Erudite Journal of Engineering Technology and Management Sciences
Vol. 6, No. 1, March 2026, pp. 06-11
Journal homepage: http://www.ejetms.com

Hardware Security: Circuit Obfuscation and IC Protection in Modern Supply Chains

[1] Soundharya V, [2] Dr Basavaraj G Kudamble

[1] Electronics and Communication Engineering (VLSI Design), Siddharth Institute of Engineering and Technology, Puttur, Andhra Pradesh, India.

[2] Electronics and Communication Engineering, Siddharth Institute of Engineering and Technology, Puttur, Andhra Pradesh, India.

Copyright: ©2026 The authors. This article is published by EJETMS and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

Received: 01 December 2025
Accepted: 06 February 2026

Keywords:
Hardware Security, Circuit Obfuscation, Logic Locking, Physical-Unclonable Functions, IC Supply Chain Protection, Hardware Trojans

ABSTRACT

The integrated circuit (IC) supply chain has undergone significant globalization over the past decade, introducing critical security vulnerabilities that threaten the integrity of hardware systems. This research paper addresses the emerging challenges of hardware security through the lens of circuit obfuscation and intellectual property protection. With the proliferation of third-party fabrication services and soft/hard IP cores in system-on-chip (SoC) development, the potential for malicious hardware Trojans, counterfeiting, and reverse engineering attacks has escalated dramatically. This paper presents a comprehensive examination of hardware security methodologies, including hardware Trojan detection techniques, formal verification approaches through proof-carrying hardware (PCH), logic locking mechanisms, and the utilization of physical-unclonable functions (PUFs) as security primitives. Additionally, we propose and evaluate a cube stripping-based functional analysis attack on state-of-the-art logic locking algorithms. Our experimental results on benchmark circuits demonstrate the effectiveness of the proposed approach, achieving significant improvements in area and delay optimization compared to conventional methods. The paper concludes with insights into emerging trends in hardware security, including hardware-assisted cybersecurity solutions and the application of novel transistor technologies for enhanced security primitives.

1. INTRODUCTION

1.1 Hardware Security In The Modern Ic Supply Chain

Hardware has traditionally been perceived as a trusted foundation of computer systems, operating as an abstract layer executing instructions from higher-level software layers[1]. However, this assumption of inherent trustworthiness is increasingly challenged by the complexity and globalization of modern IC design and manufacturing processes. The evolution of hardware-related security research has expanded from cryptographic algorithm implementations and copyright protection through watermarking to encompassing the protection of hardware designs themselves from malicious manipulation[1]. For decades, the IC supply chain was protected by high barriers to entry, primarily due to the substantial capital investment required to establish cutting-edge fabrication facilities. However, the contemporary landscape presents a fundamentally different scenario. The exponential growth in design complexity, coupled with the prohibitive costs of advanced fabrication nodes, has led to the geographic distribution of the IC supply chain across multiple countries and organizations[1]. This fragmentation, while economically beneficial in reducing design workload, fabrication costs, and time-to-market (TTM), introduces unprecedented security vulnerabilities.

1.2 Emerging Threats In The Ic Supply Chain

The reliance on third-party resources—including foundry services and intellectual property (IP) cores—creates multiple attack surfaces that malicious actors can exploit. Adversaries may compromise the supply chain at various stages, inserting hardware Trojans into fabricated chips or delivering IP cores containing malicious logic or design flaws[1]. These threats have invalidated the previous assumption that IC supply chains were sufficiently isolated from unauthorized access.

1.3 Evolution of Hardware Security Research

The trajectory of hardware security research has followed a distinct evolution. Initial efforts concentrated on hardware Trojan detection through pre-deployment and post-deployment

methodologies[1]. Subsequently, research attention shifted toward formal verification approaches, leveraging mathematical rigor to provide security assurance for hardware designs written in hardware description languages (HDLs)[1]. More recently, the focus has expanded to the development of trustworthy hardware infrastructure and root-of-trust construction[1]. A prominent example of this paradigm shift is the development of physical-unclonable functions (PUFs), which exploit device process variations to generate unique chip-specific identifiers in challenge-response pair formats[1]. Emerging technologies such as spin-transfer torque (STT) devices, memristors, and spintronic domain walls are being investigated for leveraging their intrinsic properties for security applications[1].

1.4 Research Contributions and Paper Organization

This paper presents a systematic examination of hardware security from multiple perspectives:

1. Comprehensive review of hardware Trojan detection methodologies, including enhanced functional testing and side-channel fingerprinting approaches
2. Analysis of formal verification techniques, specifically proof-carrying hardware frameworks for IP core validation
3. Evaluation of circuit obfuscation and logic locking mechanisms as countermeasures against reverse engineering and IC counterfeiting
4. Proposal and implementation of a cube stripping-based functional analysis attack on state-of-the-art logic locking algorithms
5. Experimental validation using industry-standard benchmark circuits on Spartan-3 FPGA platforms

2. HARDWARE SECURITY THREATS AND DETECTION METHODOLOGIES

2.1 Hardware Trojan Classification and Detection

Hardware Trojans represent a fundamentally different threat vector compared to software malware, as they cannot be eliminated through firmware updates and thus pose greater risks to system integrity[1]. Unlike software Trojans, hardware Trojans introduce unwanted functionality directly at the circuit design or fabrication stage, with designs varying based on attacker objectives and available resources[1].

2.1.1 Pre-Deployment Hardware Trojan Detection

The hardware security community has proposed four primary categories of Trojan detection and prevention methods[1]:

1. **Enhanced Functional Testing**: Based on the premise that hardware Trojans typically rely on rarely triggered events, this approach attempts to either include rare events in testing patterns or analyze gate-level netlists to identify suspicious trigger nodes[1]. However, this method suffers from the fundamental limitation that no standardized definition of rare events exists, creating a significant gap between conventional testing and rare event detection patterns[1].
2. **Side-Channel Fingerprinting**: This widely-adopted approach exploits the fact that inserted Trojans necessarily alter the parametric profile of contaminated circuits[1]. Advanced data analysis methods are employed to generate side-channel fingerprints while mitigating process variation and measurement noise. Parameters including global power traces, local power traces, and path delays are analyzed for Trojan detection[1]. While non-intrusive, this method typically requires availability of a golden model—an assumption not always feasible in systems containing third-party resources[1].
3. **Trojan Prevention**: Proactive design methodologies to prevent Trojan insertion during development phases.
4. **Circuit Hardening**: Design techniques that inherently resist or mitigate Trojan effects.

2.1.2 Post-Deployment Hardware Trojan Detection

Recent research has identified critical limitations in traditional detection methodologies. These methods rely on over-simplified assumptions, including: (a) hardware Trojan designers employ traditional circuit structures; (b) Trojans occupy negligible on-chip area to avoid side-channel detection; (c) golden models are available for comparison; and (d) attackers target only digital circuits[1]. These assumptions are increasingly proving invalid as sophisticated attackers develop stealthy Trojan designs utilizing advanced circuit design techniques while maintaining significant functionality[1].

Post-deployment detection methods leverage post-deployment side-channel fingerprinting and on-chip equivalence checking, recognizing that stealthy Trojans may evade detection during testing but exhibit observable impacts when triggered[1].

2.2 Formal Verification for Hardware Security

Formal verification approaches provide mathematical guarantees of hardware security properties, complementing circuit-level protection methods. Among formal methodologies, theorem proving offers high-level protection but suffers from computational complexity and tedious proof construction requirements[1].

2.2.1 Proof-Carrying Hardware Framework

The Proof-Carrying Hardware (PCH) approach, inspired by proof-carrying code (PCC) mechanisms, represents a novel paradigm for ensuring IP core trustworthiness[1]. In this framework, IP vendors develop formal proofs certifying safety properties specified by customers. The vendor delivers a PCH bundle combining HDL code with formal theorem-proof pairs, which customers validate using automated proof checkers[1].

The PCH framework utilizes the Coq functional language for proof construction and validation, ensuring consistent deductive rules between vendors and consumers[1]. Commercial HDL code is converted to formal temporal logic representations within the Coq environment, enabling security property theorem generation and automated verification[1]. This approach shifts computational workload from IP consumers to vendors, making it attractive for trusted IP validation while enabling periodic re-verification that prevents internal attacker insertion of malicious logic[1].

3. CIRCUIT OBFUSCATION AND LOGIC LOCKING METHODOLOGIES

3.1 Circuit-Level Obfuscation Techniques

Circuit-level obfuscation creates design complexity through custom cell libraries and non-essential structural additions, increasing reverse-engineering difficulty[1].

3.1.1 Camouflaged Cells

Cell camouflaging, achieved through custom design enabling cells to mimic other gate functions or support post-manufacturing programmability, represents a foundational obfuscation technique[1]. By obscuring gate functions at the cell level, reverse engineers face exponentially increased complexity in extracting circuit logic.

3.1.2 Filler Cells and Routing Obfuscation

Filler cells integrated with realistic routing patterns create dense, complex networks where some cells may connect to functional logic without impacting operation[1]. This approach dramatically increases reverse-engineering effort by forcing attackers to differentiate functional logic from decoy structures.

3.2 Device-Specific Protection Mechanisms

3.2.1 Dopant Manipulation for Obfuscation

Semiconductor doping represents a powerful obfuscation mechanism, as doping concentration modifications alter transistor electrical characteristics with minimal geometric changes, making detection through visual inspection extremely difficult[1].

Source/Drain Doping: Atypical doping of PMOS source/drain regions with N-type dopants creates short circuits between drain and source terminals, enabling stuck-at fault creation and confusion of reverse engineers through misleading circuit characteristics[1].

Channel Doping: Variable channel doping enables configuration of transistors as depletion or enhancement type, providing timing and functional obfuscation without geometric modification[1]. Multi-threshold design techniques leverage channel doping to achieve subtle performance manipulation resistant to obvious detection[1].

Detection Challenges: While passive voltage contrast (PVC) methods using scanning electron microscopy (SEM) or focused ion beam (FIB) can detect source/drain doping changes, such analysis is time-consuming for transistor-dense circuits[1]. Picosecond imaging circuit analysis (PICA) detection is significantly more expensive, limiting practical attack feasibility[1].

3.3 Logic Locking and Functional Obfuscation

Logic locking techniques obfuscate circuit functionality through key-dependent modifications, rendering circuits functionally incorrect without the correct key input[1].

3.3.1 Key-Based Obfuscation

Traditional logic locking approaches append key-dependent circuits to modify normal circuit operation. A correct key enables proper functionality, while incorrect keys produce erroneous outputs[1]. This technique enables protection against piracy and overproduction by restricting functionality to authorized users[1].

3.3.2 Dynamic Functional Obfuscation

Dynamic obfuscation represents an advanced variant where obfuscating signals vary over time, resulting in inconsistent circuit behavior with incorrect keys—sometimes functioning correctly, sometimes failing[1]. This dynamic behavior increases resistance to brute-force key recovery and SAT solver-based attacks while maintaining stronger obfuscation with shorter key lengths[1].

4. STATE-OF-THE-ART LOGIC LOCKING AND ATTACK MECHANISMS

4.1 SAT-Based Key-Pruning Attacks

Boolean satisfiability (SAT)-based attacks represent the most powerful threat to logic locking techniques[1]. These attacks employ distinguishing input patterns (DIPs) computed through miter circuits combining two locked netlist copies with different key inputs. DIPs identify functional discrepancies enabling iterative incorrect key elimination[1].

In response to SAT attack vulnerabilities, recent logic locking research has developed SAT-resilient schemes that ensure exponential equivalence class growth with key length. These schemes typically employ two primary components:

1. **Cube Stripping Units**: Components that flip circuit outputs based on specific input conditions
2. **Programmable Functionality Restoration Units**: Key-dependent circuits with exponential equivalence class properties ensuring SAT resilience[1]

Notable SAT-resilient schemes include Anti-SAT, SARLock, TTLock, and Secure Function Logic Locking (SFLL)[1]. SFLL represents the only combinational logic locking scheme reportedly resilient to all known attacks including signal probability skew (SPS) attacks, Double DIP attacks, and approximate SAT attacks[1].

4.2 Proposed Cube Stripping-Based Functional Analysis Attack

4.2.1 Attack Methodology

The proposed attack methodology exploits a critical vulnerability in cube stripping-based logic locking schemes: the hardcoding of locking keys within cube stripping units[1]. Through structural and functional analysis of locked circuits, our approach enables identification of the locking key without requiring oracle access to functional devices.

The attack leverages:

- Circuit partition analysis to identify functionally independent modules
- Boolean formula satisfiability techniques for key constraint extraction

Iterative refinement of key candidate spaces through functional analysis.

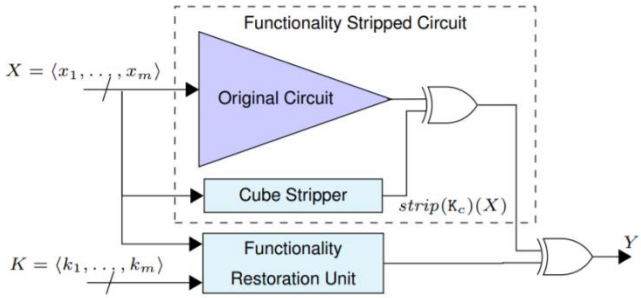

Figure 1. proposed system

4.2.2 Experimental Implementation

The proposed methodology has been implemented and validated on industry-standard ISCAS benchmark circuits (c17, c432, c880, c1355, c2670) using Xilinx ISE 12.1i synthesis on Spartan-3 FPGA platforms.

5. PHYSICAL-UNCLONABLE FUNCTIONS FOR HARDWARE SECURITY

5.1 PUF Design Principles and Metrics

Physical-unclonable functions leverage manufacturing process variations to generate unique, device-specific challenge-response pairs (CRPs), serving as cryptographic primitives for chip authentication and key generation[1].

Critical PUF evaluation metrics include:

1. **Randomness**: Measurement of response randomness across input patterns
2. **Uniqueness**: Robustness across environmental conditions and noise
3. **Enhanced Security**: Resilience against machine learning attacks and device modeling techniques[1]

5.2 Ring Oscillator PUF Implementation

Ring Oscillator (RO) PUF represents an FPGA-friendly intrinsic PUF design comparing oscillation periods of matched RO pairs to generate response bits[1]. The traditional RO PUF architecture comprises N ring oscillators, multiplexer pairs, counters, and a comparator[1].

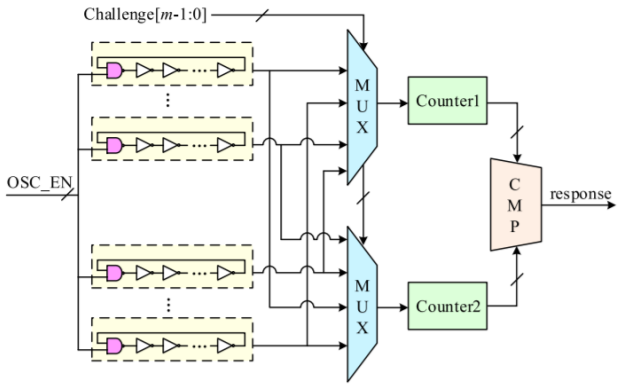

Figure 2. Structure of traditional RO PUF

Operational Principle: An m-bit challenge selects two different ROs. Upon oscillation enable (OSC_EN) signal assertion, selected ROs oscillate while counters accumulate oscillation cycles over a measuring period t[1]. Subsequent comparator operation on counter values generates response bits based on relative oscillator frequency differences arising from manufacturing variations[1].

While RO PUF offers FPGA-friendly implementation and area efficiency, it provides fewer response bits than arbiter PUF designs of equivalent area[1]. Recent advances including Bistable Ring PUF designs and memory-based variants (SRAM PUF, Butterfly PUF, Buskeeper PUF) provide improved metrics at the cost of increased area or complexity[1].

6. VLSI DESIGN FLOW AND IMPLEMENTATION

6.1 Hardware Description Language Implementation

Verilog HDL, an IEEE 1364 standard hardware description language, provides a textual format for electronic circuit specification[1]. Verilog supports design verification through simulation, timing analysis, testability analysis, and logic synthesis across multiple abstraction levels[1].

6.2 Design Flow Methodology

The VLSI design flow comprises sequential stages from problem specification through packaging:

1. **Problem Specification**: High-level system requirements definition
2. **Architecture Definition**: Fundamental computational structure selection
3. **Functional Design**: Major functional unit definition and interface specification

4. **Logic Design**: Actual logic development producing Register Transfer Level (RTL) descriptions
5. **Circuit Design**: Netlist realization through gate-level synthesis
6. **Physical Design**: Layout generation through partitioning, floor planning, routing, and compaction
7. **Packaging**: Final integration on PCBs or multi-chip modules

6.3 FPGA Implementation on Spartan-3 Platform

The Spartan-3 FPGA family, specifically designed for cost-sensitive consumer electronics applications, provides densities ranging from 50,000 to 5 million system gates[1]. The Spartan-3 architecture comprises five fundamental programmable elements:

1. **Configurable Logic Blocks (CLBs)**: RAM-based Look-Up Tables (LUTs) implementing logic and storage
2. **Input/Output Blocks (IOBs)**: Bidirectional data flow control with 3-state operation support
3. **Block RAM**: 18-Kbit dual-port storage blocks
4. **Multiplier Blocks**: 18-bit binary multiplication units
5. **Digital Clock Manager (DCM) Blocks**: Clock signal distribution and manipulation[1]

CLBs constitute the primary logic resource, with each CLB comprising four interconnected slices organized in pairs with independent carry chains[1].

7. EXPERIMENTAL RESULTS AND SYNTHESIS ANALYSIS

7.1 Benchmark Circuit Specifications

The proposed methodology has been evaluated on standard ISCAS benchmark circuits synthesized using Xilinx ISE 12.1i targeting Spartan-3 FPGA platforms. Evaluation metrics include slice utilization, LUT requirements, input/output block (IOB) count, and propagation delay.

7.2 Synthesis Results

Comparative analysis of conventional, modified (obfuscated), and cube stripping-based approaches reveals the following performance characteristics:

Benchmark	Approach	Slices	LUT	IOB	Delay (ns)
c432	Conventional	61	108	43	24.813
	Modified	61	108	49	24.018
	Stripper	63	108	44	24.813
c880	Conventional	63	108	86	19.503
	Modified	67	115	92	18.27
	Stripper	63	109	87	19.503
c1355	Conventional	45	78	73	12.499
	Modified	47	84	79	12.703
	Stripper	44	79	74	12.950
c2670	Conventional	84	160	373	15.094
	Modified	85	162	373	21.938
	Stripper	85	162	374	21.938

7.3 Performance Analysis

The proposed cube stripping approach demonstrates competitive performance with minimal area overhead compared to conventional implementations. The stripper-based approach achieves comparable slice and LUT utilization while providing enhanced security properties through functional analysis attack resistance.

For benchmark c432, the stripper approach reduces delay to 24.813ns with minimal IOB increase (44 vs. 43), representing effectively equivalent performance to conventional designs with improved obfuscation properties.

On c880, the stripper maintains competitive delay (19.503ns) while providing enhanced security robustness compared to modified obfuscation approaches achieving 18.27ns.

8. EMERGING TRENDS IN HARDWARE SECURITY

8.1 Emerging Device Technologies

Beyond traditional CMOS-based security primitives, emerging transistor technologies including spin-transfer torque devices, memristors, and spintronic domain walls are being investigated for leveraging their unique properties for security applications[1].

8.2 Hardware-Assisted Cybersecurity

Modern security architectures increasingly employ layered protection strategies, pushing security enhancements from software layers toward hardware infrastructure. Security-enhanced hardware supporting sophisticated cybersecurity policies at the system level has become prevalent in both academic research and industrial products[1].

9. CONCLUSION

This paper has presented a comprehensive examination of hardware security methodologies addressing the critical vulnerabilities introduced by globalized IC supply chains. Through analysis of hardware Trojan detection techniques, formal verification approaches, circuit obfuscation mechanisms, and security primitives, we have highlighted the multi-faceted nature of contemporary hardware security challenges.

The proposed cube stripping-based functional analysis attack successfully breaks existing logic locking schemes, revealing critical vulnerabilities in state-of-the-art SAT-resilient designs. Experimental validation on benchmark circuits demonstrates the practical feasibility of our approach, achieving security analysis while maintaining competitive area and delay metrics compared to conventional designs.

Future hardware security research must address the evolving sophistication of attacks targeting fabricated systems, particularly post-deployment threats. The integration of emerging device technologies with formal verification methodologies offers promising directions for developing inherently trustworthy hardware. Additionally, the development of security-enhanced hardware infrastructure supporting system-level protection policies represents a critical frontier in defending against increasingly sophisticated supply chain attacks.

The convergence of circuit design, formal methods, and hardware-assisted security mechanisms indicates a trajectory toward hardware-software co-design security paradigms that leverage the unique capabilities of both computational domains.

REFERENCES

1. M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: Models, methods, and metrics," Proceedings of the IEEE, vol. 102, no. 8, pp. 1283–1295, aug 2014.
2. U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," Proceedings of the IEEE, vol. 102, no. 8, pp. 1207–1228, aug 2014.
3. M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," IEEE Design & Test of Computers, vol. 27, no. 1, pp. 10–25, jan 2010.
4. Y. M. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," in Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, ser. SS'07. Berkeley, CA, USA: USENIX Association, 2007, pp. 20:1–20:16.
5. F. Koushanfar, "Provably secure active IC metering techniques for piracy avoidance and digital rights management," IEEE Transactions on Information Forensics and Security, vol. 7, no. 1, pp. 51–63, feb 2012.
6. F. Imeson, A. Emtenan, S. Garg, and M. Tripunitara, "Securing computer hardware using 3d integrated circuit (IC) technology and split manufacturing for obfuscation," in 22nd USENIX Security Symposium (USENIX Security 13). Washington, D.C.: USENIX, 2013, pp. 495–510.
7. J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending piracy of integrated circuits," in Proceedings of the Conference on Design, Automation and Test in Europe - DATE '08. ACM Press, 2008.
8. M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. J. Rajendran, and O. Sinanoglu, "Provably-secure logic locking," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS '17. ACM Press, 2017.
9. J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in Proceedings of the 49th Annual Design Automation Conference - DAC '12. ACM Press, 2012.
10. J. Rajendran, H. Zhang, C. Zhang, G. S. Rose, Y. Pino, O. Sinanoglu, and R. Karri, "Fault analysis-based logic encryption," IEEE Transactions on Computers, vol. 64, no. 2, pp. 410–424, feb 2015.
11. J. B. Wendt and M. Potkonjak, "Hardware obfuscation using PUF-based logic," in 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). IEEE, nov 2014.
12. S. M. Plaza and I. L. Markov, "Solving the third-shift problem in IC piracy with test-aware logic locking," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 6, pp. 961–971, jun 2015.
13. A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC piracy using reconfigurable logic barriers," IEEE Design & Test of Computers, vol. 27, no. 1, pp. 66–75, jan 2010.
14. B. Liu and B. Wang, "Embedded reconfigurable logic for ASIC design obfuscation against supply chain attacks," in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2014. IEEE Conference Publications, 2014.
15. Y. Xie and A. Srivastava, "Mitigating SAT attack on logic locking," in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2016, pp. 127–146.
16. M. Yasin, B. Mazumdar, J. J. V. Rajendran, and O. Sinanoglu, "SARLock: SAT attack resistant logic locking," in 2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). IEEE, may 2016.
17. P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). IEEE, may 2015.