AI Hardware Security Compliance Portfolio

Industry-ready portfolio aligned to compliance, certification, and audit workflows

Prepared for: Brojogopal Sapui
Date: 6 April 2026

Portfolio purpose	Translate PhD research into the language used by compliance engineers, certification teams, auditors, and hiring managers.
Primary frameworks	ISO/SAE 21434 style cybersecurity work products; Common Criteria style evaluation packages; audit-oriented evidence and traceability.
Technical domains covered	Side-channel analysis, fault injection, secure accelerator design, hardware verification, masking, device-level security.
Important note	This portfolio is a professionally structured alignment document. It is not an official certification submission and does not claim formal assessor approval.

Confidential working portfolio - prepared for professional positioning and interview use.

Contents

1. Executive overview

This portfolio converts a deep technical PhD body of work into a set of compliance-oriented deliverables that resemble how employees document cybersecurity engineering inside semiconductor, automotive, embedded-AI, and certification-facing organizations.

The central idea is simple: the same research that demonstrates attacks, defenses, architectures, and validation results can be reframed as threat analysis, vulnerability analysis, control design, verification evidence, residual risk, and audit traceability.

The portfolio focuses on four core research streams: side-channel attacks on Hyperdimensional Computing (HDC) accelerators, fault injection campaigns and countermeasures, secure AI accelerator design and verification, and device-level security work on emerging hardware technologies.

It is intentionally written in a style that can be understood by three audiences at once: hiring managers, compliance engineers, and technically deep reviewers.

- Use this document as a professional attachment for job applications, interviews, networking follow-up, or performance review discussions.
- Use the project dossiers as templates for converting any future paper or prototype into compliance-language work products.
- Use the appendices as a quick reference for how engineers talk about evidence, traceability, assurance, and residual risk.

2. How to read this portfolio

Each project dossier follows the same structure. First, the technical system is described. Second, the sensitive assets are identified. Third, the credible attacker actions are described. Fourth, the vulnerabilities and attack surfaces are stated in language suitable for risk and audit workflows. Fifth, the implemented or proposed controls are documented. Sixth, the validation evidence is summarized. Finally, the residual risk and next-step recommendations are recorded.

A compliance portfolio is not a replacement for technical depth. It is the packaging layer that proves the technical work can be governed, reviewed, and defended inside a real company process.

- When a sentence says 'aligned to' a framework, it means the work can be mapped into the structure of that framework.
- When a sentence says 'evidence', it means the output that a reviewer can inspect: logs, reports, traces, test results, scripts, verification artifacts, or measured data.
- When a sentence says 'residual risk', it means the risk that remains after a control is applied.

3. Professional profile and positioning

Primary identity: hardware and AI security researcher with deep hands-on experience in side-channel analysis, fault injection, secure accelerator design, and hardware-aware countermeasure development.

Professional differentiation: unlike many compliance candidates who mainly understand documentation, this profile combines attack realism, implementation detail, and measurement-backed validation. That is valuable because most certification and audit programs fail when the underlying technical reasoning is weak.

Industry translation: this profile fits roles such as AI Hardware Security Engineer, Security Certification Engineer, Product Security Architect, Security Evaluation Engineer, Hardware Threat Modeling Engineer, Security Compliance Engineer, and ML Platform Security Engineer.

- Strength 1 - ability to model practical attackers rather than paper-only threats.
- Strength 2 - ability to implement countermeasures and measure their effect.
- Strength 3 - ability to connect hardware design decisions to evidence that auditors can review.
- Strength 4 - ability to operate across device, circuit, architecture, FPGA, and system-validation layers.

4. Compliance lens used in this document

This document uses an engineering-oriented compliance lens instead of a legal or policy-only lens. The emphasis is on security work products that employees actually create: threat analysis, risk narratives, architecture notes, control descriptions, verification plans, leakage assessments, evidence tables, and traceability matrices.

Two framework families are emphasized because they are highly relevant to the user's target roles. The first is ISO/SAE 21434 style cybersecurity engineering, useful for structured threat analysis and validation in automotive or edge products. The second is Common Criteria style evaluation thinking, useful for security targets, assurance reasoning, evaluator communication, and structured claims about a target of evaluation.

The document also adopts an audit mindset. Auditors typically ask three questions: what is the claim, what is the evidence, and how do you know the evidence actually supports the claim. Every major section below is designed to answer those questions.

5. Portfolio architecture and evidence model

The portfolio is organized around the progression research -> threat -> vulnerability -> control -> evidence -> residual risk -> role relevance.

That progression mirrors how employees in mature security organizations communicate across engineering, product, compliance, and assurance teams. A technically strong employee is especially valuable when they can translate raw research outputs into this operational chain without losing fidelity.

- Research output: paper, experiment, prototype, RTL block, simulation result, or device model.
- Threat statement: what an attacker is trying to achieve and under what assumptions.
- Vulnerability statement: what in the design enables the attack or increases feasibility.
- Control statement: what was implemented or proposed to reduce the risk.
- Evidence statement: what measured result or artifact supports that the control works.
- Residual risk statement: what remains unresolved and what future improvement is required.

Portfolio summary table

Project stream	Primary asset	Main threat	Representative control	Key evidence
HDC side-channel security	Stored class hypervectors / model IP	Passive power-based model extraction	Dynamic masking	Attack accuracy reduction and TVLA
Fault injection campaigns	Inference correctness / decision integrity	Voltage glitching and timing disturbance	Timing-aware protection and validation	Misclassification campaigns and mitigation observations
Secure accelerator design	Runtime data path and memory movement	Leakage and exploitable micro-architectural behavior	Secure design choices + verification	RTL evidence, simulation, profiling
Device-level security	Low-level device behavior and entropy/security primitives	Leakage, predictability, exploitable physical behavior	Bias masking / secure circuit concepts	Modeling and architecture argumentation

6. Project dossier A - HDC side-channel security

This dossier is based on the FPGA-oriented HDC side-channel work in which deep-learning-assisted analysis was used to extract information from an HDC accelerator implementation. The technical storyline is highly valuable for compliance roles because it already contains all core elements of a mature security case: asset identification, realistic attacker access, measured evidence, targeted countermeasure design, and post-control evaluation.

The system context is an FPGA-based HDC accelerator performing encoding and similarity computation with stored class hypervectors. The protected asset is not merely data in memory; it is trained model intellectual property. That framing matters because companies care deeply about IP theft, competitive differentiation, and model reconstruction risk.

6.1 Asset definition and business relevance

The principal asset is the class hypervector set stored in associative memory. Those hypervectors encode the trained model and therefore capture proprietary inference capability. In a company setting, that means unauthorized recovery could enable model cloning, reverse engineering, performance replication, and loss of product differentiation.

A secondary asset is the integrity of the inference pipeline itself, because leakage often exposes internal computation phases that can later be exploited for higher-confidence attacks or tuning of attack windows.

6.2 Threat scenario

A realistic adversary obtains physical or near-physical measurement access to the FPGA device and records power traces during inference. The adversary does not require source code modification or logical compromise. Instead, the attack is passive and relies on observing leakage from normal device operation.

The attacker goal is model extraction. More specifically, the attacker aims to infer bits of stored hypervectors by learning patterns in side-channel traces. The work shows that a machine-learning-based attacker can do this effectively when trace quality and training are sufficient.

6.3 Vulnerability analysis

The vulnerability is rooted in observable side-channel leakage associated with HDC operations such as XOR and popcount-like behavior during similarity calculation. Even though HDC is high-dimensional and algorithmically different from traditional cryptographic targets, the implementation still emits exploitable physical signatures.

In compliance language, the issue is an absence of adequate confidentiality-preserving controls around sensitive intermediate computations. The design allows physical observables to correlate with protected internal values strongly enough for a trained classifier to exploit them.

6.4 Control design

The core control is dynamic masking integrated into the computation pipeline. This is important because it is not merely a theoretical recommendation; it is an implementation-level protection

that changes the observability of sensitive values. The control is also resource-conscious, which matters in edge and embedded deployments.

In portfolio language, this control demonstrates the ability to choose a mitigation that balances security gain with hardware cost. That trade-off thinking is highly valued in real product teams.

6.5 Validation evidence

The project includes attack efficacy measurements before control application and post-control measurements after masking. It also includes TVLA-oriented leakage reasoning to show that the control reduces observable leakage rather than simply altering attack conditions.

This is exactly the type of evidence that audit and certification workflows need: a measurable baseline, a documented intervention, and a measurable result after the intervention.

6.6 Residual risk and improvement path

Residual risk remains because masking and unmasking points can still create opportunities for higher-order leakage or refined attack strategies. The portfolio therefore does not overclaim. Instead, it records the remaining concerns and proposes future hardening such as more comprehensive masked arithmetic or stronger implementation discipline.

That honest residual-risk statement increases credibility. Companies and evaluators trust engineers more when they know where the current boundary of assurance lies.

6.7 Compliance-style work products derived from the project

- Threat analysis and risk statement for model extraction via power side-channel leakage.
- Control description for dynamic masking and runtime protection of sensitive hypervectors.
- Validation memo summarizing attack results before and after control implementation.
- Evidence register containing trace captures, training configuration, evaluation logs, and leakage-assessment outputs.
- Residual-risk note and recommended next engineering actions.

6.8 Example portfolio wording for interviews and resumes

- Conducted threat analysis and vulnerability evaluation of FPGA-based AI accelerator against passive side-channel model extraction.
- Designed and evaluated a runtime masking control to reduce leakage from sensitive HDC operations.
- Produced audit-friendly evidence using measured traces, machine-learning attack evaluation, and leakage assessment.

7. Project dossier B - fault injection and robustness

This dossier converts the user's fault injection and glitching-related work into compliance language. In many product organizations, side-channel leakage and fault injection are treated as sister threat classes because both involve physical interaction with hardware and both can undermine security claims if not systematically evaluated.

The practical strength of this body of work is that it moves beyond abstract fault models. It includes hands-on use of equipment, timing-window reasoning, campaign design, and effect observation on the target system.

7.1 Protected assets and security objective

The primary protected asset is inference correctness. A physical attacker who can induce carefully timed faults may force misclassification, bypass intended behavior, or reveal implementation weaknesses.

A secondary protected asset is trust in system robustness. Safety- and security-sensitive applications require confidence that outputs remain reliable under perturbation or that attacks are at least detected.

7.2 Threat scenario

The attacker uses voltage glitching or timing disturbance, supported by equipment such as ChipWhisperer, to influence operations at precisely selected time windows. The attacker goal may be misclassification, differential behavior, security-boundary bypass, or identification of timing-sensitive critical operations.

In compliance language, this is a physical integrity attack against runtime execution.

7.3 Vulnerability analysis

The vulnerability often arises from timing-sensitive datapaths, insufficient detection of abnormal operating conditions, or a lack of redundancy/monitoring in critical operations. From a certification perspective, the key point is whether the system can tolerate, detect, or bound the effect of a disturbance.

This project stream is valuable because it builds intuition for where a hardware design becomes fragile under active manipulation.

7.4 Control strategies and engineering responses

Potential or explored controls include timing hardening, redundancy, fault detection sensors, protected reset/exception paths, conservative clocking margins, and architecture-level strategies that reduce single-point fault leverage.

Even where a control is not yet fully productized, the portfolio can still show that the engineer understands how to convert attack insights into product requirements.

7.5 Evidence and assurance value

Fault campaign logs, trigger configurations, successful/unsuccessful glitch windows, observed output deviations, and countermeasure comparisons all become evidence artifacts. Those artifacts are extremely useful in design reviews because they show that robustness claims were challenged experimentally rather than assumed.

Employees who can produce this evidence are useful not only for security roles but also for reliability, product assurance, and pre-silicon/post-silicon validation teams.

7.6 Compliance-language summary

- Performed active physical attack assessment focused on disturbance-based integrity compromise.
- Characterized timing-sensitive windows and documented attack feasibility under realistic lab conditions.
- Mapped observed weaknesses to candidate design controls and validation requirements.

8. Project dossier C - secure HDC accelerator design and verification

This dossier covers the user's broader accelerator-design work, including secure implementation of HDC components, architecture decisions, verification planning, and threat-aware design evolution. For compliance portfolios, this is important because companies do not only hire people who can break systems; they need engineers who can embed security requirements during design and verify them systematically.

The portfolio framing here is that the engineer understands how to move from attack observation to secure architecture, then to verification, and finally to evidence suitable for internal review.

8.1 System context

The system is an HDC accelerator with encoding, storage, and similarity computation components. Security relevance arises because the architecture processes sensitive representations and may be deployed in embedded or edge settings where physical exposure is plausible.

Architectural choices such as where data is stored, how operations are partitioned, and what timing structure exists can directly affect leakage and fault sensitivity.

8.2 Design responsibilities that map to compliance work

Threat-aware architecture selection, secure module decomposition, sensitivity-aware memory organization, measurement hooks for validation, and explicit verification planning all map well to formal security engineering responsibilities.

In hiring terms, this means the portfolio demonstrates not just reactive security evaluation but proactive secure-design thinking.

8.3 Verification and assurance view

Verification assets can include RTL simulations, directed tests, measurement triggers, profiling infrastructure, equivalence checks, and structured evaluation criteria for what counts as acceptable leakage or unacceptable behavior.

A compliance-minded engineer is expected to define what evidence must exist before a security claim is accepted. This project stream supports exactly that expectation.

8.4 Example control statements

Sensitive computation should be minimized, randomized, masked, partitioned, or otherwise structured such that a physical attacker cannot reliably infer protected data.

Critical runtime behavior should be instrumented or monitored in ways that support both debugging and assurance without creating new leakage channels.

8.5 Portfolio value for employers

- Can speak with both design engineers and evaluators using a common evidence vocabulary.
- Can help write security requirements that are specific enough to verify.

- Can bridge architecture, RTL, measurement, and security-review conversations.

9. Project dossier D - device-level and emerging hardware security

This dossier positions the user's device- and circuit-oriented work, including emerging technologies such as rFET-related security concepts, as part of a broader compliance and certification narrative. This is strategically valuable because many security portfolios stop at software or architecture level, whereas future hardware products increasingly need security reasoning that begins at the device and circuit layers.

The compliance translation is not that device-level research automatically equals certified security. Rather, it demonstrates the ability to reason from physical behavior upward: which device properties could support entropy, masking, obfuscation, or resistance to probing and leakage, and what evidence would be needed to justify those claims.

9.1 Relevance to certification and assurance

Certification teams often struggle when low-level security assumptions are vague. A candidate who can explain device behavior, model limitations, implementation trade-offs, and where circuit assumptions feed into system claims brings unusual value.

This work also supports future-facing product security because emerging devices require early-stage security thinking before standardized practice is mature.

9.2 Typical work products this stream can generate

Device-level threat notes, circuit-level security design rationales, characterization plans, modeling assumptions, and evidence roadmaps for what would be required before making a stronger product-security claim.

These artifacts matter in real companies because they help teams avoid overclaiming security based on incomplete low-level understanding.

9.3 How to present this work professionally

Frame it as security-enabling R&D: translating physical device behavior into candidate controls and into future assurance cases. Do not frame it as formal certified proof unless actual certification artifacts exist.

That balanced framing preserves credibility while still highlighting advanced expertise.

10. Cross-project traceability matrix

This matrix shows how multiple research streams can be organized into the kind of end-to-end logic that auditors and hiring managers expect.

Project	Asset	Threat	Key vulnerability	Representative control	Evidence
HDC SCA	Model IP in class hypervectors	Passive power-based extraction	Observable leakage during similarity operations	Dynamic masking	Measured attack reduction + leakage tests
Fault injection	Output integrity	Voltage/ timing glitching	Timing-sensitive execution windows	Detection/ hardening strategy	Campaign logs and output deviation analysis
Secure accelerator design	Runtime data flow	Leakage and exploitation opportunities	Security-unaware module structure	Threat-aware architecture + verification	RTL, simulation, profiling hooks
Device-level security	Low-level physical behavior	Predictable or exploitable device behavior	Insufficient characterization of low-level assumptions	Circuit/device masking and secure primitives	Modeling notes and characterization plans

Why this matters

Traceability is one of the biggest differences between a research profile and a compliance-ready profile. In a research profile, achievements are often organized by paper title or technical novelty. In a compliance-ready profile, the same work is organized by asset, risk, control, and evidence.

This section therefore functions as a portfolio backbone. It can be expanded for real product use into a living spreadsheet or requirement-management artifact.