Open Tools, Interfaces, and Metrics for Implementation Security Testing
https://optimist-ose.org/

Aydin Aysu (North Carolina State University)
Lejla Batina (Radboud University)
Eswari Devi N (Society for Electronic Transactions and Security - SETS, India)
Daniel Dinu (Intel)
Fatemeh Ganji (WPI)
Debdeep Mukhopadyay (IIT KGP)
Karel Král (Google)
Seyedmohammad Nouraniboosjin (WPI)
Stjepan Picek (Radboud University)
Markku-Juhani Saarinen (Tampere University)
Patrick Schaumont (WPI)
Caner Tol (WPI)
Marc Witteman (Keysight)

WORKING DOCUMENT
ACCELERATION OF AI FOR IMPLEMENTATION SECURITY TESTING

This document reviews the state of open tools, data and methods related to the use of AI acceleration platforms and AI algorithms for implementation security testing. The document also identifies the key areas for improvements and potential for standardization. Previous OPTIMIST documents already discuss the file format for side-channel traces, the capture interface, and PQC testing methods for implementation security testing campaigns.

Version History

0.5	7/8/25	First version based on Optimist Working Group Meetings

The document takes a dual view on hardware implementation security testing and AI. First, AI techniques can improve the analysis of designs for implementation security vulnerabilities. Second, AI implementations themselves may be susceptible to implementation security vulnerabilities. Either aspect of AI and hardware security is the subject of active research with many unresolved questions. The working group aims to identify basic reference materials for new researchers, topic trees and references for existing researchers, and opportunities for open tools, interfaces and metrics to drive AI for security/ security for AI.

I. Opening Talks

The plenary session of the working group (17 April 2025) includes three invited talks. The slides are available for download.

- Debdeep Mukhopadyay, "Side Channel and Fault Attack Testing of Cryptosystems in the view of Dr AI".
- Stjepan Picek, "Machine Learning-based Side-channel Analysis and Evaluation"
- Jakub Breier, "AI-accelerated Implementation Testing: Research vs Practice"

The talks emphasize AI techniques in the context of side-channel analysis and fault injection, although the speakers point to the broad application of AI techniques for implementation attacks (such as for example Trojan detection, and cybersecurity vulnerability detection/patching). Both profiled and non-profiled techniques are applicable, and profiled techniques can outperform known classic techniques provided that the trained AI model can be ported to the actual inference target. The speakers identify the following common challenges regarding AI for Implementation Security.

1. There is a need for guidelines to help security engineers apply AI to implementation attacks. Such guidelines should cover data pre-processing, methods to avoid overfitting, recommended architectures in terms of the use-cases, and evaluation of training data quality.
2. There is a need for guidelines to support security engineers in enabling portability, such as how a model trained on one target can be applied to a different target.
3. There is a need for a consensus on how to share and/or license trained ML models and datasets for implementation security testing.
4. There is a need for additional datasets that can be used as a reference to test the quality of AI based attacks, especially for side-channel analysis. Such datasets must prioritize portability, stronger countermeasures, different cryptographic ciphers, and different hardware targets.

References:

- V. Gohil, S. Patnaik, H. Guo, D. Kalathil, and J. Rajendran, "DETERRENT: Detecting Trojans using reinforcement learning," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 43, no. 1, pp. 57–70, Jan. 2024, doi: 10.1109/TCAD.2023.3309731.
- A. K. Zhang, J. Ji, C. Menders, R. Dulepet, T. Qin, R. Y. Wang, J. Wu, K. Liao, J. Li, J. Hu, S. Hong, N. Demilew, S. Murgai, J. Tran, N. Kacheria, E. Ho, D. Liu, L. McLane, O.

Bruvik, D.-R. Han, S. Kim, A. Vyas, C. Chen, R. Li, W. Xu, J. Z. Ye, P. Choudhary, S. M. Bhatia, V. Sivashankar, Y. Bao, D. Song, D. Boneh, D. E. Ho, and P. Liang, "BountyBench: Dollar impact of AI agent attackers and defenders on real-world cybersecurity systems," arXiv preprint arXiv:2505.15216, 2025. [Online]. Available: https://arxiv.org/abs/2505.15216

- S. Hajra, S. Chowdhury, and D. Mukhopadhyay, "EstraNet: An efficient shift-invariant transformer network for side-channel analysis," IACR Cryptology ePrint Archive, vol. 2023, no. 1860, 2023. [Online]. Available: https://eprint.iacr.org/2023/1860
- S. Saha, S. N. Kumar, S. Patranabis, D. Mukhopadhyay, and P. Dasgupta, "ALAFA: Automatic leakage assessment for fault attack countermeasures," in Proc. Design Automation Conf. (DAC), San Francisco, CA, USA, Jun. 2019, Art. no. 136, pp. 1–6, doi: 10.1145/3316781.3317763.

II. Fundamentals of AI

The working group collected the following pointers as a starting point for new researchers and practitioners in the field of AI for implementation security.

- Machine Learning Networks
  - Andrew Ng's Machine Learning (Coursera) — Lectures 1–3
  - Survey on using ML for SCA
- Training and Testing
  - Practice Template Attacks
  - Practice Profiling Attacks with Neural Networks
- Tools
  - PyTorch
  - Keras and SciKit
  - JAX/Flax
  - Code to start with:
    - https://github.com/pace-tl-ntu/Pytorch_Baseline_DLSCA (Single DNN)
    - https://github.com/pace-tl-ntu/Pytorch_Baseline_Ensemble_DLSCA (Ensemble DNNs)
    - https://github.com/AISyLab/AISY_Framework
    - https://github.com/ANSSI-FR/ASCAD
    - GPAM and sedpack tutorials:
      https://google.github.io/sedpack/tutorials/sca/dataset/
      https://google.github.io/sedpack/tutorials/sca/gpam/
- Hyperparameter Search Methods
  - SciKit based hyperparameter tuning
  - Keras Tuner

Additional References

- L. Masure, C. Dumas, and E. Prouff, "A comprehensive study of deep learning for side-channel analysis," IACR Trans. Cryptogr. Hardw. Embed. Syst., vol. 2020, no. 1, pp. 348–375, 2019, doi: 10.13154/tches.v2020.i1.348-375.
- G. Perin, Ł. Chmielewski, and S. Picek, "Strength in numbers: Improving generalization with ensembles in machine learning-based profiled side-channel analysis," IACR Trans. Cryptogr. Hardw. Embed. Syst., vol. 2020, no. 4, pp. 337–364, 2020, doi: 10.13154/tches.v2020.i4.337-364.
- L. Wouters, V. Arribas, B. Gierlichs, and B. Preneel, "Revisiting a methodology for efficient CNN architectures in profiling attacks," IACR Trans. Cryptogr. Hardw. Embed. Syst., vol. 2020, no. 3, pp. 147–168, 2020, doi: 10.13154/tches.v2020.i3.147-168.
- J. Rijsdijk, L. Wu, G. Perin, and S. Picek, "Reinforcement learning for hyperparameter tuning in deep learning-based side-channel analysis," IACR Trans. Cryptogr. Hardw. Embed. Syst., vol. 2021, no. 3, pp. 677–707, 2021, doi:10.46586/tches.v2021.i3.677-707
- L. Wu, G. Perin, and S. Picek, "I choose you: Automated hyperparameter tuning for deep learning-based side-channel analysis," IEEE Trans. Emerg. Top. Comput., early access, 2022, doi: 10.1109/TETC.2022.3218372.
- R. Y. Acharya, F. Ganji, and D. Forte, "Information theory-based evolution of neural networks for side-channel analysis," IACR Trans. Cryptogr. Hardw. Embed. Syst., vol. 2023, no. 1, pp. 401–437, 2023, doi: 10.46586/tches.v2023.i1.401-437

III. AI for Implementation Attacks

The working group discussed a structured representation of the domain of AI for Implementation Attacks with 7 major topics.

- Standard Architectures for AI based Implementation Security Testing
  - CUDA
- Countermeasure Design
  - Randomization and Shuffling
  - Generating Secure Implementations
  - Hardware Trojan Detection
  - EMFI and voltage glitch detection
  - Pre-silicon vs Post-silicon
    - Presilicon - need specific training, need specific testing (develop countermeasures iteratively)
- Side Channel Analysis
  - Portability
  - Profiled vs Non-profiled Analysis
  - Collision Neural Networks
- Fault Injection
  - AI-based approaches for Fault Detection
  - Fault Injection Parameter Search
  - Formal/Symbolic AI
- Advanced AI Techniques

    - Reinforcement Learning
    - Hyperparameter Optimization
    - Graph Neural Networks
    - Genetic Algorithms
    - Bayesian Techniques
    - Diffusion Models
    - Explainability: Explainable Artificial Intelligence (XAI) in hardware security enhances the trust and accountability of AI systems. By applying XAI concepts to hardware security, development of secure and transparent AI systems can be achieved.
    - Attribution
    - Occlusion Methods
    - Uncertainty Estimation
    - Interpretable Neural Networks
- New Directions and Needs
    - Tiny Models for the edge
    - Zero-day evaluation
    - Explainability Toolkit
    - Pretrained libraries
    - Model Zoo
    - Pretrained Datasets (Transfer Learning):
        - Standard Dataset
        - Higher Order Masking
        - Portability
    - Huggingface for dataset storage
    - Kaggle for dataset storage (around 200GB)

AI algorithms can also be used for the evaluation of implemented countermeasures against implementation attacks along with Test Vector Test Vector Leakage Assessment (TVLA).

References

- S. Karayalcin, M. Krcek, and S. Picek, "A practical tutorial on deep learning-based side-channel analysis," Cryptology ePrint Arch., Paper 2025/471, 2025. [Online]. Available: https://eprint.iacr.org/2025/471
- D. Koblah, R. Acharya, D. Capecci, O. Dizon-Paradis, S. Tajik, F. Ganji, D. Woodard, and D. Forte, “A survey and perspective on artificial intelligence for security-aware electronic design automation,” ACM Trans. Des. Autom. Electron. Syst., vol. 28, no. 2, pp. 1–57, 2023, doi: 10.1145/3563391.
- T. Moos, F. Wegener, and A. Moradi, "DL-LA: Deep learning leakage assessment: A modern roadmap for SCA evaluations," Cryptology ePrint Arch., Paper 2019/505, 2019. [Online]. Available: https://eprint.iacr.org/2019/505

- A. Gambra, U. Rioja, D. Chatterjee, I. Armendariz, and L. Batina: “Machine Learning Fault Injection Detection in Clock Signals: An Analysis of Frequency Impact", IEEE ISVLSI 2025, July 5-9, Kalamata, Greece. Available: https://eprint.iacr.org/2024/1939
- S. Nouraniboosjin and F. Ganji, "Uncertainty estimation in neural network-enabled side-channel analysis and links to explainability," Cryptology ePrint Arch., Paper 2025/688, 2025. [Online]. Available: https://eprint.iacr.org/2025/688
- S. Picek, G. Perin, L. Mariot, L. Wu, and L. Batina, "SoK: Deep learning-based physical side-channel analysis," ACM Comput. Surv., vol. 55, no. 11, Art. no. 227, pp. 1–35, 2023, doi: 10.1145/3569577.
- M. C. Tol and B. Sunar, "Zeroleak: Using LLMs for scalable and cost-effective side-channel patching," arXiv preprint, arXiv:2308.13062, 2023. [Online]. Available: https://arxiv.org/abs/2308.13062
- Federal Office for Information Security (BSI). (2024). *Guidelines for evaluating machine-learning based side-channel attack resistance – Part of AIS 46 (Version 1)*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_46_AI_guide.pdf
- D. Van der Valk, S. Picek, and S. Bhasin, “Kilroy was here: The first step towards explainability of neural networks in profiled side-channel analysis,” in *Proc. 11th Int. Workshop Constructive Side-Channel Analysis and Secure Design (COSADE)*, Lugano, Switzerland, Apr. 2020, Revised Selected Papers, vol. 12612, Springer, 2021, pp. 175–199. doi: 10.1007/978-3-030-68773-1_9
- L. Wu, Y.-S. Won, D. Jap, G. Perin, S. Bhasin, and S. Picek, “Explain some noise: Ablation analysis for deep learning-based physical side-channel analysis,” *IACR Cryptology ePrint Archive*, vol. 2021, no. 717, pp. 1–24, 2021. [Online]. Available: https://eprint.iacr.org/2021/717

IV. Implementation Attacks on AI

The working group also discussed implementation attacks on AI by starting with the attacker model. Because of the diversity of use cases and attack models, when compared to traditional cryptographic assets, the working group selected two specific attacker models for further discussion. The first attacker model applies to the embedded case, where the attacker has access to the physical implementation of the neural network. The second attacker model applies to the cloud setting, where the attacker needs to mount attacks indirectly by manipulating/observing computing resources that are physically close to the neural network implementation.

Scenario 1: Attack on an embedded platform. Consider a sensor that is used to drive an authentication processor, such as a camera sensor for face recognition. The sensor captures potentially complex and noisy data that requires processing in a neural network. The final labeling (neural network output) authenticates the user and is susceptible to impersonation or manipulation. The working group considers side-channel analysis on the neural network processing as a starting point to reveal the authentication token.
Reverse engineering of neural networks also becomes possible through side-channel analysis:

- Retrieval of the number of neurons and layers through simple power/EM analysis
- Retrieval of trained weight value through correlation power analysis

Scenario 2: Attack on a cloud platform. Consider a neural network that operates in a shared processing architecture, and an attacker that aims to manipulate (not just reveal) the token processed by the neural network. Because of the network's complexity, redundancy techniques such as commonly applied against fault injection, are less suitable. Instead the defender aims to reveal the injected fault as soon and as reliable as possible. And conversely, the attacker aims to identify the location for the most effective fault injection.

The attacker model is traditionally defined in terms of the level of access.

- Physical access
    - Side channel measurements
    - Fault injection
    - Preventing firmware updates or hijacking GPU hardware during firmware update
- Memory attacker
    - Neural networks have complex memory hierarchies, leading to multiple attacker models: intra- and inter-GPU, intra- and inter-VM. Confidential computing implements logical isolation in multi-processor context and is a target in each of these cases.
- Input/output attacker
    - Manipulation of data, such as adversarial training, prompt injection, LLM inversion, are potential vulnerabilities with their own defenses. However, the I/O attacker is considered out of scope for the implementation attacker.
- Disclosure of new attacks and attacker models is considered to be a challenge.

References

- S. Tajik and F. Ganji, "Artificial neural networks and fault injection attacks," in Security and Artificial Intelligence: A Crossdisciplinary Approach, Cham, Switzerland: Springer International Publishing, 2022, pp. 72–84. [Online]. Available: https://arxiv.org/pdf/2008.07072.
- M. C. Tol and B. Sunar, "Zeroleak: Using LLMs for scalable and cost-effective side-channel patching," arXiv preprint, arXiv:2308.13062, 2023. [Online]. Available: https://arxiv.org/pdf/2505.00817
- P. Horváth, D. Lauret, Z. Liu, and L. Batina, "SoK: Neural network extraction through physical side channels," in Proc. USENIX Security Symp., 2024. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/horvath
- P. Horváth, L. Chmielewski, L. Weissbart, L. Batina, Y. Yarom: BarraCUDA: GPUs do Leak DNN Weights. to appear at USENIX Security Symposium 2025. Available: https://arxiv.org/abs/2312.07783

- A. Adiletta and B. Sunar, “Spill the beans: Exploiting CPU cache side-channels to leak tokens from large language models,” arXiv preprint, arXiv:2505.00817, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.00817.
- K. Lee, A. Alshahrani, W. Wang, B. Malekian, and J. Szefer, “Secure machine learning hardware: Challenges and progress,” IEEE Circuits Syst. Mag., vol. 25, no. 1, pp. 8–34, 2025. [Online]. Available: https://doi.org/10.1109/MCAS.2024.3509376
- A. Adiletta, Z. Weissman, F. Khojasteh Dana, B. Sunar, and S. Tajik, “Rubber Mallet: A study of high frequency localized bit flips and their impact on security,” arXiv preprint, arXiv:2505.01518, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.01518
- J. Breier, D. Jap, X. Hou, S. Bhasin, and Y. Liu, “SNIFF: Reverse engineering of neural networks with fault attacks,” IEEE Transactions on Reliability, vol. 71, no. 4, pp. 1527–1539, Dec. 2022. doi: 10.1109/TR.2021.3102840
- L. Batina, S. Bhasin, D. Jap, and S. Picek, “CSI NN: Reverse engineering of neural network architectures through electromagnetic side channel,” in Proc. 28th USENIX Security Symposium (USENIX Security 2019), Santa Clara, CA, USA, Aug. 2019, pp. 515–532. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/batina

V. Datasets

The working group observes that there is no common methodology or practice to systematically share datasets in the context of AI for implementation security testing. This problem was also observed earlier during the OPTIMIST discussions on Standard File Formats for side-channel traces. ASCAD is a good starting point; ScapeGOAT offers an ability to store metadata and to organize trace sets hierarchically. For long-term storage, the working group concluded that zenodo.org, figshare.com, and huggingface.co are possible containers. Zenodo offers a DOI for the data; huggingface offers free storage as long as the data is publicly shared. The following table is a partial list of known public datasets with variable and fixed keys.

Standard Datasets for Side-channel

ID	**SW/HW (Seq/Par)**	**Prot/ Unprot**	**Features**
ASCAD	SW	Both	Alignment, 8bit, Fix/Var Key AES
ASCADv2	SW	Prot	32bit, Fix/Var Key AES
AESRD	SW	Prot	8bit, Random Delay AES
AESHD	HW	Unprot	FPGA, AES
CS3, CS5	HW	Unprot	FPGA, misaligned tr, PRESENT
ECC	SW	Prot	32 bit, Curve25519 EdDSA

WolfSSL	SW	Prot	32 bit, Curve25519 EdDSA
CHES CTF			Partially on aisylab
GPAM ECC	HW	Prot	ECC scalar multiplication, large
DPA Contest V2	HW	Both	AES-128 on SASEBO GII https://cloud.telecom-paris.fr/s/N5qgyMdxEcqipN2 https://cloud.telecom-paris.fr/s/iScPMi78Jg8jere
DPA Contest V4	SW	Both	AES-256 on ATMega-163 https://cloud.telecom-paris.fr/s/PP79GTSj9mmg4xL
DPA Contest V4.2	SW	Both	AES-128 on ATMega-163 https://cloud.telecom-paris.fr/s/JM2iaRZfwrNKtSp
AES_HD_MM	HW		*Missing- AES 128 on SASEBO GII*
Ed25519	SW	Both	EdDSA on STM32F4
Curve25519	SW	Both	EdDSA on STM32F4
Kyber	SW	Unprot	https://eprint.iacr.org/2025/811
Ascon	SW/HW	Unprot	https://zenodo.org/records/10229484
SMAesH	HW	Prot	AES block cipher with masking as a countermeasure
scaaml	NXP K82F	Both	ECC on NXP K82F https://github.com/google/scaaml/tree/main/papers/datasets/ECC/GPAM

Need for other dataset:
Side-channel dataset are broadly available for AES, ECC, EdDSA implementations, so the creation of side-channel dataset (SW/HW) for other ciphers, standardized Post Quantum Cryptographic algorithms (protected and unprotected) would be useful.

References

- S. Picek, G. Perin, L. Mariot, L. Wu, and L. Batina, “SoK: Deep learning-based physical side-channel analysis,” ACM Comput. Surv., vol. 55, no. 11, Art. no. 227, pp. 1–35, 2023. [Online]. Available: https://doi.org/10.1145/3569577 (Table on Page 13)

- D. Mehta, T. Marcantino, M. Hashemi, S. Karkache, D. Shanmugam, P. Schaumont, and F. Ganji, “SCAPEgoat: Side-channel Analysis Library,” in Proceedings of the IEEE 43rd VLSI Test Symposium (VTS), 2025, pp. 1–7, doi: 10.1109/VTS65138.2025.11022809.
- Side-channel Analysis section on Papers With Code: https://paperswithcode.com/task/side-channel-analysis
- E. Prouff, R. Strullu, R. Benadjila, E. Cagli, and C. Dumas, “Study of deep learning techniques for side-channel analysis and introduction to ASCAD database,” *J. Cryptographic Engineering*, vol. 10, no. 2, pp. 163–188, 2019, doi: 10.1007/s13389-019-00220-8.