A Novel Completeness Test for Leakage Models and its Application to Side Channel Attacks and Responsibly Engineered Simulators

Si Gao[1] and Elisabeth Oswald[1]

[1]Digital Age Research Center (D!ARC), University of Klagenfurt, Austria

May 27, 2022

Outline

1. Side channel analysis: achievements & challenges
2. Finding a complete set of *state*
3. Application: dissecting Attacks
4. Application: leakage simulators
5. Ethical considerations

Side channel analysis

SCA

- Attacks based on information leakage (timing, power consumption, electromagnetic emission, etc.)
- Recover the secret key potentially within a few minutes (1 — several million traces)

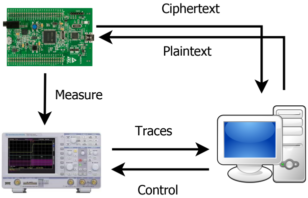
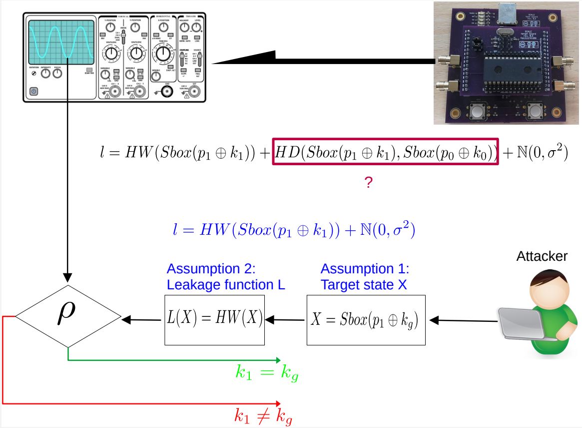


Figure: Side Channel Analysis

Side channel analysis

As a (non-profiled) attacker...



$$l = HW(Sbox(p_1 \oplus k_1)) + \boxed{HD(Sbox(p_1 \oplus k_1), Sbox(p_0 \oplus k_0))} + \mathbb{N}(0, \sigma^2)$$

?

$$l = HW(Sbox(p_1 \oplus k_1)) + \mathbb{N}(0, \sigma^2)$$

Attacker

Assumption 2:
Leakage function L

Assumption 1:
Target state X

ρ

$L(X) = HW(X)$

$X = Sbox(p_1 \oplus k_g)$

$k_1 = k_g$

$k_1 \neq k_g$

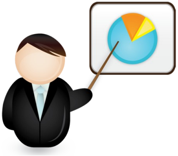
Side channel analysis

From an attacker's perspective...

Correct assumptions could be more costly than the secret key...

$$l = HW(Sbox(p_1 \oplus k_1)) + \boxed{HD(Sbox(p_1 \oplus k_1), Sbox(p_0 \oplus k_0))} + \mathbb{N}(0, \sigma^2)$$

$$l = HW(Sbox(p_1 \oplus k_1)) + \mathbb{N}(0, \sigma^2)$$



"Your assumptions are (partly) wrong!"

Attacker

"I already got the key, whatever…."

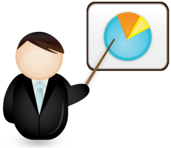
Side channel analysis

For evaluation/certification...

Partly effective countermeasures are certainly not desirable...

$$l = HW(Sbox(p_1 \oplus k_1)) + \boxed{HD(Sbox(p_1 \oplus k_1), Sbox(p_0 \oplus k_0))} + \mathbb{N}(0, \sigma^2)$$

$$l = HW(Sbox(p_1 \oplus k_1)) + \mathbb{N}(0, \sigma^2)$$



"Your assumptions are (partly) wrong!"

Security Evaluation

"Er… then what does my results mean?"

Our contribution

We propose/clarify in this paper...

- "Leakage models" contain both X and L
    - Emphasis on X (there exist other solutions for L)
- "Completeness" test
    - "Completeness": X contains all relevant states for l
    - Using F-test to verify whether a selected X is complete (or not)
- Impacts of "completeness"
    - For attacks: revealing unexpected new leakage
    - For leakage simulators: finding leaks that would otherwise missed by overly-simplified models

Outline

1 Side channel analysis: achievements & challenges

2 Finding a complete set of *state*

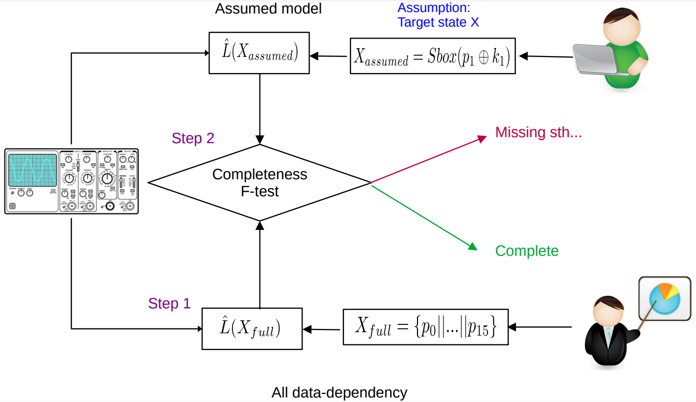
3 Application: dissecting Attacks

4 Application: leakage simulators

5 Ethical considerations

Finding a complete X

Road map



Assumed model

Assumption:
Target state X

$\hat{L}(X_{assumed})$

$X_{assumed} = Sbox(p_1 \oplus k_1)$

Step 2

Completeness
F-test

Missing sth...

Complete

Step 1

$\hat{L}(X_{full})$

$X_{full} = \{p_0||...||p_{15}\}$

All data-dependency

Step 1: Construct a *full* model

Defining a full model X_{full} (aka all leakage):

All-input model

For unmasked AES-128, let $\hat{X}$ = all 128-bit plaintext

- All leakage will be captured
- Requires $> 2^{128}$ traces to attack/analysis
- Collapsed models
  - Bound the inputs (as in leakage detection)
  - E.g. each byte in AES-128 takes only 11...1 or 00...0
  - Now the input space is bounded to 2^{16}

Step 2: Completeness F-test

F-test for ANOVA

$\hat{L}(p_0||...||p_{15})$ vs. $\hat{L}(Sbox(p_1 \oplus k_1))$

- Does the latter miss something?
  - $F > th$, $\hat{L}(Sbox(p_1 \oplus k_1))$ misses some factor that has significant contribution to the observed leakage
  - otherwise, complete up to the statistical power (i.e. provided number of traces)

Finding a complete X

Put it all together,

Collapsed F-test for completeness

1. Construct a full model $\tilde{L}(X_{full})$ and an assumed model $\tilde{L}(X_{assumed})$
2. Comparison in F-test: if $F > th$, $X_{assumed}$ is not complete

Example: rejected because $HD(S(p_0 \oplus k_0), S(p_1 \oplus k_1))$ is missing

Outline

1 Side channel analysis: achievements & challenges

2 Finding a complete set of *state*

3 Application: dissecting Attacks

4 Application: leakage simulators

5 Ethical considerations

Goal

Not neccessarily "for the attacker"...

E.g. $Sbox(p_0 \oplus k_0) \oplus Sbox(p_1 \oplus k_1)$

- takes intensive effort to find
- 2 relevant key bytes ((k_0, k_1) vs. k_1)
- reveal unexpected μ-arch features (in a profiling setup)

Affine masked AES-128

From ANSSI, @https://github.com/ANSSI-FR/SecAESSTM32

Scheme

$C(x) = rm \otimes x \oplus ra$

- One multiplicative mask rm and one additional mask ra
- Sbox input mask r_{in} and r_{out}

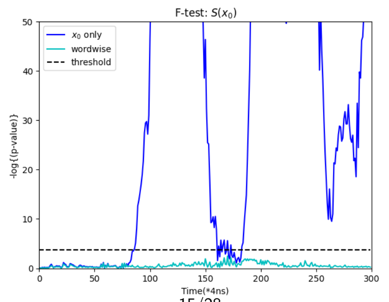
$$S'(rm \otimes x \oplus r_{in}) = rm \otimes S(x) \oplus r_{out}$$

- ra different for each byte
- rm, r_{in}, r_{out} shared within one encryption

Affine masked AES-128

When computing masked $S'(C(x_0))$...

- Assumption: all relevant term for $S'(C(x_0))$ could leak
  - $X_{assumed} = \{x_0 = p_0 \oplus k_0 || ra_0 || r_{in} || r_{out}\}$
- $-log(pv) > th \Rightarrow$ not complete
- Not complete (blue line in the figure)

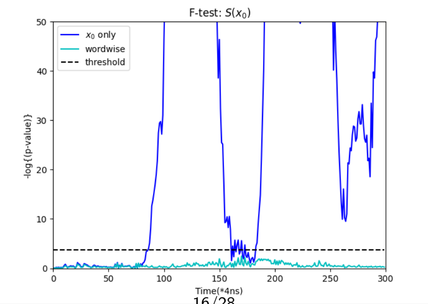


Affine masked AES-128

Why?

Cortex M3 is a 32-bit core

- Load/Store bus is likely also 32-bit
- LDRB: always load word, discarding unnecessary bytes
- Word-wise load for all instructions ⇒ complete(cyan line)

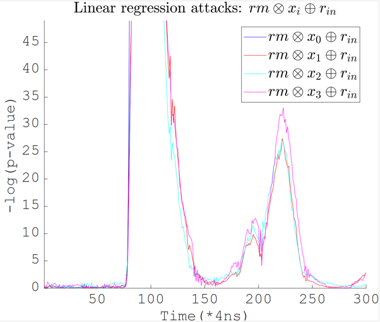


Affine masked AES-128

Verifying word-wise leakage

Leakage of all 4 bytes when computing the first S-box

- all 4 bytes leak simultaneously (right figure)



Affine masked AES-128

Impact on attacks?

- x_i and x_j on different points on the trace
- can use the same point
- Bivariate ⇒ Univariate
- more details in the paper...

Outline

1 Side channel analysis: achievements & challenges

2 Finding a complete set of *state*

3 Application: dissecting Attacks

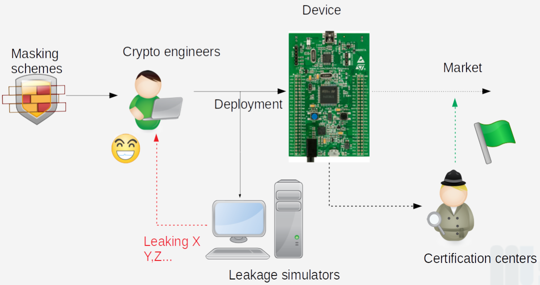
4 Application: leakage simulators

5 Ethical considerations

Application: leakage simulators

Leakage simulators

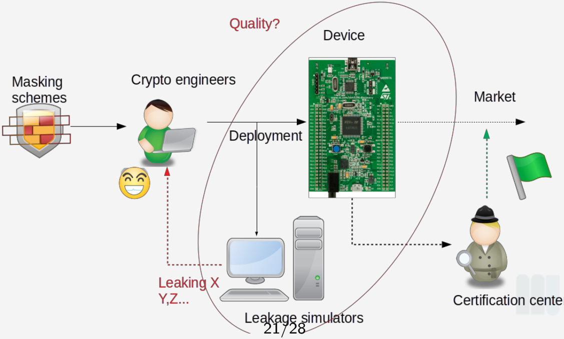
- Early stage feedback ⇒ Cheaper/faster development
- Leakage reasoning ⇒ Targeted security patch



Application: leakage simulators

Leakage simulators

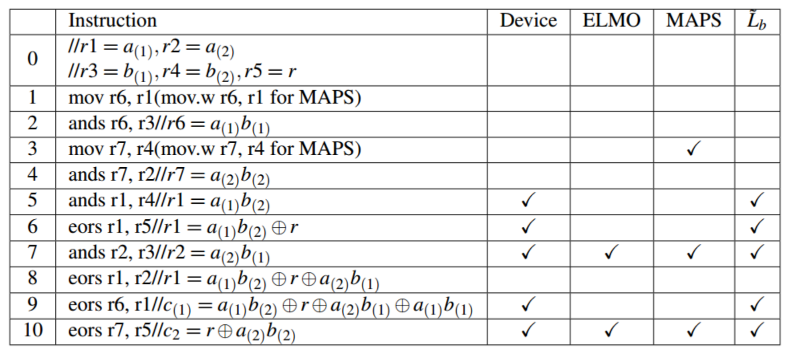
- Existing tools (Cortex M3, binary code level)
  - ELMO/ELMO*: model built from measurements
  - MAPS: RTL code from ARM
- Challenge: quality? ← completeness test



Application: leakage simulators

Target gadget

A bitwise 2-share ISW multiplication

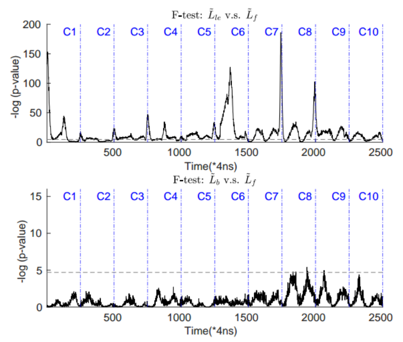


	Instruction	Device	ELMO	MAPS	$\tilde{L}_b$
0	$//r1 = a_{(1)}, r2 = a_{(2)}$ $//r3 = b_{(1)}, r4 = b_{(2)}, r5 = r$				
1	mov r6, r1(mov.w r6, r1 for MAPS)				
2	ands r6, r3$//r6 = a_{(1)}b_{(1)}$				
3	mov r7, r4(mov.w r7, r4 for MAPS)			✓	
4	ands r7, r2$//r7 = a_{(2)}b_{(2)}$				
5	ands r1, r4$//r1 = a_{(1)}b_{(2)}$	✓			✓
6	eors r1, r5$//r1 = a_{(1)}b_{(2)} \oplus r$	✓			✓
7	ands r2, r3$//r2 = a_{(2)}b_{(1)}$	✓	✓	✓	✓
8	eors r1, r2$//r1 = a_{(1)}b_{(2)} \oplus r \oplus a_{(2)}b_{(1)}$				
9	eors r6, r1$//c_{(1)} = a_{(1)}b_{(2)} \oplus r \oplus a_{(2)}b_{(1)} \oplus a_{(1)}b_{(1)}$	✓			✓
10	eors r7, r5$//c_2 = r \oplus a_{(2)}b_{(2)}$	✓	✓	✓	✓

Application: leakage simulators

Completeness test

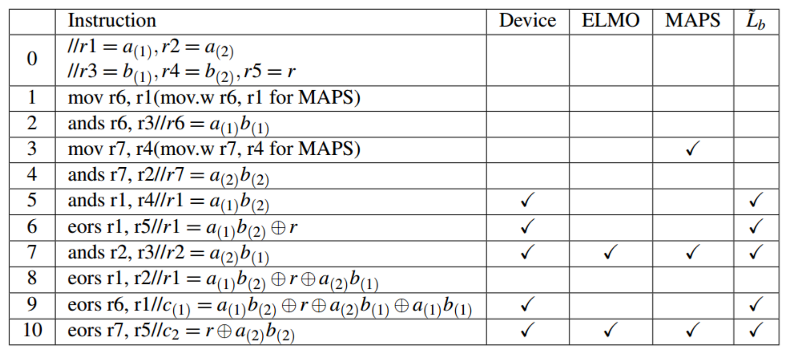
- $\tilde{L}_{le}$: a superset for both ELMO and MAPS
    - both ELMO and MAPS fail in almost every cycle...
- $\tilde{L}_b$: recursively adding missing factors to $\tilde{L}_{le}$



Application: leakage simulators

Impacts on leakage detections

- both ELMO and MAPS miss leaks
- better models ⇒ more accurate detections



	Instruction	Device	ELMO	MAPS	$\tilde{L}_b$
0	$//r1 = a_{(1)}, r2 = a_{(2)}$ $//r3 = b_{(1)}, r4 = b_{(2)}, r5 = r$				
1	mov r6, r1(mov.w r6, r1 for MAPS)				
2	ands r6, r3$//r6 = a_{(1)}b_{(1)}$				
3	mov r7, r4(mov.w r7, r4 for MAPS)			✓	
4	ands r7, r2$//r7 = a_{(2)}b_{(2)}$				
5	ands r1, r4$//r1 = a_{(1)}b_{(2)}$	✓			✓
6	eors r1, r5$//r1 = a_{(1)}b_{(2)} \oplus r$	✓			✓
7	ands r2, r3$//r2 = a_{(2)}b_{(1)}$	✓	✓	✓	✓
8	eors r1, r2$//r1 = a_{(1)}b_{(2)} \oplus r \oplus a_{(2)}b_{(1)}$				
9	eors r6, r1$//c_{(1)} = a_{(1)}b_{(2)} \oplus r \oplus a_{(2)}b_{(1)} \oplus a_{(1)}b_{(1)}$	✓			✓
10	eors r7, r5$//c_2 = r \oplus a_{(2)}b_{(2)}$	✓	✓	✓	✓

Outline

1 Side channel analysis: achievements & challenges

2 Finding a complete set of *state*

3 Application: dissecting Attacks

4 Application: leakage simulators

5 Ethical considerations

Ethical considerations

Threats of proportional leakage simulators

ELMO/ELMO*

- Proportional: “close to realistic measurements”
- Pros
  - Good for attack estimation
  - Can estimate power consumption
- Cons
  - Free templates for attackers?

Ethical considerations

Nominal leakage simulators

- Nominal: finding state X, not estimating L
- Pros
  - Good for leakage detection
  - Cannot be used as "free templates"
- Cons
  - Qualitative only

The End

Questions?