# Side-Channel and Fault Attack Testing in the age of AI

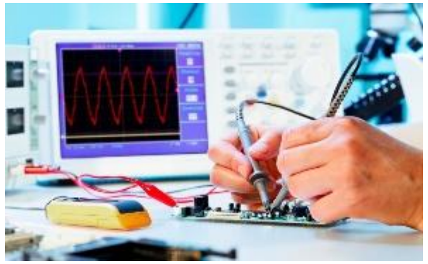

Debdeep Mukhopadhyay
Institute Chair Professor
(FIEEE, FNA, FASc, FNAE, FAAIA)

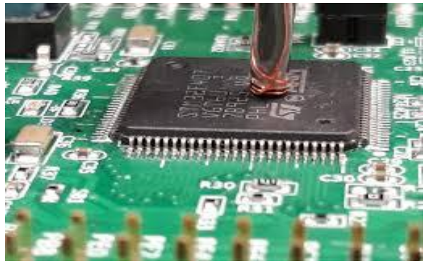
Secured Embedded Architecture Laboratory (SEAL)

Department of Computer Science and Engineering
IIT Kharagpur
debdeep@cse.iitkgp.ac.in
debdeep.mukhopadhyay@gmail.com

# **Artificial Intelligence:** Boon for Security Evaluations of Cryptosystems

Targets insecure real-life implementation of provably secure cryptographic encryption








**AI for Attacks**

- Can be used to attack cryptographic implementation using side-channel information
  - More *efficient* than standard statistical techniques
  - Can even *break efficient side-channel countermeasure* implementation like masking
- Can be used to evaluate fault attack countermeasures
  - *Aids to mount efficient fault attacks* on cryptographic implementation

Image Source: Google Images

# This Talk

(Our experience with) **Challenges and Future of Machine Learning in SCA**

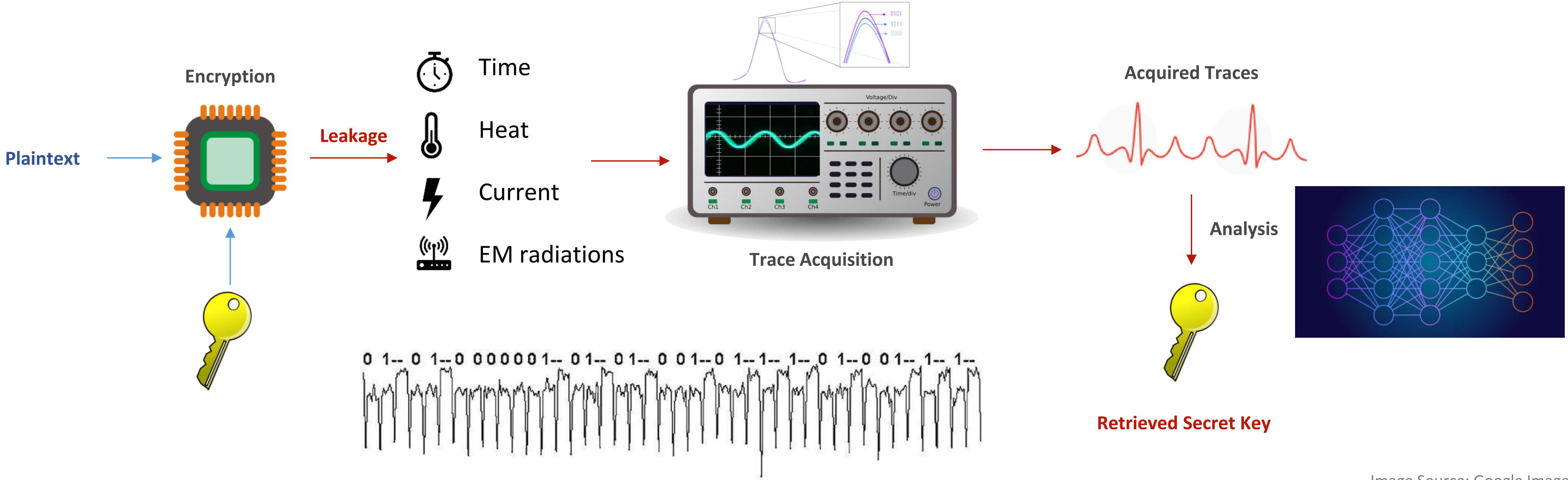


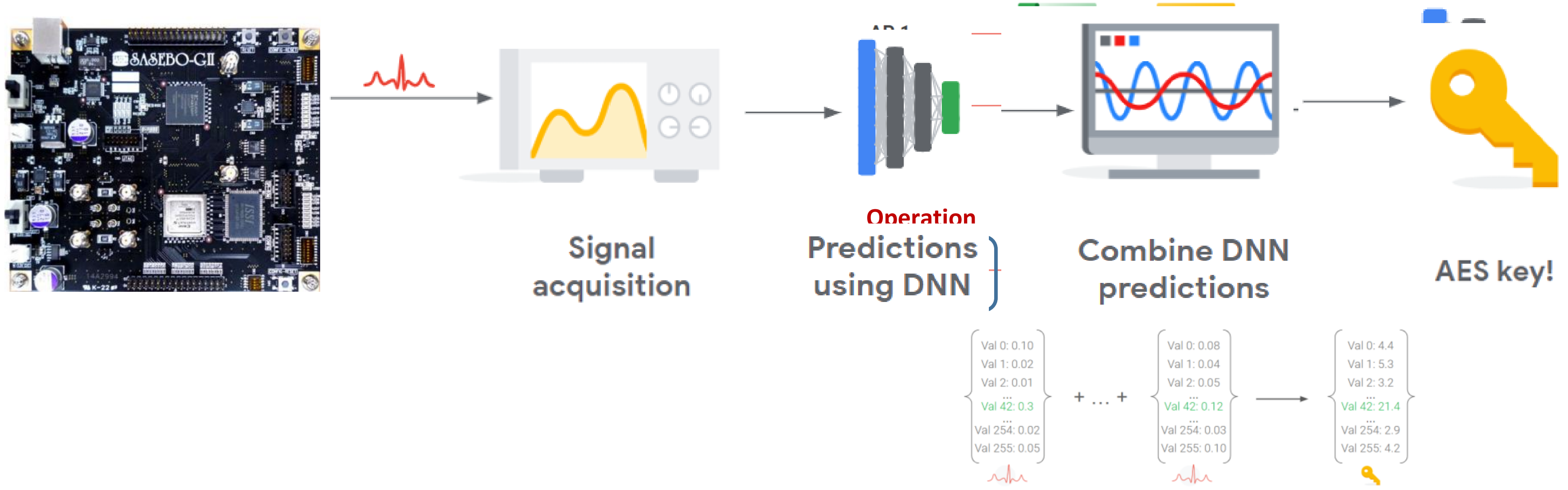

Image Source: Google Images

# **Profiled Deep Learning** based Attack

## Advantage of Deep Learning Approach

- Template attacks (profiled attacks) are based on the Gaussian Assumption of the leakage distribution
  - Deep Learning-based approach relaxes the Gaussian Assumption
- The acquired traces do not require any types of pre-processing (like trace synchronization, denoising, feature selection, feature extraction, etc.)
- The deep learning model can combine multiple leakage points
  - Works efficiently against masking-based countermeasures than classical statistical methods

Unknown Key




Profiling Phase

Attack Phase

H. Maghrebi et al., "Breaking cryptographic implementations using deep learning techniques." SPACE 2016

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Lack of documentation for AI specific pitfalls in implementation security context**

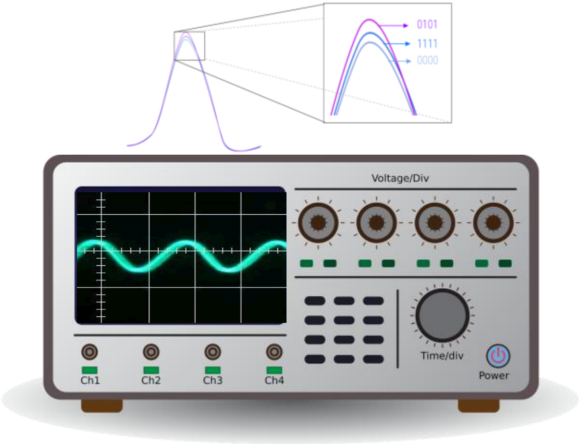
    - How to pre-process data (like side-channel traces) to render them training friendly ?
    - How to avoid overfitting ?
    - Which architectures to use for what use-cases? **Is deep = better?**
    - How to ensure quality of training data? Issues like class imbalance, requisite amount of data.
    - Is the inference result interpretation reliable (like high accuracy, but on unbalanced data) ?
    - Statistical tests (like CPA) are inherently portable. But what about ML models?

Not all implementation security engineers may know ML internals/details to take decisional calls on these issues

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Lack of documentation for AI specific pitfalls in implementation security context**
  - ***Example***: [1]



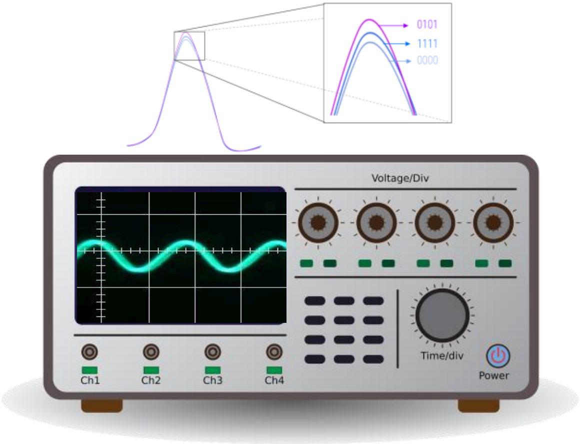

**Long traces in SCA**
Few informative sample points in the trace


[1] Hajra, Suvadeep, et al. "On the instability of softmax attention-based deep learning models in side-channel analysis." Ieee transactions on information forensics and security 19 (2023): 514-528.

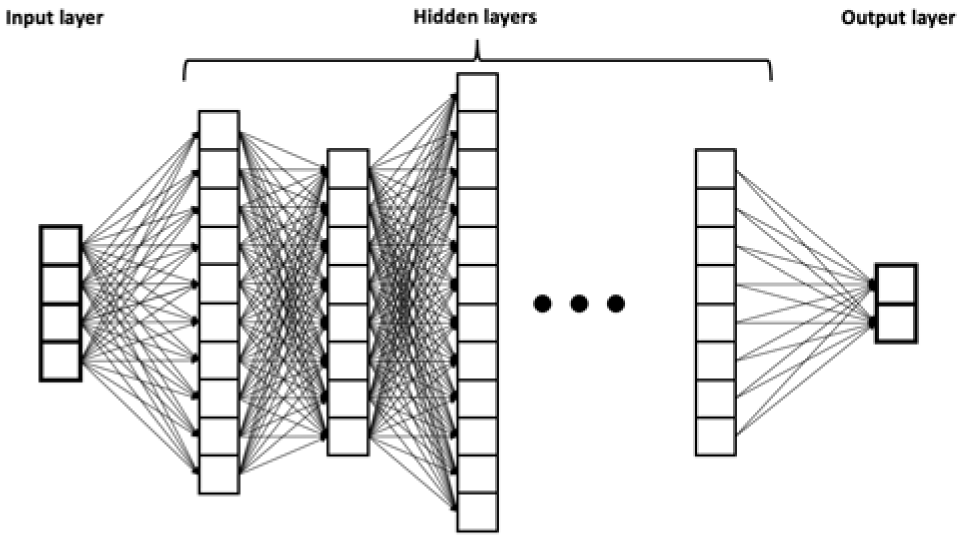
# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Lack of documentation for AI specific pitfalls in implementation security context**
    - ***Example***: [1]




**Long traces in SCA**
Few informative sample points in the trace



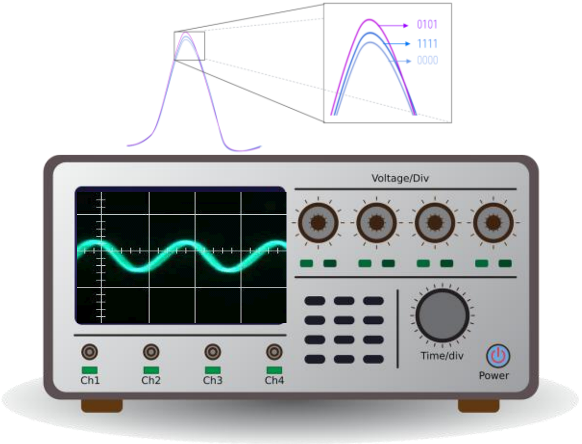

**Attack using Convolutional Neural Network**


[1] Hajra, Suvadeep, et al. "On the instability of softmax attention-based deep learning models in side-channel analysis." Ieee transactions on information forensics and security 19 (2023): 514-528.

# AI for Implementation Security : The Way Forward

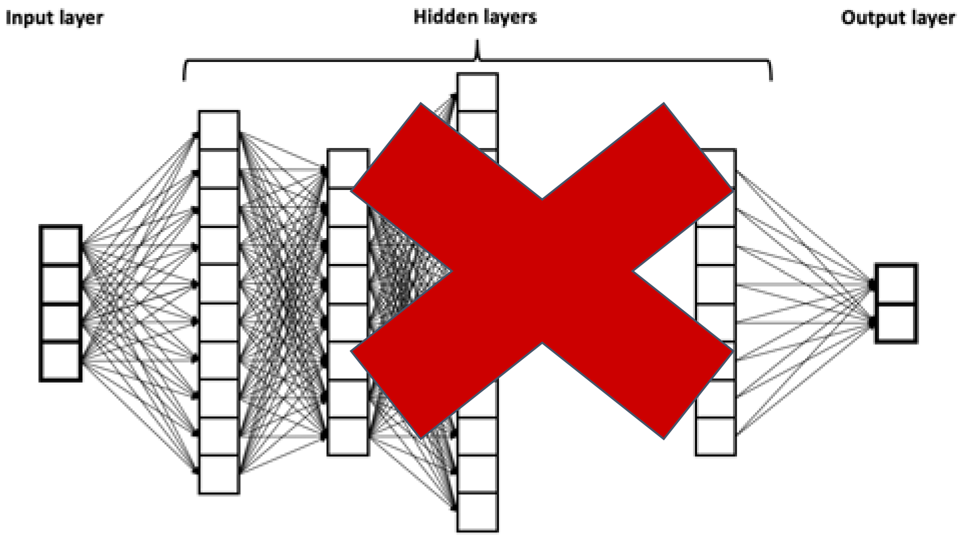
Pertinent questions facing the implementation security community for AI adoption:

- **Lack of documentation for AI specific pitfalls in implementation security context**
    - ***Example***: [1]




**Long traces in SCA**

Few informative sample points in the trace




**Attack using Convolutional Neural Network**

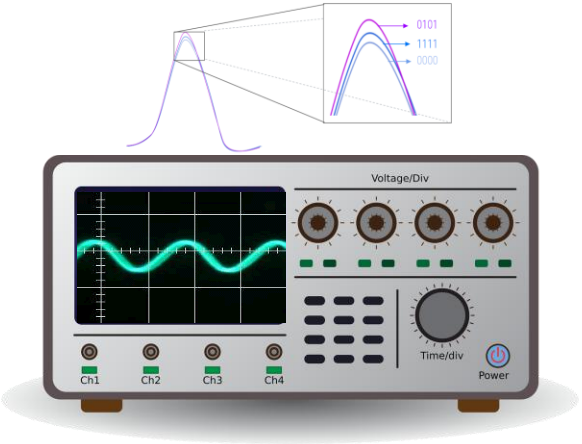
Pitfall!!! High gradient for softmax; Highly non-smooth loss surface; unstable learning

[1] Hajra, Suvadeep, et al. "On the instability of softmax attention-based deep learning models in side-channel analysis." Ieee transactions on information forensics and security 19 (2023): 514-528.

# AI for Implementation Security : The Way Forward

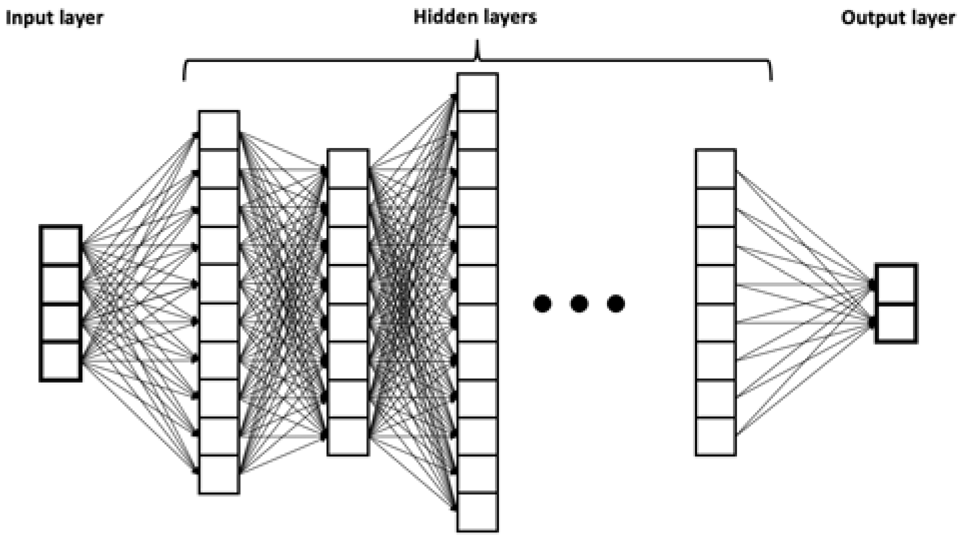
Pertinent questions facing the implementation security community for AI adoption:

- **Lack of documentation for AI specific pitfalls in implementation security context**
  - ***Example***: [1]




**Long traces in SCA**

Few informative sample points in the trace




**Attack using Convolutional Neural Network**

Fix: Machine Learning specific fixes (multi-head softmax) for accurate SCA using CNNs


[1] Hajra, Suvadeep, et al. "On the instability of softmax attention-based deep learning models in side-channel analysis." Ieee transactions on information forensics and security 19 (2023): 514-528.

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Lack of documentation for AI specific pitfalls in implementation security context**
  - How to pre-process data (like side-channel traces) to render them training friendly ?
  - How to avoid overfitting ?
  - Which architectures to use for what use-cases? **Is deep = better?**
  - How to ensure quality of training data? Issues like class imbalance, requisite amount of data.
  - Is the inference result interpretation reliable (like high accuracy, but on unbalanced data) ?
  - Statistical tests (like CPA) are inherently portable. But what about ML models?

Not all implementation security engineers may know ML internals/details to take decisional calls on these issues

**Way Forward:**

- Document problems with vanilla ML approaches (i.e. what *does not* work) for dissemination.
- Foster community dialogue at the intersection of SCA and Machine Learning
- Develop whitepapers and RFC documents to answer some of the questions raised here

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

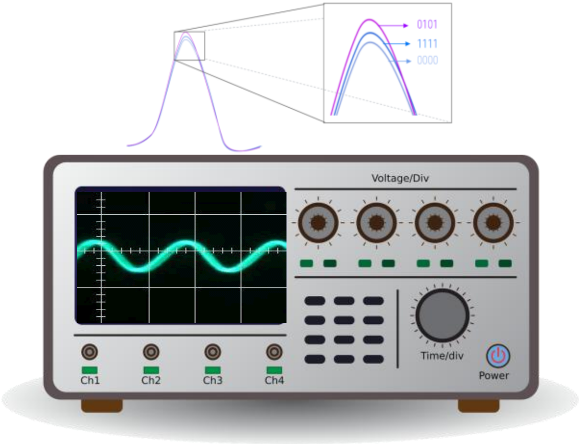
- **Need for interdisciplinary know-how to *improve* implementation security testing through AI**
    - Are there problems in SCA whose counterparts exist in machine learning?
    - Does AI make certain SCA specific problems *easier* to solve?

Exploring the intersection of AI and SCA requires intricate intertwining of two completely different domains.

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Need for interdisciplinary know-how to *improve* implementation security testing through AI**
    - **Example**: [1]




**Long traces in SCA**
Few informative sample points in the trace

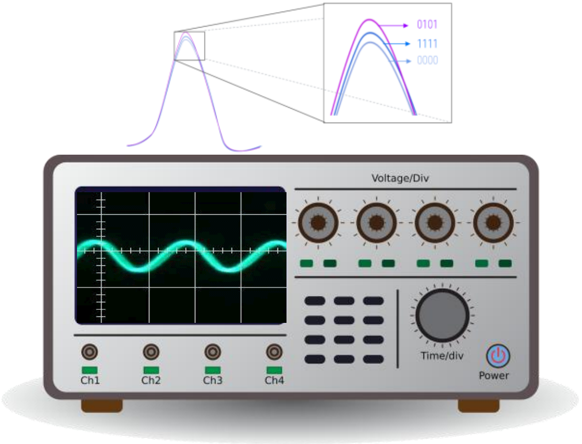
→ **Mechanisms in machine learning capable of capturing long, temporal input semantics ?**

[1] Hajra, Suvadeep, Siddhartha Chowdhury, and Debdeep Mukhopadhyay. "Estranet: An efficient shift-invariant transformer network for side-channel analysis." IACR Transactions on Cryptographic Hardware and Embedded Systems 2024.1 (2024): 336-374.

# AI for Implementation Security : The Way Forward

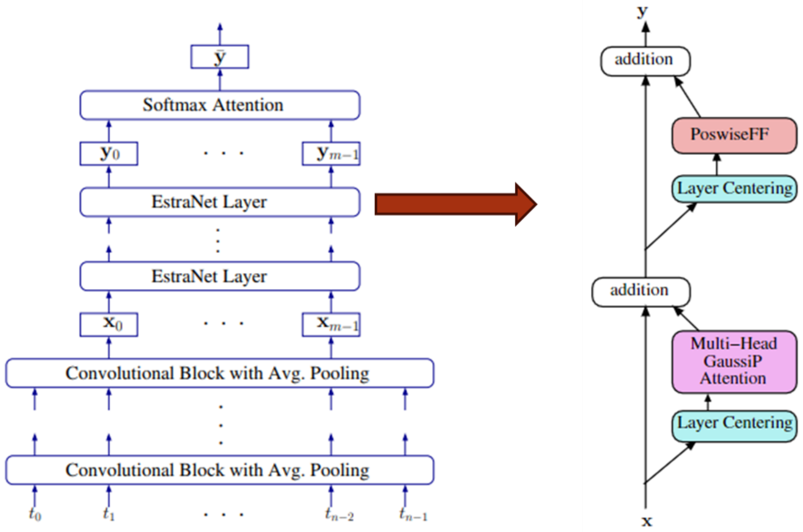
Pertinent questions facing the implementation security community for AI adoption:

- **Need for interdisciplinary know-how to *improve* implementation security testing through AI**
  - **Example**: [1]




**Long** **traces in SCA**

Few informative sample points in the trace



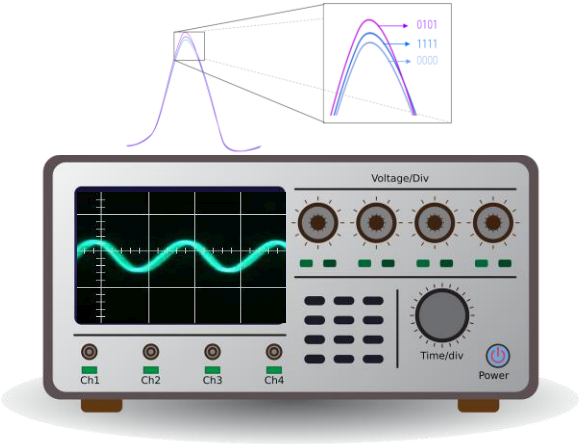

EstraNet : SCA specific architecture based on advances in ML specific techniques - new attention mechanisms (relative positional encoding) and layer normalization


[1] Hajra, Suvadeep, Siddhartha Chowdhury, and Debdeep Mukhopadhyay. "Estranet: An efficient shift-invariant transformer network for side-channel analysis." IACR Transactions on Cryptographic Hardware and Embedded Systems 2024.1 (2024): 336-374.

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Need for interdisciplinary know-how to *improve* implementation security testing through AI**
  - **Example**: [1], [2]




**Very Long traces in SCA**

EstraNet does not work beyond 40K features. What if the target implementation has > 40K features in side-channel trace?

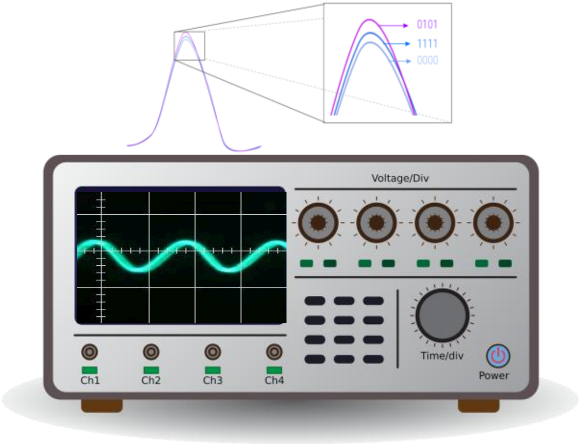
**Mechanisms in machine learning for operating a "collection" of models in specific topologies?**

[2] Hajra, Suvadeep, Soumi Chatterjee, and Debdeep Mukhopadhyay. "UNIDLE: A Unified Framework for Deep Learning-based Side-channel Analysis." Cryptology ePrint Archive (2024).

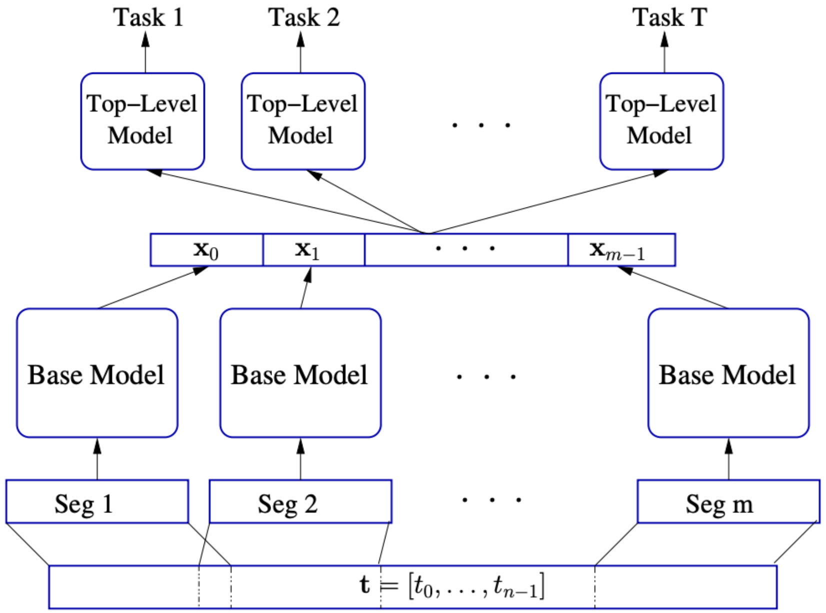
# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Need for interdisciplinary know-how to *improve* implementation security testing through AI**
  - **Example**: [1], [2]




**Very Long traces in SCA**



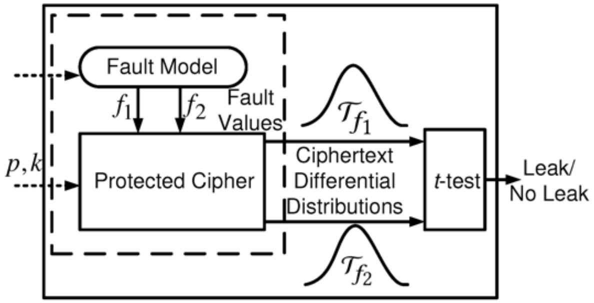

Exploits properties of interactions between ML models to construct a hierarchical topology of models (~100K features in SCA traces)


[2] Hajra, Suvadeep, Soumi Chatterjee, and Debdeep Mukhopadhyay. "UNIDLE: A Unified Framework for Deep Learning-based Side-channel Analysis." Cryptology ePrint Archive (2024).

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Need for interdisciplinary know-how to *improve* implementation security testing through AI**
  - **Example**: [1], [2], [3]




Automated Fault Analysis

Higher-order **t-test can only capture different statistical moments**, which has already been shown to be sub-optimal in the context of SCA leakages, even resulting in **false negatives**

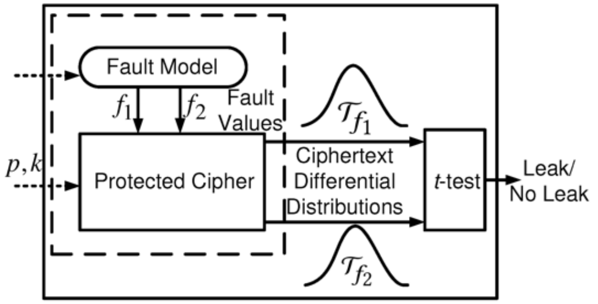
**Can machine learning help build an automated test for multivariate inputs?**


[3] Saha, Sayandeep, et al. "Learn from your faults: leakage assessment in fault attacks using deep learning." Journal of Cryptology 36.3 (2023): 19.

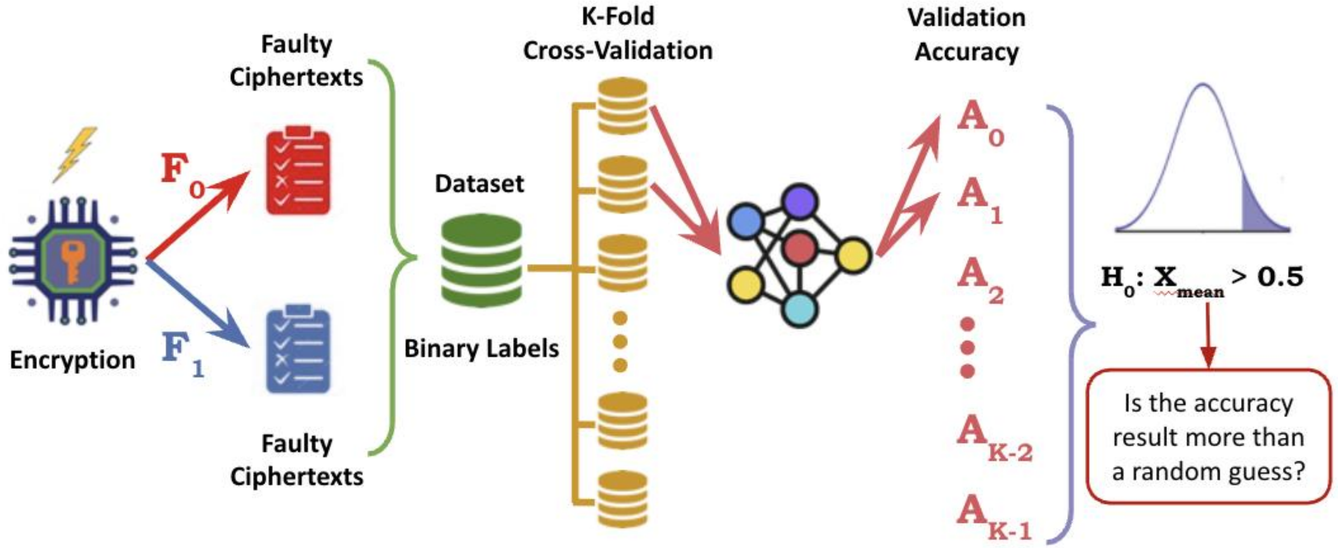
# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Need for interdisciplinary know-how to *improve* implementation security testing through AI**
    - **Example**: [1], [2], [3]




Automated Fault Analysis



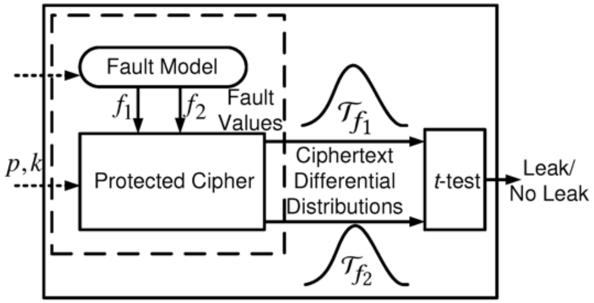

DL-FALAT : **Machine learning can learn highly multivariate functions and complex inter-relations between inputs**, implying an ideal candidate fault analysis automation.


[3] Saha, Sayandeep, et al. "Learn from your faults: leakage assessment in fault attacks using deep learning." Journal of Cryptology 36.3 (2023): 19.

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Need for interdisciplinary know-how to *improve* implementation security testing through AI**
  - **Example**: [1], [2], [3], [4]




Automated Fault Analysis with unknown fault models

Even most ML based fault detection methods require functioning over *known* fault models. What about *unknown* models?

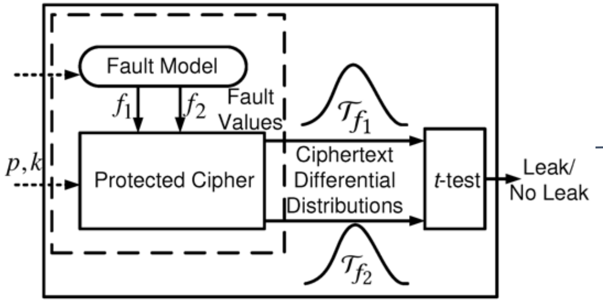
**Can learning agents be added to existing Machine Learning based fault automation?**


[4] Guo, Hao, et al. "ExploreFault: Identifying exploitable fault models in block ciphers with reinforcement learning." 2023 60th ACM/IEEE Design Automation Conference (DAC). IEEE, 2023.

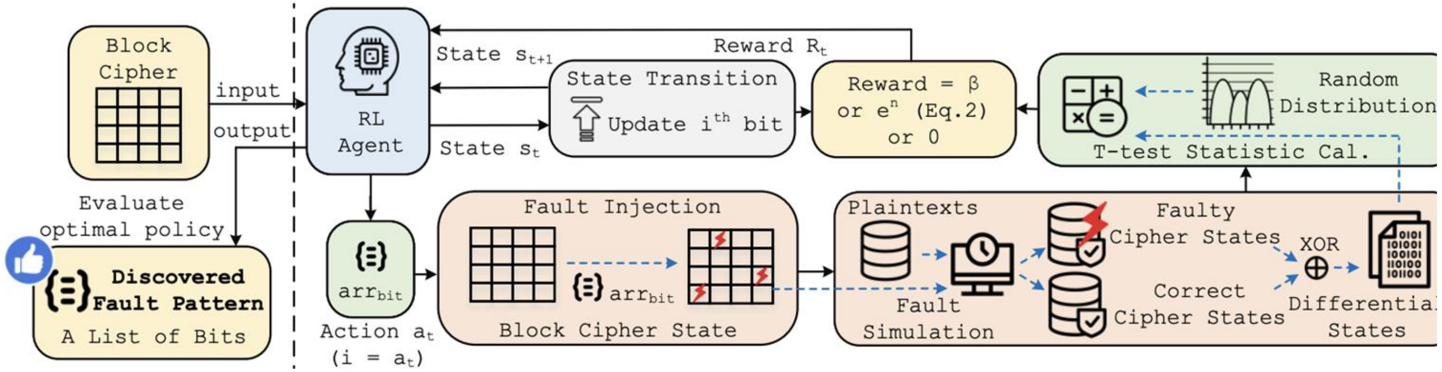
# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Need for interdisciplinary know-how to *improve* implementation security testing through AI**
  - **Example**: [1], [2], [3], [4]




Automated Fault Analysis with unknown fault models




**Reinforcement Learning (RL) based Machine Learning pipeline** with active feedback loop for exploring newer, previously undiscovered fault models

[4] Guo, Hao, et al. "ExploreFault: Identifying exploitable fault models in block ciphers with reinforcement learning." 2023 60th ACM/IEEE Design Automation Conference (DAC). IEEE, 2023.

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Need for interdisciplinary know-how to *improve* implementation security testing through AI**
  - **Example**: [1], [2], [3], [4], [5]

Testing quality of entropy sources used in cryptosystem-under-test

Statistical Analyses is difficult (even NIST rolled back SP-800-22 for use for testing PRNGs in production level cryptosystems) ?

**Can properties of machine learning help in designing measures of entropy quality?**

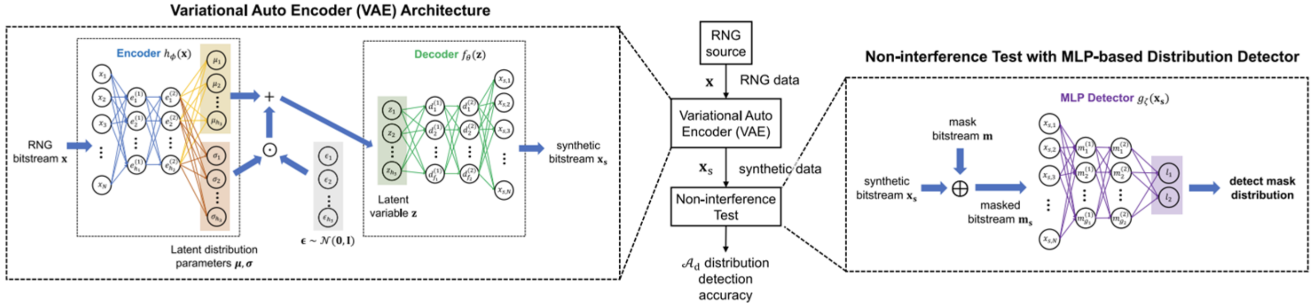
[5] Pratihar, Kuheli, Rajat Subhra Chakraborty, and Debdeep Mukhopadhyay. "Latent RAGE: Randomness Assessment Using Generative Entropy Models." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 43.11 (2024): 3503-3514.

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Need for interdisciplinary know-how to *improve* implementation security testing through AI**
  - **Example**: [1], [2], [3], [4], [5]

Testing quality of entropy sources used in cryptosystem-under-test




Use ML specific hypothesis - patterns are likely to occur on low-dimensional manifolds - to detect weak entropy sources by studying their projections to lower dimensional manifolds using Variational Auto-Encoders.


[4] Pratihar, Kuheli, Rajat Subhra Chakraborty, and Debdeep Mukhopadhyay. "Latent RAGE: Randomness Assessment Using Generative Entropy Models." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 43.11 (2024): 3503-3514.

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Need for interdisciplinary know-how to *improve* implementation security testing through AI**
    - Are there problems in SCA whose counterparts exist in machine learning?
    - Does AI make certain SCA specific problems *easier* to solve?

Exploring the intersection of AI and SCA requires intricate intertwining of two completely different domains.

**Way Forward**:

- Explore more and more interdisciplinary solutions to problems in SCA
- Induct interdisciplinary experts, foster wider collaboration, and keep trying to find intersections in problems

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Consensus on how we build upon each other’s research**
    - Statistical tests are algorithms; we publish the same, and anyone can use it.
    - Trained ML models + collected datasets are often considered as IPs. Not everyone open-sources them!
    - A **lack of consensus on how to share models/datasets will create isolated islands** in the implementation security community.

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Consensus on how we build upon each other's research**
  - Statistical tests are algorithms; we publish the same, and anyone can use it.
  - Trained ML models + collected datasets are often considered as IPs. Not everyone open-sources them!
  - A **lack of consensus on how to share models/datasets will create isolated islands** in the implementation security community.

- **Licensing**
  - What is the goodwill policy of licensing a model for testing *my* implementation, that had been partially trained on dataset curated by somebody else?
  - **Consensus on licensing of such models**, especially if they are transfer learned!

Agreeing upon suitable licensing of ML in SCA research is crucial for the field to boom!

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Consensus on how we build upon each other's research**
  - Statistical tests are algorithms; we publish the same, and anyone can use it.
  - Trained ML models + collected datasets are often considered as IPs. Not everyone open-sources them!
  - A **lack of consensus on how to share models/datasets will create isolated islands** in the implementation security community.

- **Licensing**
  - What is the goodwill policy of licensing a model for testing *my* implementation, that had been partially trained on dataset curated by somebody else?
  - **Consensus on licensing of such models**, especially if they are transfer learned!

Agreeing upon suitable licensing of ML in SCA research is crucial for the field to boom!

As a lab, we try to open-source both source code (model architecture, hyperparameters) and datasets for the wider community to build upon.

- https://github.com/felu-mittir/DL_FALAT
- https://github.com/suvadeep-iitb/EstraNet

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Explainability and Democratization!**

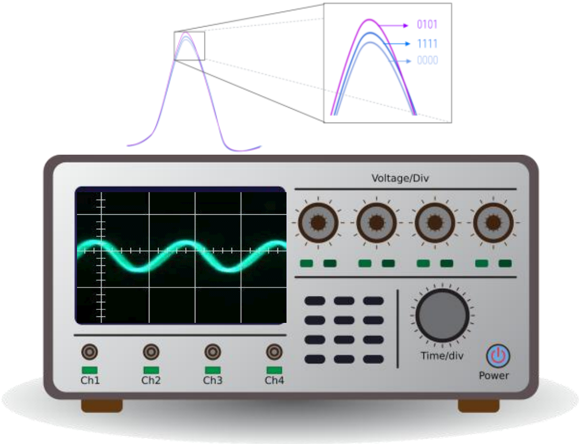
  - Sophisticated toolings => use-cases limited to a small community of experts!
  - **Enabling wider adoption** of implementation security testing is paramount.
  - Apart from automation, **explainability** and **democratization** of our research goes a long way.

Constant efforts are needed on our part to engage non-experts in ML to adopt our workflows in their testing workflows.

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Explainability and Democratization!**
    - Explainability example : [1]




How does a security auditer explain even a successful audit by machine learning deployment?

Hard to know *why* a machine learning model makes the decision it does. And hard to capture SCA specific semantics in such contexts. Without a clearer understanding of how the model internally processes information, systematic improvements beyond trial-and-error become difficult

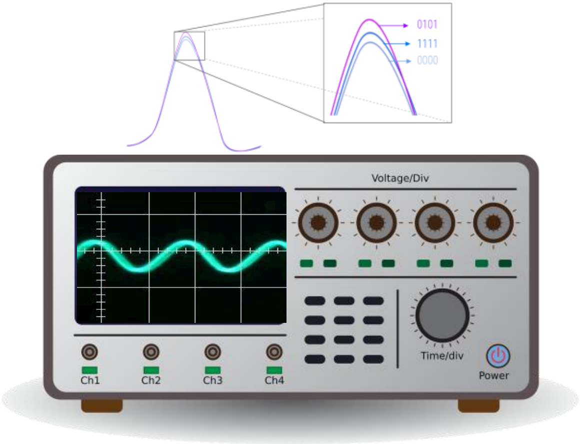
**Can a combination of (1) machine learning, (2) visualization techniques, and (3) statistics help with *explaining* SCA leakage in ML models?**

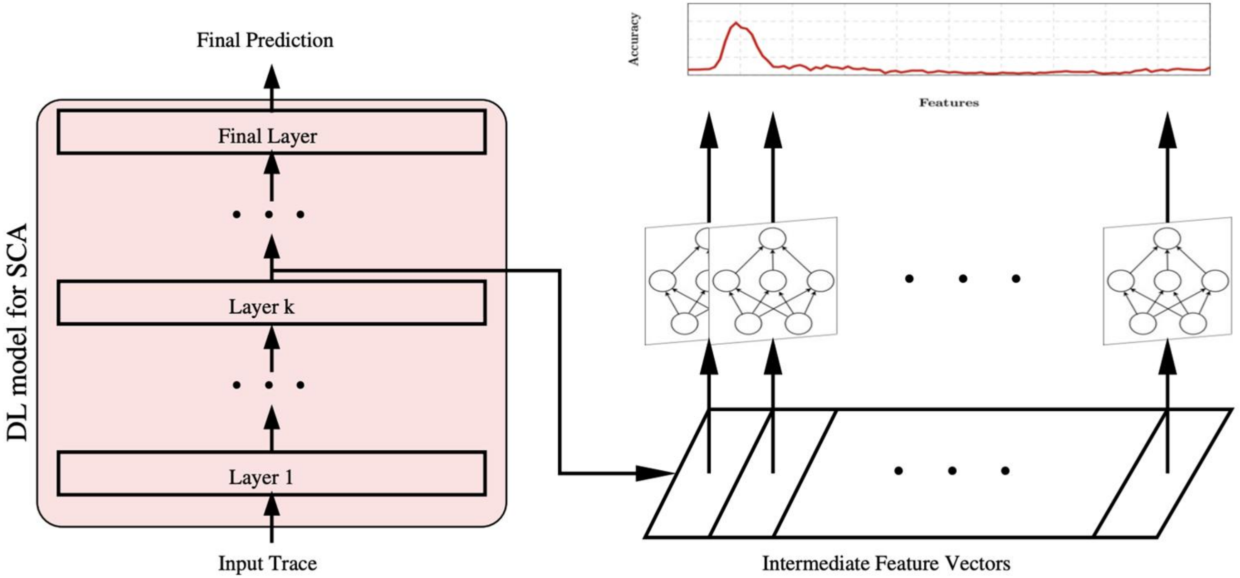
[1] Hajra, Suvadeep, and Debdeep Mukhopadhyay. "Black Box to Blueprint: Visualizing Leakage Propagation in Deep Learning Models for SCA." Cryptology ePrint Archive (2025).

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Explainability and Democratization!**
    - Explainability example : [1]




How does a security auditer explain even a successful audit by machine learning deployment?

Generalizable method to visualize flow of leakage while machine learning model processes SCA traces

[1] Hajra, Suvadeep, and Debdeep Mukhopadhyay. "Black Box to Blueprint: Visualizing Leakage Propagation in Deep Learning Models for SCA." Cryptology ePrint Archive (2025).

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Explainability and Democratization!**
    - Explainability example : [1]
    - Democratization example: [2]


Given a framework for side-channel leakage assessment, how do we get a non-expert to use it?

**Can we use AI agents to automate and abstract out details of the framework?**

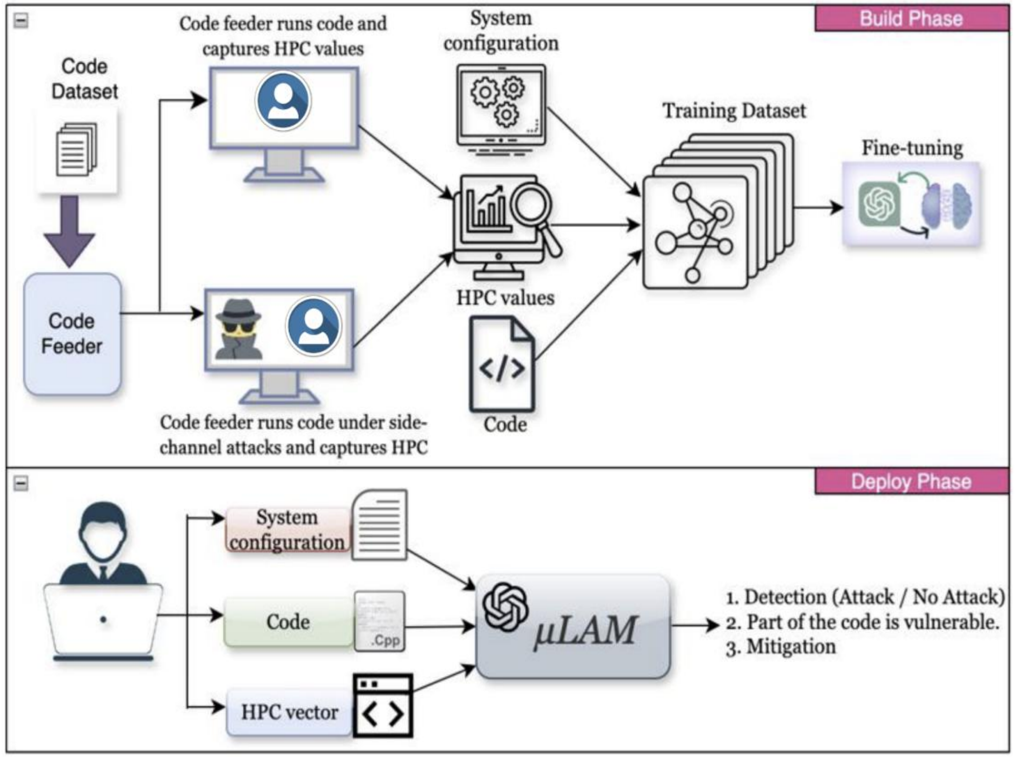
[2] Mandal, Upasana, et al. "µLAM: A LLM-Powered Assistant for Real-Time Micro-architectural Attack Detection and Mitigation." Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design. 2024.

# AI for Implementation Security : The Way Forward

Pertinent questions facing the implementation security community for AI adoption:

- **Explainability and Democratization!**
  - Explainability example : [1]
  - Democratization example: [2]


Given a framework for side-channel leakage assessment, how do we get a non-expert to use it?




Design of a LLM based framework to abstract out details of internal side-channel leakage detection, to be used naturally by non-experts by simple English

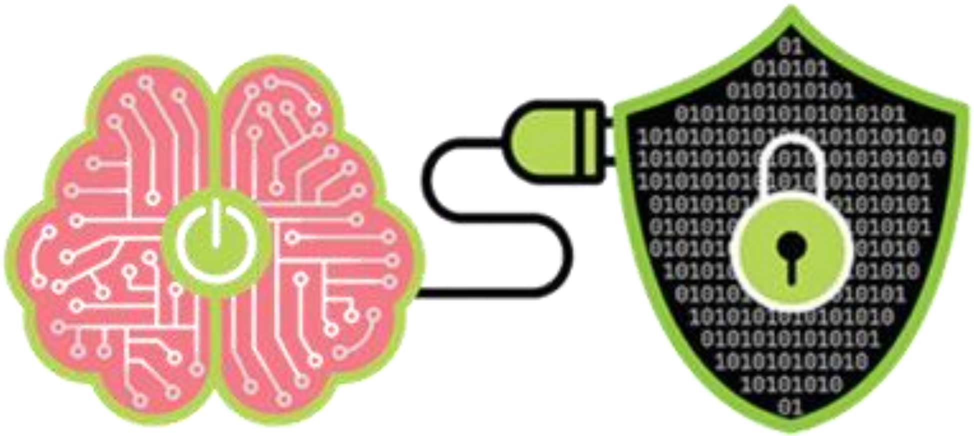
[2] Mandal, Upasana, et al. "µLAM: A LLM-Powered Assistant for Real-Time Micro-architectural Attack Detection and Mitigation." Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design. 2024.

# AI for Implementation Security : The Way Forward

(Summary) Pertinent questions facing the implementation security community for AI adoption:

- **Lack of documentation for AI specific pitfalls in implementation security context**
- **Need for interdisciplinary know-how to improve implementation security testing through AI**
- **Consensus on how we build upon each other’s research**
- **Explainability and Democratization**

A big shout out to OPTIMIST in making inroads on taking steps in improving the quality and reproducibility of implementation testing!



# *Thank You*

For any query please feel free to contact

**Debdeep Mukhopadhyay**: debdeep.mukhopadhyay@gmail.com