Security In Embedded Systems With Differential Dimensional View

Karthikeyan Sundarasamy
Assistant Professor, Dept of Electronics and Communication Systems
KG College of Arts and Science, Coimbatore, India

Abstract- *The primary goals of this paper are to investigate the protection of embedded systems at completely different levels of abstraction and to propose a new procedure to assess and improve the protection of embedded systems during numerous product life cycle phases. To realize these goals, this paper introduces new classification of embedded systems attacks using a novel multi-dimensional illustration, explores the attainable threats to embedded systems, and proposes a new procedure to evaluate and improve the protection of embedded systems during numerous product development phases.*

Keywords- systems-on-chip (SoC), embedded systems security, Common Criteria

I. INTRODUCTION

Embedded systems are widely used in many fields, yet current work on embedded systems security considers only simple physical attacks against the hardware itself and straightforward software defenses. This has raised serious concerns regarding possible threats to military systems, financial infrastructures, and even household consumer appliances. In fact, security professionals concluded that the failure of military devices in different incidents was due to electronic warfare. In particular, Trojans were added to ICs used in suspected military equipments to shut them down at certain times. Even at the regular consumer level, electronic devices, such as cell phones, are currently being integrated into enterprises, government agencies, and even in the military. These devices hold valuable and sensitive contents and thus face the same risk of being attacked on a daily basis. The problem with current straightforward software defenses in most systems is that hardware is the base physical layer in any embedded system and an attack on that layer can allow a full control over the software running above. This low-level control enables sophisticated attacks that can defeat regular software-based defenses.

Attacks on embedded systems can have different forms, such as theft of service, cloning, spoofing, and reverse engineering. In this paper, we categorize the possible attacks on embedded systems and visualize the different types of attacks using a multi-dimensional analysis. Based on our analysis, we introduce a new methodological security evaluation scheme to help designers better evaluate the security of their designs.

1.1. *Main contributions*

This paper presents two main contributions:

1. Creating a new classification of embedded systems attacks using a novel multi-dimensional representation. This new classification allows system designers to study the security of their embedded systems at 27 different scenarios.
2. Developing a new methodological security evaluation scheme to assess and improve the security of embedded systems during various product life cycle phases. This new scheme identifies the requirements of four security levels and complementary to other methods, such as the Cryptographic Module Validation Program (CMVP) and Common Criteria (CC).

This paper is organized as follows. Section 2 reviews existing security standards. Section 3 highlights related work. Section 4 introduces a new systematic classification of implementation-oriented attacks on embedded systems and presents three main perspectives that could be used to classify attacks on embedded systems. Section 5 discusses our proposed procedure to evaluate the security of embedded systems. Finally, we draw our conclusion and suggest new ideas for future work in Section 6.

II. REVIEW OF EXISTING SECURITY STANDARDS

Cryptographic Module Validation Program (CMVP) was established by the National Institute of Standards and Technology (NIST) and Communications Security Establishment Canada (CSEC) in 1995. CMVP validates commercial cryptographic modules to the Federal Information Processing Standard (FIPS) 140-2 and other cryptography-

based standards. On the same context, Common Criteria (CC) lists seven Evaluation Assurance Levels (EALs).

2.1. Review of CMVP and FIPS 140-2:

In 2005, NIST and CSEC identified four security levels for cryptographic modules to protect sensitive information in computer and telecommunication systems. The first security level requires minimal physical protection and no specific physical security mechanisms are required beyond the requirement for production-grade components. The second security level adds the requirement for tamper evident mechanisms, which includes the use of tamper evident coatings or seals on removable covers of the module. The third security level intends to have a high probability of detecting and responding to attempts at physical access, use, or modification of the cryptographic module. The fourth security level provides the highest level of security defined in the FIPS 140-2 standard. At this security level, the physical security mechanisms provide a complete envelope of protection around the cryptographic module. This includes protective a crypto logical module against a security compromise as a result of environmental conditions or fluctuations outside of the module's traditional operational ranges for voltage and temperature.

2.2. Review of CC :

CC is another security scheme that identifies seven Evaluation Assurance Levels (EVLs). EAL-1 provides a basic level of assurance just to make sure that the Target of Evaluation (TOE) is consistent with its specifications. EAL-2 requires developer testing, a vulnerability analysis, and independent testing based upon more detailed TOE specifications. EAL-3 requires more complete test coverage of the security functionality to make sure that the TOE will not be tampered with during development. EAL-4 adds the requirement for more design description, the implementation representation for the entire TOE Security Functions (TSF), and improved mechanisms and/or procedures that provide confidence that the TOE will not be tampered with during development. EAL-5 requires semiformal style descriptions, a lot of structured architecture, and improved mechanisms and/or procedures that offer confidence that the TOE will not be tampered with during development. EAL-6 requires additional comprehensive analysis, a structured representation of the implementation, additional architectural structure, additional comprehensive independent vulnerability analysis demonstrating resistance to penetration attackers with a high attack potential. EAL-7 is applicable to the development of security TOEs for application in extraordinarily high risk situations and/or wherever the high value of the assets justifies the higher prices.

2.3. Limitations of CMVP/FIPS 140-2 and CC

Although CMVP and FIPS 140-2 provide an essential standard that helps protecting sensitive information in computer and telecommunication systems, they focus on the cryptographic modules and do not cover the complete system, including hardware modules. The cryptographic modules considered by the standard are assumed to be completely secured and inherently free from any malicious content. Furthermore, the existing standard does not provide security measures to assess and classify threats during various development phases, programmability levels, or integration levels. On the same context, the US National Institute of Standards and Technology (NIST) has proposed using the CC and system-level protection profiles (SLPPs) to specify security requirements in large systems. CC is widely used by software vendors, biometric system designers and smart-card application-developers. A substantial research and practical experiences exist for the CC, such as frame-work development, vulnerability awareness improvement, and structuring modular safety software certification by using CC concepts. However, attempts to apply CC policies in the USA federal systems engineering environment faced three specific is-sues that made it difficult to implement CC. These issues are: (1) complex technology environments, (2) complex and inflexible standards, and (3) the lack of a clear relationship between the CC and the systems development approach.

Because of these issues, and many others, some independent consultants started to question the future of the CC. Hearn listed the following three specific key observations based on the 4th International CC Conference:

1. Little commercial interest is driving the CC market; most evaluations and certifications result from government regulations or purchases.
2. Buyers see certifications as a "tick in the box" for procurement and seldom read the security target or certification reports, or even use the evaluated configurations.
3. Sellers do not see CC as a product-improvement evaluation methodology.

After complying with the CC requirements, many users still wonder how this CC-evaluated product improves their IT systems security. Specific for hardware systems, CC does not provide a clear implementation of the requirements. Furthermore, CC focuses on the development phase of the

product and is missing the possible attacks during and after the production phase.

Therefore, in this paper, we develop a new multi-dimensional scheme to address these missing issues and provide a complementary vision to existing hardware security requirements in both CMVP and FIPS 140-2, as well as CC.

III. RELATED WORK

This section highlights related work in embedded systems security. The work published in this area can be classified into three categories: (1) modeling and analyzing hardware attacks and security requirements, (2) providing solutions for the security of embedded memories and supporting on-chip secure communications, and (3) managing security requirements in system-on-chip (SoC) and FPGA-based designs.

3.1. Modeling and analyzing hardware attacks and security requirements

Analyzing attacks and evaluating systems' security are becoming more challenging with the increasing complexity of integrated circuits (ICs) . Companies tend to outsource several parts of their designs and integrate third-party IPs to achieve cost efficiency and fast time-to-market. Because of the lack of enforcing a common standard for hardware security in the IC industry, researchers made several attempts to standardize the security requirements for embedded systems. Researcher presented a classification of several hardware threat models and discussed possible evaluation metrics for important hardware-based attacks. Koppel et al. analyzed the Hardware Security Modules (HSM) high availability settings and discussed two possible flaws that could lead to security problems. The authors also discussed possible solutions that could be applied by targeted organizations. At a higher level, Lee discussed two classes of hardware security: an architecture for hardware-enhanced security and a secure hardware platform .

3.2. Providing solutions for the security of embedded memories and supporting on-chip secure communication

Memories are at the heart of any embedded system and the center of information storage. Protecting memories is one of the main goals for any system security requirement. Researches proposed several solutions to address memory issues. Researcher developed MemTracker, a new hardware mechanism that can be configured by developers to perform several tasks related to memory access monitoring. The main idea of MemTracker is associating memory words with few bits that represent several states. Then, the system monitors the memory access and logs any event that can affect the current state. A programmable state transition table is used to switch to the next state after detecting the event. At the silicon level, proposed a replacement to the classical memory elements, called NOVeA. NOVeA uses a scalable embedded flash technology with an integrated on-chip SRAM array to facilitate password authentication. A different approach proposed to utilize a physics-based model of the domain wall memory (DMW) to comprehend the process variations and use physically unclonable functions (PUFs) to secure the key generation process. PUFs are preferred to be used, especially relay-PUF and memory-PUF de-signs, as they could provide a higher degree of resilience against reverse engineering. At the micro-architecture level, researchers addressed covert timing channels through different approaches. In Chen presented CC-Hunter, which is a framework that allows users to detect the possible presence of covert timing channels. This is done by developing an algorithm to analyze conflict patterns used in covert transmission. With a focus on contention based covert timing channels, the algorithm presented by Chen. The proposed work was evaluated using covert timing channels on wires, logic, and memories.

3.3. Managing security requirements in SoC and FPGA-based designs

The emerging utilization of system-on-chip (SoC) and networks-on-chip (NoC) designs in current embedded systems comes with high risk of systems failures due to hardware based attacks. Researchers proposed different solutions to address the security issues in current emerging technologies. Kim et al. explored different methods to recover systems from hardware attacks by changing the configuration and mode of operations [10]. They proposed architectural features of SoC that can minimize the impact of hardware attacks and provide seamless system operation during and after function replacement [10]. Tiwari et al. presented a new approach for microkernel, processor, and I/O system with strict and provable information flow security [11]. Their main idea is constructing a configurable architectural skeleton that couples the microkernel with low level hardware implementation. This integration allows information flow properties of the entire construction to be captured and statically verified from the system level all the way down to the gate-level implementation [11]. Several work was done also to address security issues in NoC and FPGA applications. Wassel et al. introduced SurfNoC, an on-chip network that improves the security of on-chip communication [12]. Swierczynski et al. investigated a possible attack vector against cryptography. They demonstrated how attackers could modify an FPGA bitstream to break cryptographic algorithms

and discussed possible solutions to countermeasure these attacks [13].

Although a lot of research has been done in this area, system designers need a new classification of embedded systems attacks that takes into consideration the whole system perspective. It is also clear that there is a great need for a standard methodological security evaluation scheme to assess and improve the security of embedded systems during various product life cycle phases. Our work in this paper addresses these two issues.

IV. CLASSIFICATIONS OF ATTACKS ON EMBEDDED SYSTEMS

Fig. 1 shows a multi-dimensional representation of embedded systems that could be used to visualize possible attacks from different perspectives. We developed this figure to help engineers better understand the possible threats to their applications at each integration level, taking into consideration the programmability level and the product life cycle phase. For each level of integration, designers must (1) explore all possible threats to the target de-sign based on the other two dimensions and then (2) apply the required security measures to protect the target design against these threats.

There are many different ways to classify attacks on embedded systems [14]. In this paper, we present three main perspectives that could be used to classify attacks on embedded systems. These perspectives are: programmability level, integration level, and life cycle phase.

Other dimensions could also be added to the representation in Fig. 1, e.g., controllability and observability. However, we found the three dimensions, shown in Fig. 1, better represent embedded systems from the architectural and life cycle perspectives, whereas controllability and observability could not be used to visualize the possible security threats of embedded systems at the same level of abstraction. We selected the dimensions of our representation based on their significance to quantify the overall security of the different embedded systems, while being reasonably practical to be considered by designers and manufacturers.

4.1. Classification based on programmability level

4.1.1. Hardware (HW) attacks

Hardware (HW) attacks include hijacking, data monitoring, and denial-of-service attacks that prevent the system from functioning correctly after being triggered by a predetermined input sequence [15]. Hardware attacks also include physical attacks, such as re-verse engineering of a chip or a printed circuit board (PCB) and using probes to monitor inter-component communications. Another type of hardware attacks is hardware Trojan, which is a malicious addition or modification to a hardware circuit to change its functionality .Hardware-based side-channel attacks rely on monitoring the physical behavior of the system. This can be done by monitoring the voltage level, electromagnetic radiation, and data traffic between chips. It can also be done by adding a malicious circuit to facilitate the observation process [16].

4.1.2. Firmware (FW) attacks

Firmware attacks include attacks against the OS kernel, as demonstrated in [17]. While firmware in ROMs and EPROMs are not usually updated by the customer, EEPROMs or flash memories can be re-written from software. Embedded systems currently store the firmware code, in most cases, in flash memories to allow OS upgrades through any Internet connection. Although in-field updates ease system upgrades, they open the door to bypassing security features and could result in illegal privilege escalation.

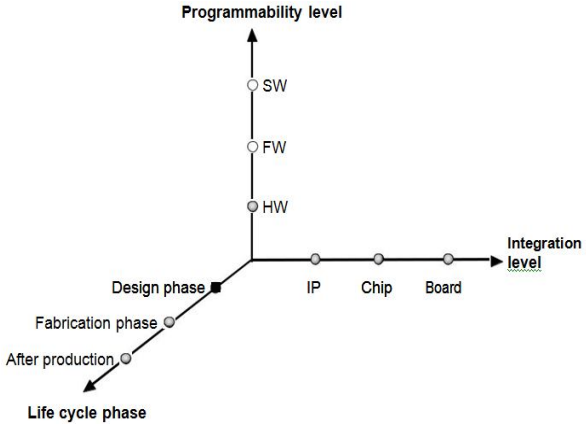

Fig. 1. A multi-dimensional representation of embedded systems.

4.1.3. Software (SW) attacks

Software attacks are usually designed to alter the behavior of the system, such as viruses, software Trojans, etc. Software attacks could also target the packet switching protocols and could result in a malicious behavior, e.g., packet replay, unknown destination, or deadlock. One of the common software attacks is the buffer overflow, which causes serious damages to application-specific embedded systems [18]. Some of the software attacks on micro-controllers aim at either exploiting vulnerabilities that are not malicious themselves or executing malicious code. On the other hand, software-based side-channel attacks rely on monitoring the logical behavior of the system. This can be done by monitoring the data traffic

and extracting confidential information or by adding a malicious code to facilitate the observation process.

4.2. Classification based on integration level

4.2.1. Intellectual property (IP)-level attacks

There are three types of IPs: soft, firm, and hard IPs [19]. Each type of IPs is vulnerable to various attacks. A soft IP is subject to modifying the source code so that hackers can fully control the whole system later on. A firm IP is subject to connecting malicious IPs to its IO interfaces so that hackers can affect data integrity and cause silent data corruption. A hard IP is subject to cloning during and after the fabrication process. Attacks on IP cores also include adding Trojan circuits to try to do attacks at specific times or after specific sequence of events.

4.2.2. Chip-level attacks

Chip level attacks include chip cloning using advanced imaging techniques and fault attacks, which is the new class of at-tacks on secure microcontrollers [20]. Another type of attacks is changing the mode of operation based on the geographical location, time zone, time of the day, day of the year, etc. This includes remote shutting down, remote utilization, remote reconfiguration, etc. There are also other types of attacks related to programmable logic designs, e.g., those related to FPGA-based designs. These attacks include bit stream reverse engineering, radiation-induced faults, and illegal remote reconfiguration .

4.2.3. Board-level attacks

Board level attacks can be categorized into three main groups: invasive, semi-invasive, and non-invasive attacks [21]. Invasive at-tacks require physical access to the board to reverse engineer the design layout. Hackers use mechanical, chemical, and image processing techniques to copy and reproduce the PCB layout design layer-by-layer. Semi-invasive attacks work only for simple, double-sided PCBs, as hackers use photo scanners to scan the top and bottom layers, then convert the scanned image pixel-format into a vector-format that could be read by CAD tools. Non-invasive at-tacks can be either passive or active. Passive attacks do not interact with the board and just monitor or observe the data traffic between different chips, whereas active attacks tend to play with the supply voltage and clock signals to disable the board protection or force a chip to do a wrong operation.

4.3. Classification based on life cycle phase

4.3.1. Design phase attacks

Attacks during the design phase are usually executed by an insider. Insider threats are among the most significant security breaches because the attacker can gain access to protected information, which could result in destruction or cloning of a design, fault generation, spoofing, adding a kill switch, or gaining illegal access to the target system at any time in the future . One way to address these attacks is by using processor encryption to ensure that an insider cannot activate any Trojans inserted at the design phase .

4.3.2. Fabrication phase attacks

Attacks during the fabrication phase are usually related to market competition. During the fabrication phase, an attacker can try to reverse engineer or copy a specific IP, a chip, or a aboard from the owner with a primary goal of gaining an advantage in the marketplace.

4.3.3. After-production attacks

After-production attacks happen when the design, whether it is an IP, a chip, or a board, is released to the market and it is al-ready in the customer hands. Attacks, in this case, can be either a physical attack aiming at cloning the design, achieving privilege escalation, or extracting confidential information; or a remote at-tack through online updates to firmware and software applications .

V. PROCEDURE TO EVALUATE THE SECURITY OF EMBEDDED SYSTEMS

5.1. Attack examples

To give concrete examples of different types of attacks that be-long to the 27-point classification, we present the following three examples.

1. Example 1: *A SW attack that operates on a chip level during the design phase.* At early design phases of a SoC-based product that requires a SW-HW co-design, an insider developer can add a malicious SW that runs on a SW-HW Co-design framework to shut the system down, delete sensitive information, or store and transmit RAM contents when triggered by a unique external signal. This type of attacks is sometimes classified as a *Kill-Switch* and it is a SW attack that starts during the design phase and targets a chip level execution. One real case was presented by Adee in [1].

2. Example 2: *A FW attack that operates on an IP level during the fabrication phase.* With the current trend of IC

design companies being fabless and outsourcing fabrication to various semiconductor manufacturing companies, IPs are becoming more vulnerable to reliability attacks. Reliability attacks are induced in the offshore fab house during the fabrication phase to modify the original firmware or chip design. Due to budget limitations, managers might consider one-time programmable (OTP) memories and ship their FW that works on a certain IP to the fab house to integrate the firmware programming in the fabrication process in order to save the cost. This can also introduce a great risk of man-in-the-middle attacks after the product is shipped to end users.

3. Example 3: *A HW attack that operates at the board level after-production*. One example is the PCB reverse engineering. Various techniques to attack PCBs have been reported, including using X-ray stereo imaging to separate the layers of two layered PCB. Our classification helps system designers consider PCB protection methods at early phases of the design to eliminate after-production attacks.

These examples demonstrate the strength of our 3-D classification system and help explain how attacks could be mapped to each one of the 27 points.

5.2. Evaluation procedure

To improve a system's immunity to possible threats, the target system must be protected at all levels. To achieve this goal, the following evaluation procedure is suggested.

1. Using Fig. 1, create a list of 27 scenarios, representing all possible threats to the target system in each scenario.
2. For each the 27 possible scenarios, indicate whether the system is protected against the possible attacks or not and assign a severity level for each corresponding attack.
3. The security of the system is as strong as the weakest protected point.

To simplify this procedure, each one of the 27 scenarios can be assigned one of four security levels. The following subsection introduces these security levels.

No.	Life cycle	Integration phase	Programmability level	Security level
1	Design phase	IP	HW	x
2			FW	x
3			SW	x
4		Chip	HW	x
5			FW	Level 2
6			SW	x
7		Board	HW	Level 2
8			FW	x
9			SW	x
10	Fabrication phase	IP	HW	x
11			FW	x
12			SW	x
13		Chip	HW	Level 1
14			FW	Level 1
15			SW	x
16		Board	HW	Level 1
17			FW	x
18			SW	x
19	After production	IP	HW	x
20			FW	x
21			SW	x
22		Chip	HW	Level 1
23			FW	Level 3
24			SW	x
25		Board	HW	Level 1
26			FW	x
27			SW	x

5.3. Security levels

Since security requirements depend on the target system and application, we introduce, for the first time, four security levels for embedded systems. The proposed levels consider a layered approach of multiple security mechanisms to protect against a specific threat or to reduce a vulnerability. This proposal aims to apply security measures during product design and to treat security as an integral part of the overall system design.

5.3.1. Security level 1: basic security

Security level 1 is the basic acceptable level for any embedded system. It requires that designers use tamper-resistance mechanisms to make tampering of an IP, a chip, or a board very difficult. For example, encapsulating the entire PCB with resistant epoxy compound will help protect the circuitry. Designers can hide the PCB tracks in the inner layers and move the power polygons to the top and bottom layers, as opposed to exposing the tracks to the surface of the board. Another example is using an embedded component technology, such as buried resistors and capacitors in the internal PCB layers. At the chip level, designers can employ security measures to prevent attackers from reading stored data, such as using physical fuses on ROMs, boot-block protection in flash memories, and lock bits in microcontrollers.

5.3.2. Security level 2: intermediate security

Security level 2 requires applying all security measures in level 1 plus adding two more features. The first feature is detecting any internal malicious behaviors that

prevent the system from functioning correctly. This includes access control by employing authentication techniques, IP profiling and monitoring during different modes of operation, testing all third-party design modules to make sure that no Trojans are hidden inside. The second feature is protecting the design against attacks during the fabrication phase, including hardware obfuscation, watermarking, secret key activation, custom-generated boot-loaders, and other techniques to prevent overproduction and reverse engineering during the fabrication phase.

5.3.3. Security level 3: high security

Security level 3 requires that designers apply all security measures in level 2 plus adding more features for tamper evidence, detection, and response. Tamper evidence and detection mechanisms allow a system to be aware of tampering and ensure that a visible evidence is left behind when tampering occurs. The detection can be done using micro switches, sensors, or other circuitry for hardware devices. For software modules, this can be done using firewalls and authentication methods. Once an attack is detected, tamper response varies according to the target system. For example, the system can respond by completely shutting down or disabling itself, or erasing all memory units to prevent an attacker from accessing secret data, etc. Actions in level 3 do not include any physical destruction of the system.

5.3.4. Security level 4: advanced security

Security level 4 prevents attackers from gaining access in any way, shape, or form to any part of the system. In addition to all security measures in level 3, level 4 allows physical destruction of a system using a small explosive charge to completely prevent any access to the system once an illegal attempt is detected.

These four presented security levels are based on the 27 scenarios presented in Fig. 1. To consider systems' security at early design phases, design tools should accommodate security-plug in features to evaluate the probability of successful hacking for different implementations.

VI. CONCLUSION

This paper presented a completely new classification for embedded systems security. This new multi-dimensional classification can help engineers have a better understanding of the security level of their final product and, hence, protect their embedded systems designs at different life cycle phases. We plan to extend this work on two directions. The first direction is to develop a reliable measurement method to quantify the level of the target security so that it can be represented as one number, based on our multi-dimensional diagram. The second direction is to apply our method on real case studies that currently use CMVP/FIPS 140-2 and CC to evaluate the accuracy of the pro-posed model compared to them.

REFERENCES

[1] S. Adee, The hunt for the kill switch, IEEE Spect. 45 (5) (2008) 34–39.
[2] Q. Li, G. Clark, Mobile security: a look ahead, IEEE Secur. Priv. 11 (1) (2013) 78–81.
[3] The Cryptographic Module Validation Program (CMVP), 2014. Online: https://www.cse-cst.gc.ca/en/group-groupe/crypto-module-validation-program (accessed January 2016).
[4] Common Criteria Portal, The common criteria for information technology se-curity evaluation (CC), part 3: security assurance components, 2012. Online: http://www.commoncriteriaportal.org (accessed January 2016).
[5] K. Taguchi, N. Yoshioka, T. Tobita, H. Kaneko, Aligning security requirements and security assurance using the common criteria, in: Proceedings of the 2010 Fourth International Conference on Secure Software Integration and Reliability Improvement (SSIRI 2010), Singapore, 2010, pp. 69–77.
[6] S. Ardi, N. Shahmehri, Introducing vulnerability awareness to common crite-ria's security targets, in: Proceedings of the Fourth International Conference on Software Engineering Advances (ICSEA 2009), Porto, Portugal, 2009, pp. 419– 424.
[7] C. Preschern, K. Dietrich, Structuring modular safety software certification by using common criteria concepts, in: Proceedings of the 2012 38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), Izmir, Turkey, 2012, pp. 47–50.
[8] F. Keblawi, D. Sullivan, Applying the common criteria in systems engineering, IEEE Secur. Priv. 4 (2) (2006) 50–55.
[9] J. Hearn, Does the common criteria paradigm have a future? IEEE Secur. Priv. 2 (1) (2004) 64–65.
[10] L.-W. Kim, J.D. Villasenor, Dynamic function replacement for system-on-chip security in the presence of hardware-based attacks, IEEE Trans. Reliab. 63 (2) (2014) 661–675.
[11] M. Tiwari, J.K. Oberg, X. Li, J. Valamehr, T. Levin, B. Hardekopf, R. Kastner, F.T. Chong, T. Sherwood, Crafting a usable microkernel, processor, and I/O sys-tem with strict and provable information flow security, in:

Proceedings of the 38th Annual International Symposium on Computer Architecture (ISCA), San Jose, CA, USA, 2011, pp. 189–199.

[12] H.M.G. Wassel, Y. Gao, J.K. Oberg, T. Huffmire, R. Kastner, F.T. Chong, T. Sher-wood, Networks on chip with provable security properties, IEEE Micro 34 (3) (2014) 57–68.

[13] P. Swierczynski, M. Fyrbiak, P. Koppe, C. Paar, FPGA Trojans through detecting and weakening of cryptographic primitives, IEEE Trans. Computer-Aided Des. Integr. Circ. Syst. 34 (8) (2015) 1236–1249.

[14] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, S. Ravi, Security as a new di-mension in embedded system design, in: Proceedings of the 41st Design Au-tomation Conference (DAC 2004), San Diego, CA, USA, 2004, pp. 753–760.

[15] M. Tehranipoor, C. Wang, Introduction to Hardware Security and Trust, Springer-Verlag, Berlin, Germany, 2012.

[16] K. Hu, H. Chandrikakutty, R. Tessier, T. Wolf, Scalable hardware monitors to protect network processors from data plane attacks, in: Proceedings of the First IEEE Conference on Communications and Network Security (IEEE CNS 2013), Washington D.C., USA, 2013, pp. 314–322.

[17] G. Hoglund, G. McGraw, Exploiting Software: How to Break Code, Addison-Wesley Professional, Reading, MA, USA, 2004.

[18] Z. Shao, Q. Zhuge, Y. He, E.H.-M. Sha, Defending embedded systems against buffer overflow via hardware/software, in: Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC'03), Las Vegas, NV, USA, 2003, pp. 352–361.

[19] M. Tehranipoor, C. Wang, Introduction to Hardware Security and Trust, Springer-Verlag, Berlin, Germany, 2012.

[20] S. Khaleghi, K.D. Zhao, W. Rao, IC piracy prevention via design withholding and entanglement, in: Proceedings of the 2015 20th Asia and South Pacific Design Automation Conference (ASP-DAC), Tokyo, Japan, 2015, pp. 821–826.

[21] S. Ghosh, A. Basak, S. Bhunia, How secure are printed circuit boards against Trojan attacks? IEEE Des. Test 32 (2) (2015) 7–16.