EM Side-Channel Attacks on SoCs

Kiem Lukas

December 6, 2023

Content

- Basics of EM side-channel attacks
- Introduction
- Background
- Software-based AES
- Hardware-based AES
- NEON
- Conclusions
- Countermeasures

Basics of EM side-channel attacks

Basics

- Exploit the electromagnetic emanations from an electronic device [1]
- Powerful hardware attacks
- Non-Invasive
- Higher SNR than power attacks
- Successful against cryptographic implementations [2]
- Goal: Extract secret key

Setup

- Spectrum analyzer / oscilloscope
- Pre-amplifier (plus hardware filters)
- Near field probe
- Equipment under test (EUT)

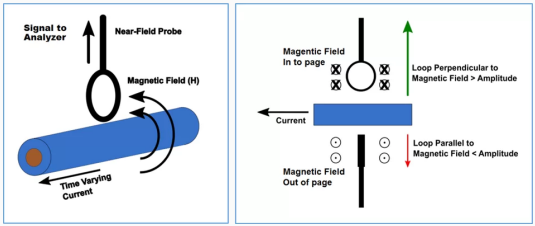


Figure 1: Magnetic field probe orientation and position affect measurement amplitude. [3]

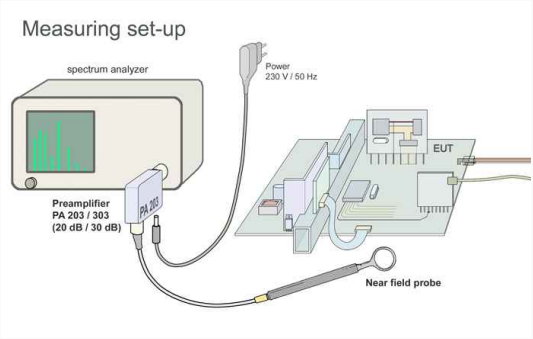


Figure 2: Measuring setup [4]

Steps

1. Measure the emitted electromagnetic radiation
2. Preprocessing
3. Signal analysis (hamming distance/weight)

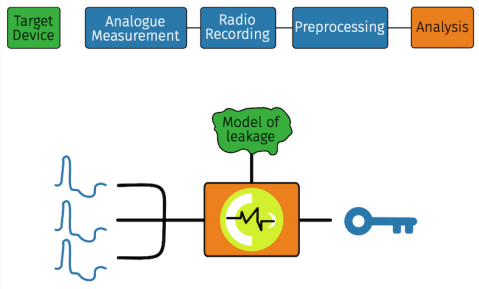


Figure 3: EM-based attacks [1]

Types of EM side-channel attacks

1. Simple Electromagnetic Analysis (SEMA)
   - Try to interpret the data traces directly
   - Simple and fast

2. **Differential Electromagnetic Analysis (DEMA)**
   - Collect large numbers of traces
   - Run differential statistical methods on the data to identify correlations
   - More robust but also more complicated

Introduction

Introduction

- Investigate electromagnetic-based leakage from executing cryptographic workloads on...
  1. AES executed by an OpenSSL server on the ARM core
  2. the proprietary AES co-processor
  3. the NEON core, including bit-sliced AES

Background

Used DUT

- BeagleBone Black Board
- with AM335x "Sitara" SoC
- Sub-systems communicate via NoC
- Focus on
  1. MPU
  2. cryptographic co-processor (lacks public documentation)

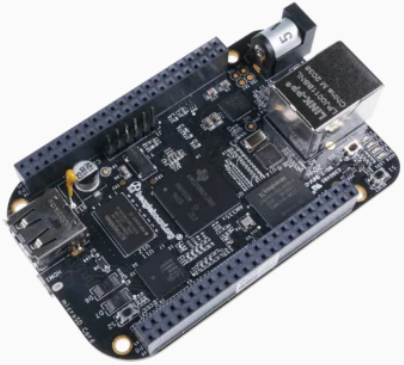


Figure 4: BeagleBone Black with AM3358 Sitara [5]

Leakage detection

Test Vector Leakage Assessment (TVLA) [6]

- Construct two sets of test vectors
  - V_0 (single (semi-)fixed vector)
  - V_1 (large number of uniformly random chosen vectors)
- Steps:
  1. Randomly select a test vector type $b \xleftarrow{\$} 0, 1$
  2. Randomly select a test vector from the chosen set $x_i \xleftarrow{\$} V_b$
  3. Process the vector and add the resulting trace λ_i to a set Λ_b
  4. Compute the t-statistic trace $t = \frac{\bar{\Lambda}_0 - \bar{\Lambda}_1}{\sqrt{\frac{\sigma_0^2}{|\Lambda_0|} + \frac{\sigma_1^2}{|\Lambda_1|}}}$
  5. Check if $|t[j]| > \tau$
  6. If yes, significant leakage is detected at the j-th sample

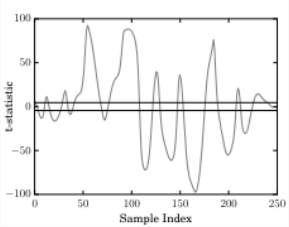


Figure 5:
Fixed-versus-random test [7]

Software-based AES

Introduction

Imagine target device as a communications device engaged in a TLS-based session with an OpenSSL server executing AES in software

- Attacker can observe $c_i = DUT_k^{AES-128-CBC}(m_i) \rightsquigarrow \lambda_i$
  - AES-128 encryption, in CBC mode
  - unknown plaintext m_i
  - key k
  - known ciphertext c_i (since it its communicated across the network)

Probe location

- Manual scan of the SoC surface
- While executing three kernels
  1. A set of memory intensive operations
  2. A spin-lock
  3. A set of computationally intensive operations (AES encryption)
- Cycling through the kernels and monitoring the frequency response

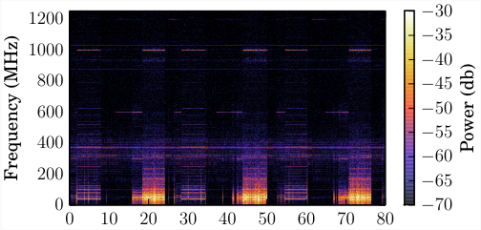


Figure 6: Spectogram plot. Three iterations. [7]

Probe location

- Two regions identified
  1. Memory intensive kernel (No further investigation)
  2. **Centrally on the AM335x**

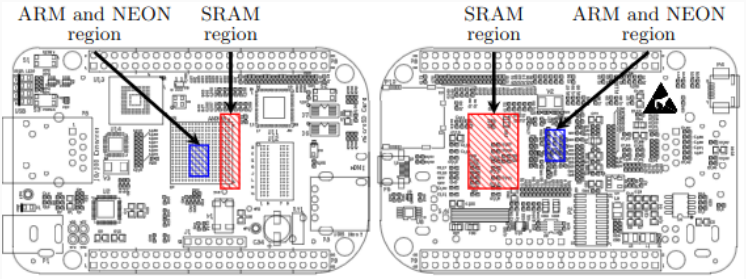


Figure 7: BeagleBone Black schematic. Front-side on the left, back-side on the right [7]

Interrupt detection and synchronisation

- The OS may preempt a user process
- Automate identification using the trace alignment scores
  1. Manually identify an uninterrupted trace → template
  2. Perform coarse alignment
  3. Calculate the least squares score
  4. If score > threshold → interrupted
- Two choices
  1. **Discarding**
  2. Cleaning (pruning the interrupted region)

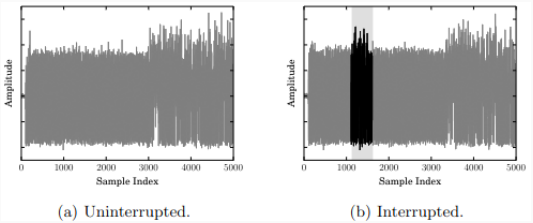


Figure 8: Impact of interrupts on the acquisition process [7]

Clock scaling

- OS scales the clock frequency, to optimise power consumption
- Stabilise at 600 MHz once the OpenSSL process became active
- Create templates at each clock frequency
- Comparison between target and template trace reveals the clock frequency used and yields a subset of traces with the same clock frequency

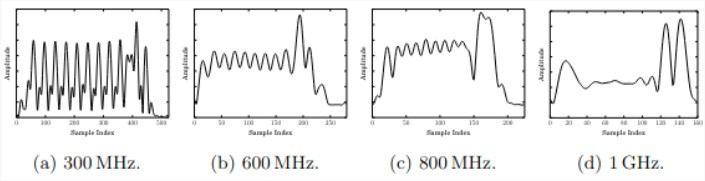


Figure 9: Impact of clock scaling on acquisition process [7]

Analysis - Acquisition

- Acquisition phase
  1. Bulk acquire $n = 1000$ traces, each with $l = 256$ encryptions $\rightarrow$ 4kB of traffic per trace
  2. Deal with systemic noise (filtering for interrupts and clock scaling) $\rightarrow$ discarding $\sim 20\%$
  3. From each trace, extract a fragment for each encryption operation; match with the associated ciphertexts
  4. Realign each sub-trace; discard any low-quality cases $\rightarrow$ discarding $\sim 5\%$
- Took $\sim 6min$
- Next step: Exploit the leakage

Analysis - Exploitation

- Single-bit correlation-based attack
- Target the S-box look-up in the final AES round
- Correct hypothesis can be clearly distinguished using around 20.000 sub-traces ($\sim$ 100 traces)

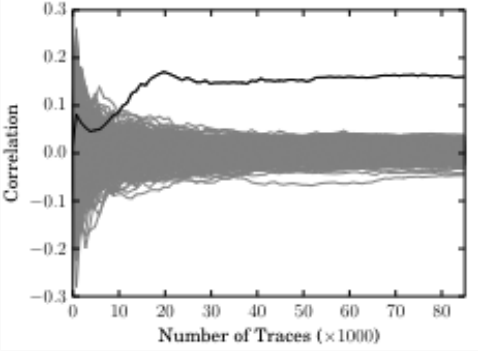


Figure 10: Single-bit correlation. [7]

Hardware-based AES

Introduction

- Focus on hardware-based execution of AES
- Using the cryptographic co-processor
- Attacker still can observe $c_i = DUT_k^{AES-128-CBC}(m_i) \rightsquigarrow \lambda_i$
- Imagine as Full Disk Encryption (FDE)
  - c_i is known because it lies as ciphertext on a readable disk

Signal hunting - failure

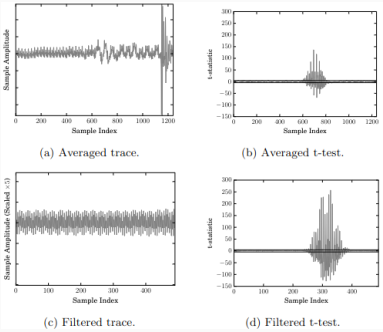
- Scant documentation of the cryptographic co-processor
- Unclear how AES is computed
- No identifiable periodic leakage signature found
- Failed to locate leakage
  - Fixed-versus-random test with probe locations from attack 1 failed
  - Manual scan of the AM335x surface failed

Signal hunting - found

- Managed to detect the DMA strobes
- Not usable as a synchronous trigger because any memory intensive instruction can cause false positives
- Solution
  - Saturating the DMA engine with other work
  - Forces driver into a non-DMA fall-back mode $\rightarrow$ issues interrupts for any memory management
  - Can be used as a trigger for AES operations on the co-processor
- With the trigger mechanism active, several probes were placed at various locations and leakage tests were executed $\rightarrow$ success with hamming distance

Analysis - Acquisition

- Acquisition phase was performed as follows:
  1. Saturate the DMA mechanism so that non-DMA fall-back mode is used
  2. Acquire $n = 500.000$ traces, each with 1 encryption and $l = 1.000$ trials; match these traces with the ciphertexts
  3. Apply wavelet post-processing to each trace to maximise SNR
- The acquisition took around 3 days
- To exploit: **single-bit correlation-based attack**
- Succeeded in recovering k



(a) Averaged trace. (b) Averaged t-test.

(c) Filtered trace. (d) Filtered t-test.

Figure 11: Single-bit correlation. [7]

Analysis - Conclusions

- Effort to identify leakage far outweighs that of acquisition and attack phases
- Initial failure because
  - the trigger mechanism is not accurate enough to align traces correctly and/or
  - the co-processor is protected against side-channel attacks (if yes, it is not clear which countermeasures)

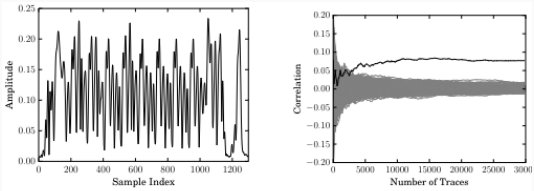
NEON

Introduction

- NEON is a general-purpose SIMD extension to Cortex A-series ARM cores
- To accelerate cryptographic workload and deliver constant execution time
- Attacker can observe $m_i = DUT_k^{AES-128-CBC}(c_i) \rightsquigarrow \lambda_i$
- AES is realized using the NEON-based bit-sliced implementation in OpenSSL (if c_i 128 bytes or more)

Attack

- Experimental environment $\rightarrow$ same as in first attack
- Key hypotheses based on Hamming weight of the intermediate state after the first round *InvSubBytes* operation
- **Attack succeeds**
  - $\sim$ 5.000 traces
  - requiring $5000 * 128 = 625kB$ of ciphertext



(a) Single averaged trace. (b) Single byte correlation.

Figure 12: Decryption using the NEON-based bit-sliced implementation of AES in OpenSSL [7]

Conclusions

Results

Section	Operation	Implementation	Hardware	Trigger	Acquisitions	Data
3	Decryption	T-tables	ARM core	GPIO-based	3,000	46 kB
3	Encryption	T-tables	ARM core	Network-based	100	400 kB
4	Encryption	Hardware	Co-processor	DMA-based	500,000	7 GB
5	Decryption	Bit-sliced	NEON core	GPIO-based	5,000	625 kB

Table 1: Summary of results.

Countermeasures

Countermeasures

- Signal Information Reduction [8]
  - Hiding & Masking Schemes
  - Randomization and Refreshing
- Signal Strength Reduction
  - Shielding
  - Low-level Routing
  - Spatial Randomization

References

[1] J. Hertz, **Em side-channel attacks on cryptography,** (2023), [Online]. Available: https://www.allaboutcircuits.com/technical-articles/em-side-channel-attacks-on-cryptography/ (visited on 12/02/2023).

[2] Wikipedia, **Electromagnetic attack,** (2023), [Online]. Available: https://en.wikipedia.org/w/index.php?title=Electromagnetic_attack&oldid=1180522497 (visited on 12/03/2023).

[3] Siglent, **Electromagnetic compliance: Troubleshooting with near-field and current probes,** (), [Online]. Available: https://siglentna.com/application-note/electromagnetic-compliance-troubleshooting-near-field-current-probes/ (visited on 12/03/2023).

References ii

[4] L. EMV-Technik, **Rf3 mini set,** (), [Online]. Available: https://www.langer-emv.de/en/product/rf-passive-30-mhz-up-to-3-ghz/35/rf3-mini-set-near-field-probes-30-mhz-up-to-3-ghz/855 (visited on 12/03/2023).

[5] beagleboard, **Beaglebone black,** (), [Online]. Available: https://www.beagleboard.org/boards/beaglebone-black (visited on 12/03/2023).

[6] G. Goodwill, B. Jun, J. Jaffe, and P. R. C. R. Inc., **A testing methodology for side channel resistance validation,** (), [Online]. Available: https://csrc.nist.rip/CSRC/media/Events/Non-Invasive-Attack-Testing-Workshop/documents/08_Goodwill.pdf.

References iii

[7] J. Longo, E. D. Mulder, D. Page, and M. Tunstall, **Soc it to em: Electromagnetic side-channel attacks on a complex system-on-chip,**. [Online]. Available: https://www.iacr.org/archive/ches2015/92930599/92930599.pdf.

[8] C. in Cybersecurity Community, **Electromagnetic side-channel attacks and potential countermeasures,** (2021), [Online]. Available: https://www.youtube.com/watch?v=YFTfgzkz3EI (visited on 12/05/2023).