 3-8 and define them within the model. Finally, a Simulation Configuration Diagram and a Graphical User Interface (GUI) provide model users and system stakeholders with visual output of model results in the form of a 5x5 risk matrix.

3.2.1 Parametric & Constraint Modelling

A single Parametric Diagram relates the constraint relationships already defined in the model. This parametric diagram integrates local code, MATLAB scripts, and Excel files to quantify the effects of the specific threat and countermeasure profile selected by the model user. The constraint blocks within the model serve as the connection between the CATIA model and the

MATLAB and Excel plug-ins. These elements enable the integration of our mathematical model defined above and inform the visualization tools utilized to clearly display model results.

3.2.2 Graphical User Interface

To easily visualize model output and results, we integrated a simulation configuration diagram and a Graphical User Interface (GUI). The GUI utilizes two dropdown menus to accept input user input for selection of applicable threats and countermeasures. After processing these inputs, the GUI integrates parametric and activity diagram functionalities using MATLAB code references to generate two side-by-side 5x5 risk matrix visualizations, as shown in Figure 2. The first matrix shows the IRP_{res} prior to countermeasure implementation and the second risk cube shows the IRP_{res} after countermeasure implementation. The cumulative results are displayed on the GUI interface. The device's residual risk before and after countermeasure mitigation are displayed as "Dev_RR Threats" and "Dev_RR w/CMs".

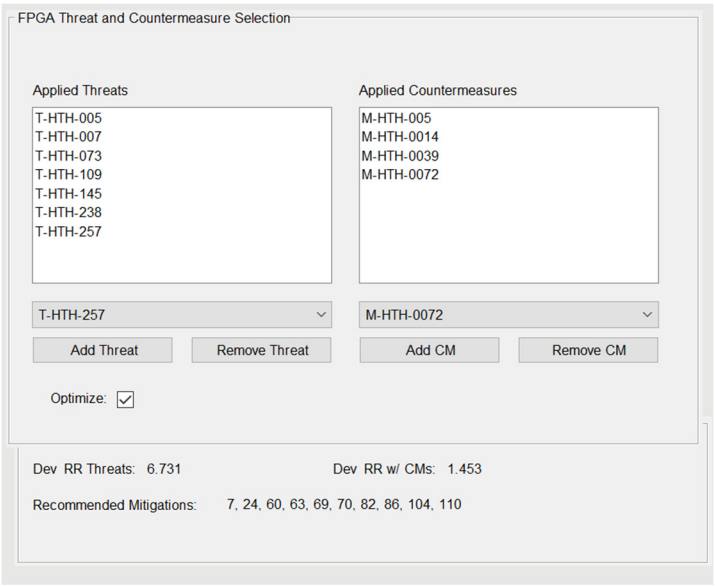

Figure 1: Graphical User Interface

Figure 2. Generated Risk Matrices

3.3 Optimization

The CRAM tool features a built-in optimization function. When selected via a checkbox in the GUI, the CRAM tool sorts through the entire mitigation suite to find the combination of the 10 most effective mitigations for the selected threat profile. The optimization doesn't consider the severity of individual threats, which means that that a threat with a risk value of one has equal impact on the mitigations selected as a threat with a value of nine. Because of this, a SME may be able to find a combination of mitigations that get the residual risk number lower than what the optimization provides; however, this would likely be extremely time consuming. The optimization tool helps the user find an optimized solution on a much shorter timeline.

4. Conclusion

MBSE digital tools enable engineers to conduct trade-off analyses and identify optimal system designs at a faster rate than previously possible under traditional acquisition programs and processes. This paper outlined the creation of the CRAM tool, a risk assessment tool that quantifies the effect of cybersecurity threats on micro-electronic system components. The example above was scoped to focus on HTH threats for FPGAs; however, the benefits of creating the CRAM tool in an MBSE environment enable engineers to adapt the model to fit different system designs and/or update design decisions within the model to re-assess risk.

The mathematical process and MBSE model work together to define the impact of HTH threat vectors on system security and integrity. User input enables initial quantification of threat vectors, providing concrete data to inform model definition specific to the user's system application. The model integrates external coding solutions and simulation capabilities to set up a Graphical User Interface and visualize of model output. This visualization provides relative comparison between countermeasure strategies taken by users to mitigate the effect of threat vectors, enabling optimization of countermeasure selection on the relative effectiveness of system safety. It is important to note that the methodology outlined in this paper is not limited in application to only HTH threats. The same process can be applied for other cyber-security threats and problem sets. The outputs from the model inform stakeholder financial decisions, providing relative comparison of countermeasure effectiveness. A user can iterate through different combinations of countermeasures, identifying the profile that provides the most utility for risk mitigation. The purpose of this model is to exemplify the impact MBSE tools can have on ensuring capable, secure digital systems are delivered on time and on budget to stakeholders.

5. References

Agrafiotis, I., Nurse, J. R. C., Goldsmith, M., Creese, S., & Upton, D. (2018). *A taxonomy of cyber-harms: Defining the impacts of cyber-attacks and understanding how they propagate.* Journal of Cybersecurity, 4(1).

Aslan, O., & Samet, R. (2017). *Mitigating Cyber Security Attacks by Being Aware of Vulnerabilities and Bugs.* 2017 International Conference on Cyberworlds, 222–225.

Dorta, T., Jiménez, J., Martín, J. L., Bidarte, U., & Astarloa, A. (2009). *Overview of FPGA-Based Multiprocessor Systems.* 2009 International Conference on Reconfigurable Computing and FPGAs, 273–278.

Friedenthal, S., Griego, R., & Sampson, M. (2009). INCOSE Model Based Systems Engineering (MBSE) Initiative.

Mencer, O., Allison, D., Blatt, E., Cummings, M., Flynn, M. J., Harris, J., Hewitt, C., Jacobson, Q., Lavasani, M., Moazami, M., Murray, H., Nikravesh, M., Nowatzyk, A., Shand, M., & Shirazi, S. (2020). *The History, Status, and Future of FPGAs: Hitting a nerve with field-programmable gate arrays*. Queue, 18(3), 71–82.

Orlando, A. (2021). *Cyber Risk Quantification: Investigating the Role of Cyber Value at Risk. Risks*, 9(10), Article 10. https://doi.org/10.3390/risks9100184

Sunkavilli, S., Zhang, Z., & Yu, Q. (2021). *New Security Threats on FPGAs: From FPGA Design Tools Perspective.* 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 278–283.

Vosatka, J. (2018). *Introduction to Hardware Trojans*. In S. Bhunia & M. M. Tehranipoor (Eds.), The Hardware Trojan War: Attacks, Myths, and Defenses (pp. 15–51). Springer International Publishing.

Walliser, J., Tossell, C., Burcham, S., Haasl, C., Cabo, A., Kotter, C., & Bearden, K. (2023). *Model-Based Systems Engineering Cybersecurity of Field Programmable Gate Arrays for Future Weapons Systems*.

Wolthuis, R., Phillipson, F., Jongsma, H.-J., & Langenkamp, P. (2021). *A framework for quantifying cyber security risks.* Cyber Security: A Peer-Reviewed Journal, 4(4), 302–316.