NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

ZHEJIANG UNIVERSITY

Reality Check on Side-Channels: Lessons learnt from breaking AES on ARM Cortex-A72 processor with Out-of-Order Execution

Harishma Boyapally[1,2], Dirmanto Jap[1,2], Qianmei Wu[3,4], Fan Zhang[4], Shivam Bhasin[1,2]

[1]Temasek Laboratories, Nanyang Technological University, Singapore.

[2]National integrated Centre For Evaluation, Nanyang Technological University, Singapore.

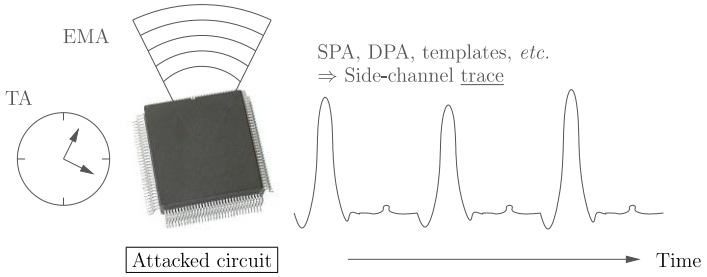
[3]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore.

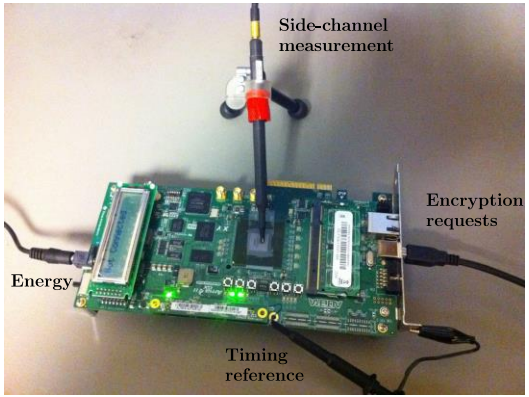
[4]School of Cyber Science and Technology, College of Computer Science and Technology, Zhejiang University, China



Side-Channel Analysis (SCA)

- Non-invasive (power, EM, timing, …)
- Powerful & practical. Ex:
  - *Keeloq*
  - *FPGA Bitstream encryption*
  - *Bitcoin wallets*
  - *…*
- Applications: Secret key recovery, etc …
- Serious threat for embedded systems





Side-Channel Analysis (SCA)

- Observations: Previous works are mostly tested on development boards (low noise) and some reported attack on commercial devices are very target-specific, and requires very specific knowledge

- Key Challenge: Can SCA be practically applied on real world target, where multiple operations running in parallel, resulting in *noisier* and *desynchronized* measurements?

Side-Channel Analysis (SCA)

- Attack on AES:
  - Conducting SCA on AES is **NOT** the main objective of this work!
  - This work evaluates practical attack on **non-ideal setting**

- Choosing Target Device:
  - A realistic target device is considered, **ARM Cortex A72**, which is widely used
  - It is representative of what we are using today and as such, it helps in understanding the practical challenges
  - Additional difficulties when compared to previous targets: OS interfering with the measurement, out-of-order execution, running frequency, I/O, etc

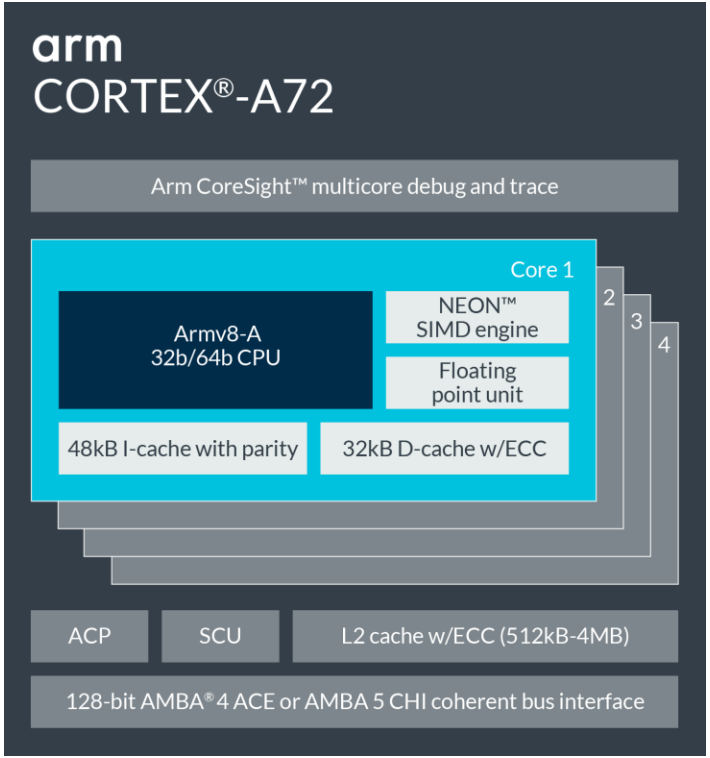
ARM Cortex A72

- ARMv8-A 64-bit instruction set
- Processor cluster has **one to four** cores with L1 and unified L2 caches
- Processor with **out-of-order**, speculative issue 3-way superscalar pipeline









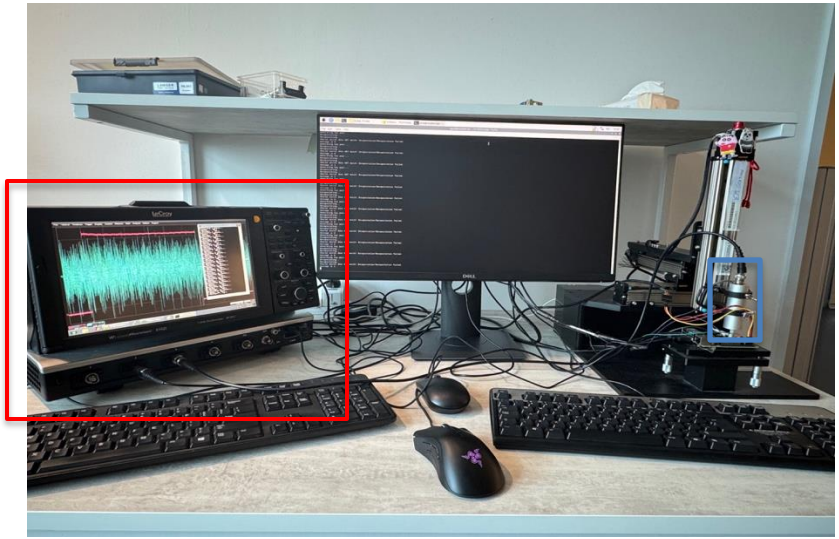
https://www.arm.com/products/silicon-ip-cpu/cortex-a/cortex-a72#RELATEDPRODUCTS

Contributions

- Conduct **comprehensive SCA of AES on the Raspberry Pi 4B** (ARM Cortex-A72), highlighting key challenges described earlier, such as noise, jitter, OS interference, and **out-of-order execution**
- Consider **multi-level threat** models, which helps to understand the difficulty or effort needed by an adversary to mount side channel on a more complex processor
- Present a **novel AES dataset** collected from a complex processor, offering more exploration and hopefully, help advancing explainability for SCA research

Experimental Setup

- Target Device: Raspberry Pi Model 4B with ARM Cortex-A72
- Algorithm: Advanced Encryption Standard (AES), C implementation
- Attack Type: EM SCA, Profiled & Non-Profiled



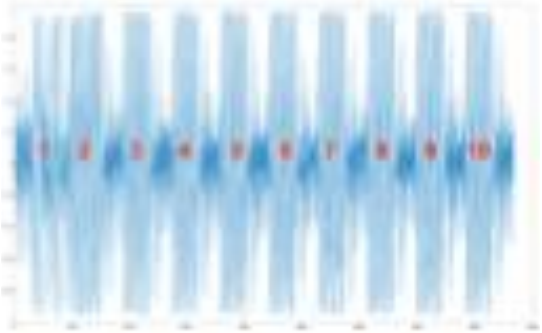
Riscure EM Probe with
In-built Amplifier
(to improve trace quality)
LeCroy Oscilloscope

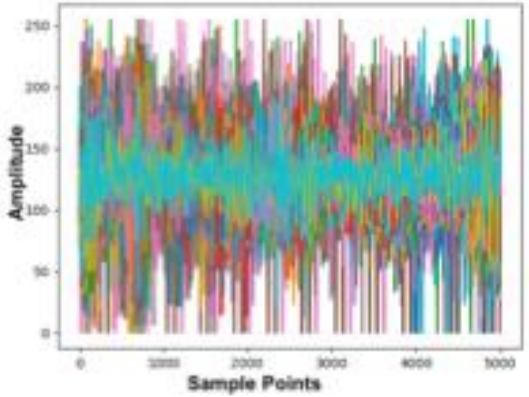
Key Challenges

- The presence of **noise**, due to other operations running in background and out-of-order execution, leading to **inconsistent waveform pattern**
- **Probe positioning** is non-trivial, due to desynchronization and noise
- **Preprocessing** methods are required for handling desynchronization:
  - Trace filtering based on trigger length
  - Realignment method such as Elastic Alignment (EA)
- More **number of traces** are required

Measurements





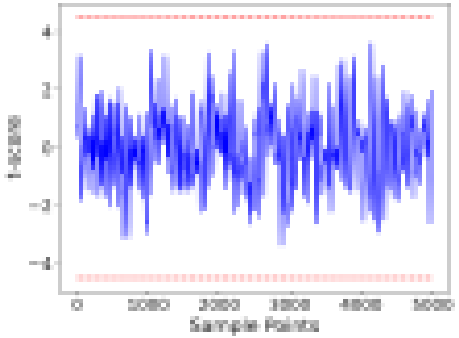


- Perform grid search to identify best position for probing
- Identify area of interest using visual inspection
- Observe high desynchronization level

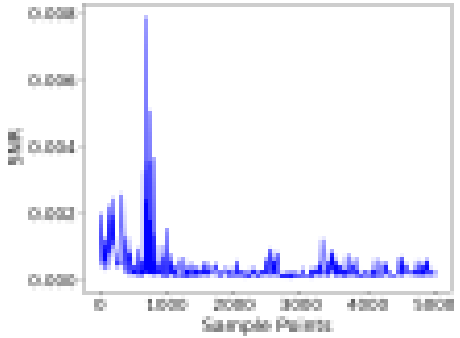
Measurements

Dataset	Summary
S-Trace	Only unprotected AES S-Box
M-Trace	Only masked AES S-Box
F-Trace	Full AES unprotected

- Initial assessment using leakage evaluation methods (TVLA and SNR) could not fully confirm the presence of leakages



(a) TVLA results



(b) SNR results

Threat Models

- **Basic Adversary**: Has no prior knowledge about the implementation, only perform non-profiling attack, with basic leakage model (i.e., Correlation Power Analysis or CPA with HW/ID leakage model)

- **Educated Adversary**: Has priori knowledge about the underlying countermeasures (i.e., CPA with aligned traces)

- **Advanced Adversary**: Has access to clone device and can build profiled model(s) (i.e., Template Attacks)

This Work

Table 1: Comparison of related works on practical side-channel attacks on ARM Cortex-A processors.

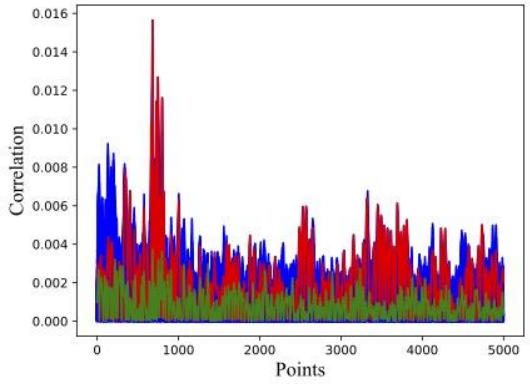
	Target Processor	Platform	Adversary Type
Balasch et al. [1]	ARM Cortex-A8	BeagleBone Black	Educated
Longo et al. [11]	ARM Cortex-A8	BeagleBone Black	Educated
Frieslaar and Irwin [6]	ARM Cortex-A7	Raspberry Pi 2	Educated
Wang et al. [18]	ARM Cortex-A53	Raspberry Pi 2B v1.2	Educated
Haas and Aysu [9]	ARM Hurricane (iPhone 7)	iPhone 7	Advanced
Wang et al. [18]	ARM Cortex-A53	Raspberry Pi 2B	Advanced
Current Work	ARM Cortex-A72	Raspberry Pi 4B	Basic to Advanced

Multi Level Threat Model

Out-of-Order Execution

Basic Adversary

- Perform basic CPA with basic HW & ID leakage model on measured traces, no preprocessing performed

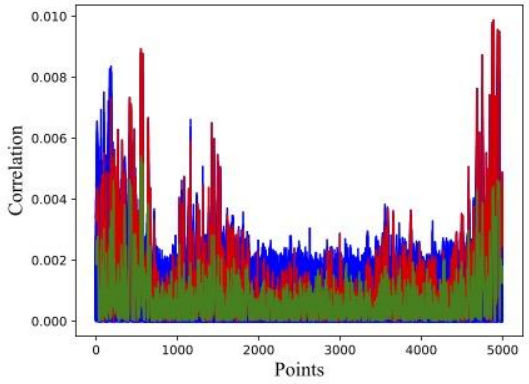


UNSUCCESSFUL ATTACK…

- From the figure, CPA on raw S-traces could not recover the correct key (green color)

Educated Adversary

- Perform basic CPA with basic HW & ID leakage model with realignment techniques, Elastic Alignment (EA) & Trace Filtering (based on trigger)



UNSUCCESSFUL ATTACK…

- From the figure, similarly, CPA on (EA) aligned S-traces could not recover the correct key (green color)

Advanced Adversary

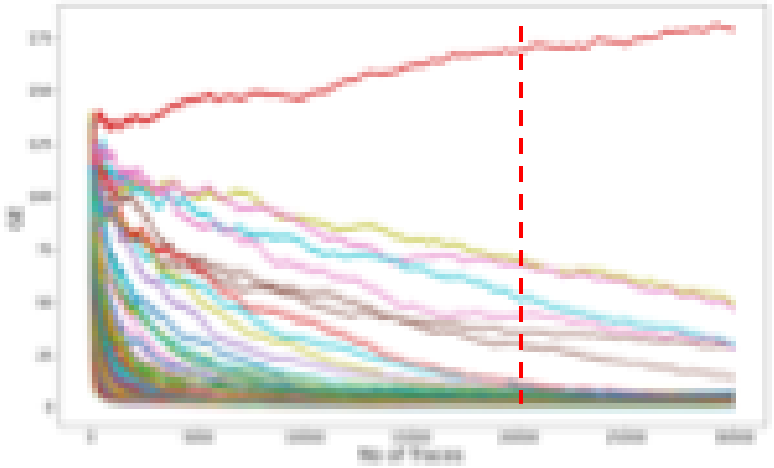
- The adversary could get access to clone devices and build profile(s) of the devices
  - Commonly used Template Attacks (TA) does not work in this case, due to desynchronization issue
- Alternative approach: use **Deep Learning (DL)** for profiling
  - DL has been used before in context of SCA, **DL-SCA**, with various successes
- Previous works on DL-SCA have only been tested on low end microcontrollers or on low noise target

Approach: DL-SCA

- ***Challenge***: finding proper hyperparameters
- ***Solution***: use random parameter search
- ***Evaluation metric***: Guessing Entropy (GE) & NtGE

Table 2: Range for hyperparameter random search.

Parameters	Range
Batch Size	100, 200, ... 1000
Learning Rate	1e-3, 5e-4, 1e-4, 5e-5, 1e-5
Optimizer	RMSprop, Adam
Layers	1,2,...,8
Neurons	10,20,50,100,200,300,400,500
Kernel Initializers	{random,glorot,he}_uniform
Pooling	{max,average}_pool
Pooling size	2,4,6,8,10
Convolution layer	1,2,3,4
Filters	4,8,12,16
Kernels	26, 28, ..., 52
Padding	0, 4, ..., 16



GE = 1 at 21,626 → NtGE = 21,626 (NtGE < ∞ indicates successful attacks)

Results with DL-SCA

Table 3: Evaluation under an Advanced Adversary. The number of features, the number of traces required for training and the number of traces required for attack are represented as "No of Feats", "No of Train", "No of Att". "NTGE" represents the least number of traces required to attain GE as 1.

Dataset	No of Feats.	No of Train	No of Att	NTGE
Raw *S-trace*	5,000	400,000	30,000	23,366
Filtered *S-trace*	405	100,000	30,000	6,132
EA *S-trace*	500	100,000	30,000	3,902
Raw *F-trace*	7,000	150,000	50,000	49,799
Filtered *F-trace*	7,000	100,000	30,000	21,626
EA *F-trace*	7,000	150,000	50,000	49,989
Raw *M-trace*	1200	400,000	200,000	197,578
Filtered *M-trace*	624	400,000	200,000	92,670
EA *M-trace*	1,000	400,000	200,000	193,961

THE CORRECT KEY BYTE CAN BE SUCCESSFULLY RECOVERED!!

Discussion

- We observed that preprocessing techniques such as EA and Trace Filtering helps to improve the attacks
- As more devices are incorporating advanced processors, it brings up the issue of cost vs security trade-off
  - While SCA countermeasures are costly
  - An ***advanced*** adversary with substantial resources are required to break even an **unprotected** implementation on complex processor
- **Future Works**: Better profiled model? Better preprocessing methods? More complex targets?

Conclusion

- Demonstrate attack efficiency on **complex processors**, extending prior work that focused primarily on microcontrollers
- Highlight **key challenges** to in performing SCA on a processor with out-of-order execution, showing how practical attacks are not trivial
- Investigate the **effort/cost required by the attacker to recover the key in a practical scenario** using a multi-level threat model
- **Advanced adversary** is required to **successfully recover the secret keys** under challenging conditions like noise and desynchronization
- Provide **novel public dataset** collected from our experiments

Dataset

- We have also submitted the data as part of the CHES challenge, which hopefully could help:
  - Achieving better understanding of the leakage
  - Advancement of DL-SCA research

- Scan the QR code for instruction on how to download and use the dataset



Thank You!