CEP v2.0 Security Evaluation Targets

Brendon Chetwynd
brendon.chetwynd@ll.mit.edu

Kevin Bush
kevin.bush@ll.mit.edu

Kyle Ingols
kwi@ll.mit.edu

MIT Lincoln Laboratory

August 29, 2019

1 Background

The Common Evaluation Platform (CEP) was developed to enable and facilitate the evaluation of various integrated circuit (IC) security-enhancing design and fabrication techniques across a variety of DoD sponsored research and development programs. The CEP is a mission-relevant and license-unencumbered System on Chip (SoC) design with representative scale and features such that it can serve as a surrogate for trusted US Government designs. The CEP is an entirely open-source benchmark design that features:

- **Scale:** Sufficient SoC complexity to stress and challenge defensive design techniques
- **Diversity:** DoD Mission-relevant surrogate modules that offer diversity of digital computation functions
- **Releasability:** Open-source license compatibility permits free distribution to any performer seeking to evaluate a defensive technique
- **Extensibility:** Modular approach to design that offers easy adaptation to meet emerging and future evaluation objectives

The CEP features (1) accelerators for common DoD-relevant mission functions including digital signal processing (DSP) and secure communications (cryptography); (2) a verification test suite for validating that baseline functionality and performance is unperturbed by any security-enhancing technique; and (3) annotated and labeled security-sensitive design elements. These security-sensitive design elements will be referred to as security evaluation targets (SETs) throughout the remainder of this document and will serve to help define a basis for evaluation of defensive design transformation techniques, as well the development of security metrics.

As of the time of this writing, the CEP has evolved through multiple revisions. It was initially based on an OR1200 processor. Further revisions migrated to a mor1k processor. The most recent release – CEP v2.0 – migrated to a RISC-V platform leveraging the UC Berkeley Rocket-Chip [2]. With each revision, additional features have been added and test suites and documentation expanded. CEP v2.0 benefits the US Government by enabling risk-reducing collaborations across Government and academia, by facilitating more rigorous evaluation of defensive techniques, and by enabling the developmental test and evaluation of those techniques by US Government performers.

In summary, the CEP is a DoD-relevant surrogate SoC for IC security technology assessments, and the SETs within the CEP serve as the basis for evaluating the *performance and* efficacy of those security-enhancing technologies.

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.

This material is based upon work supported under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Air Force.

1.1 Security Reference Model

As part of a suite of on-going US Government security-focused programs, an *On-Chip Security Reference Model* was developed to organize important and pressing security challenges of integrated circuit design and manufacturing. The reference model includes the following four challenges: (1) Malicious Hardware; (2) Reverse Engineering; (3) Side Channels; and (4) Supply Chain. We summarize this reference model and expand on it with derived security objectives in Table 1.

On-chip Security Reference Model	**Derived Security Objectives**
Malicious Hardware: The insertion of hidden functionality that can be secretly triggered to deliver disruptive payloads (also known as: ”Hardware Trojans”).	**Design Integrity:** US Government seeks to make malicious modifications either *intractable* or readily *detectable* via inspection.
Reverse Engineering: The interpretation of design intent or implementation from available or derived representation to understand secret or proprietary algorithms.	**Design Confidentiality:** US Government seeks to make the design either *intractable* to reverse engineer or *unusable* to non-approved users.
Side Channel: The extraction of secret or proprietary information from the integrated circuit through communication channels other than those intended by the design.	**Data Confidentiality:** US Government seeks to protect data running on the integrated circuit, protecting the data *in situ*.
Supply Chain: The risk of non-genuine integrated circuits sold as real, but realized through cloning, counterfeiting, recycling, remarking, etc.	**Device Integrity:** US Government seeks to detect and prevent deployment of *non-authentic* (non-genuine) parts.

Table 1: Security Reference Model and derived Security Objectives

These four derived security objectives (design integrity, design confidentiality, data confidentiality, and device integrity) will be used to refine specific security evaluation targets in the following sections.

2 Security Evaluation Targets

The security evaluation targets (SETs) comprise specifically identified components within the CEP along with suggested scopes of protection. These scopes are intended to provide an increasing scale of how much of the SET needs to be protected to achieve a given security objective. Each security objective is addressed in its own section below. The identified components are intended to be representative, but not necessarily exhaustive; users of the CEP may reasonably identify additional design components that require protection.

2.1 Design Integrity

The Design Integrity objective captures the scenario when the US Government seeks to make malicious modifications to its integrated circuits either *intractable* or readily *detectable* via inspection. This is more colloquially known as the “Hardware Trojan” problem. These attacks can be launched from both the design phase (e.g. malicious insider, compromised synthesis tools, etc.) and the manufacturing phase (e.g. untrustworthy fabrication center, packager, etc.). We focus the CEP SETs for design integrity on the issue of fabrication-phase attacks.

Fabrication-phase design integrity attacks can be characterized as any malicious deviation from a trusted design carried out during the fabrication process. This can include, for example, modifying a physical design file (GDS II), adding lithography masks, replacing masks, or substituting masks.

Table 2 presents the selected design integrity SETs within the CEP as previously described in [1]. Depending on the defensive technique being explored, a user of the CEP may wish to apply the transformation to the relevant SET within the Chisel code, the generated Verilog, or in downstream synthesis products.

Module	File name	Description	Consequence if Compromised
CSRFile	`<CEPROOT>/hdl/cores/freedom/rocket-chip/src/main/scala/rocket/CSR.scala`	Control Status register file	Privilege escalation
DecodeUnit	`<CEPROOT>/hdl/cores/freedom/rocket-chip/src/main/scala/rocket/Decode.scala`	Instruction decoder	Execute privileged instructions, inject code
Frontend	`<CEPROOT>/hdl/cores/freedom/rocket-chip/src/main/scala/rocket/Frontend.scala`	Program counter manager	Compromise control flow
PTW	`<CEPROOT>/hdl/cores/freedom/rocket-chip/src/main/scala/rocket/PTW.scala`	Page table walker	Access privileged pages
TLB	`<CEPROOT>/hdl/cores/freedom/rocket-chip/src/main/scala/rocket/TLB.scala`	Virtual to physical address translation	Access privileged pages, expose physical addresses

Table 2: Design Integrity SETs as described in [1]

Level	Encoding	Name	Abbreviation
0	00	User / Application	U
1	01	Supervisor	S
2	10	Hypervisor	H
3	11	Machine	M

Table 3: Rocket Chip privilege modes as described in [3]

Level	Scope	Description
0	Bits	Select bits within the SET (e.g., privilege register)
1	Bits & Logic	Select bits and the immediate logic that controls the setting and clearing of said bits
2	Bits, Logic, & Fan-out	In addition to the bits themselves and the immediate fan-in logic, the fan-out logic for design elements that consume the SET bits should be considered
3	Bits, Logic, Fan-out, & Fan-in	Expand the scope of protection by adding a greater amount of fan-in logic for the SET bits.
4	Module	Protect the entire module containing the SET
5	Integrated Circuit	Protect the entire integrated circuit

Table 4: Scopes of Design Integrity

For example, in `CSR.scala`, the privilege bit is contained within the `MStatus` class under the `prv` field. The generated verilog has a `reg_mstatus_prv` register that is initialized to Machine Mode (`2'h3`). Table 3 presents the privilege mode encoding [3]. Selective flipping of a single bit would result in an unexpected privilege escalation.

When considering the protection of a given SET, one needs to asses how much needs to be protected. In the case of the privilege register, is it sufficient to protect the register bits? Should the output be protected such that consumers of the privilege bit are getting authentic values? Perhaps the upstream fan-in logic should be protected? We therefore present Table 4 which provides examples of which aspects of a given SET should be protected with a notion of increasing scopes of protection.

To assist with dependency analysis, MIT-LL has developed and released *Bomberman*, a tool that can create signal-level data-flow graphs of entire circuit designs from behavioral HDL (Verilog). Using Bomberman, one can generate data-flow graphs of cores within the CEP to perform fan-in and fan-out static analyses of SETs . For more information on how to use Bombermans data-flow graph generation tool, please refer to the Bomberman repository: `https://github.com/mit-ll/bomberman`.

2.2 Design Confidentiality

For the Design Confidentiality objective, the US Government seeks to protect intellectual property and deny an adversary's ability to reverse-engineer or mis-use a design. Should the adversary be successful, their goals may include creating *counterfeits* of the design or analyzing the design for *vulnerabilities*. In the case of the former, a counterfeit design may not be of the same quality as the original part or could contain a hardware Trojan, thus violating both the *Design Integrity* and *Device Integrity* objectives. In the case of the latter, the adversary can analyze the design for vulnerabilities. In addition to the aforementioned goals, an adversary may choose to duplicate a design for the purpose of gaining a capability that they previously lacked, thus reducing or eliminating the *technological advantage* the US Government gained in creating the original design.

Module	File name(s)	Description	Consequence if Compromised
FIR Filter	`<CEPROOT>/generated_dsp_code/FIR_filter.v`	Finite Impulse Response Filter	Loss of a critical technology

Table 5: CEP Design Confidentiality SETs

Table 5 presents the selected Design Confidentiality SETs within the CEP. They have been chosen from the Digital Signal Processing (DSP) class of functions and are representative of designs elements in communication, sensor-based, and navigation aid systems. Additional options will be added in future releases of the CEP. Users of the CEP are reminded that these modules are *representative* modules; an FIR filter does not by itself warrant confidentiality protections, but a DoD-specific FIR filter may contain sensitive coefficients and customizations that do warrant protections.

Level	Scope	Description
0	Coefficients, Constants, and Taps	For SETs whose operations rely on designer-selectable constants, obfuscate these in the SET implementation or dynamically load them at run-time
1	SET Architecture	Even if the SET is known, the way in which it has been implemented has been hidden from the adversary
2	SET Identity	Obfuscate the identity of the entire SET within the IC
3	Intergrated Circuit	Hide the SET by obfuscating the entire IC design

Table 6: Example Scopes of Design Confidentiality

For each design confidentiality SET, the gradient security examples identified in Table 6 provide a increasing range of scope of what needs to be protected. For processor based systems, care must be taken to ensure the resident software does not negate the protection (e.g., FIR Filter Driver interacts with the unknown SET). Mitigations such as encrypted software stored in non-volatile memory should be considered.

2.3 Data Confidentiality

The Data Confidentiality objective is focused on protecting data processed within the IC. The identified class of threats are *Side Channel Attacks* in which an adversary uses unintended channels (timing, temperature, acoustics, electromagnetics, etc.) to extract critical information while the IC is operating. For example, an adversary may passively observe the electromagnetic emissions of an AES implementation within an FPGA. Through measurements and calculations, they can derive the AES key used [4].

Table 7 presents the selected Data Confidentiality SETs within the CEP. Cryptographic algorithms have been chosen as SETs for this objective given the amount of research being done extracting key material from

Module	File name(s)	Description	Consequence if Compromised
AES-192	<CEPROOT>/hdl/cores/aes/aes_192.v <CEPROOT>/hdl/cores/aes/round.v <CEPROOT>/hdl/cores/aes/table.v <CEPROOT>/hdl/cores/freedom/mitll-blocks/src/main/scala/aes.scala	AES-192 Encryption Core	Unauthorized data access

Table 7: CEP Data Confidentiality SETs

operating cryptographic circuits. Should a key be compromised, an adversary could potentially have unrestricted access to protected data in transit or data at rest, including current, past, and future transmissions.

Level	Scope	Description
0	Key Registers	Protect unencrypted keys stored within the target
1	Keys and Subkeys	Protect keys and interim key products (subkeys)
2	Entire SET	Protect all the key and data products as they reside within the SET
3	Data-path & Target	Protect the SET as well as the data-paths that transport the unencrypted key(s) to the SET
4	Data-path, Target, & Keystore	Protect the SET as well as the data-paths that transport and store both unencrypted and encrypted key(s)
5	Integrated Circuit	Protect all data within the integrated circuit

Table 8: Example Scopes of Data Confidentiality

As with the other objectives, a range of protection options have been identified. Table 8 presents these options ranging from just protecting the unencrypted keys at rest to protecting the entire integrated circuit.

2.4 Device Integrity

The Device Integrity objective ensures the IC is authentic through the validation of some form of pedigree. An adversary may attempt to introduce cloned or counterfeited parts to undermine legitimate manufacturers and steal profits for themselves, or to produce parts with reduced reliability that affect an adopting system's *availability*.

Unlike the other objectives, device integrity applies to the whole IC and thus the entire IC serves as the SET. While there are multiple techniques for providing pedigrees (e.g., watermarking, authentication of trusted elements, etc.), it is assumed that the efficacy of a particular technique is a function of an adversary's ability to replicate it.

3 Summary

The Common Evaluation Platform (CEP) was created to facilitate the development and evaluation of IC security-enhancing techniques across a range of US Government DoD programs. This paper describes four security objectives that are derived from an On-Chip Security Reference Model.

It is important to note that these objectives can overlap. For example, an adversary may be able to circumvent a design confidentiality mechanism to create a counterfeit part, thus violating the device integrity and potentially design integrity objectives.

The Common Evaluation Platform (CEP) has been presented as a relevant DoD surrogate SoC that can serve as a basis for evaluation tools and techniques developed to meet the aforementioned security objectives.

References

[1] Timothy Linscott, Pete Ehrett, Valeria Bertacco, and Todd Austin. 2018. SWAN: Mitigating Hardware Trojans with Design Ambiguity . In IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN (ICCAD 18), November 58, 2018, San Diego, CA, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3240765.3240854

[2] Krste Asanovi, Rimas Aviienis, Et al., The Rocket Chip Generator, Technical Report UCB/EECS-2016-17, EECS Department, University of California, Berkeley, April 2016. http://www.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-17.html

[3] Editors: Andrew Waterman1, Krste Asanovic. 8 June, 2019. The RISC-V Instruction Set Manual, Volume II: Privileged Architecture, Document Version: 20190608-Priv-MSU-Ratified. https://riscv.org/specifications/privileged-isa/

[4] Vincent Carlier and Herve Chabanne and Emmanuelle Dottax and Herve Pelletier. 2004. Electromagnetic Side Channels of an FPGA Implementation of AES. CRYPTOLOGY EPRINT ARCHIVE, REPORT 2004/145.