

Share-slicing: Friend or Foe?

Si Gao ^{1,2} Ben Marshall ¹ Dan Page ¹ Elisabeth Oswald ^{1,2}

¹University of Bristol, Bristol, UK

²University of Klagenfurt, Klagenfurt, Austria

September 4, 2020

Outline

- 1 Intro
- 2 Evaluations in practice
- 3 Read pass the "headlines"...
- 4 Discussion

SCA

- Attacks based on information leakage
- Recover the secret key potentially within a few minutes

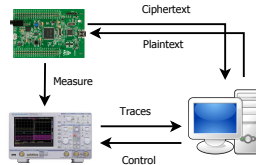


Figure: Side Channel Analysis

Academia

"Countermeasures to rescue!"

- Threshold implementation [NRR06]
- Domain-Oriented Masking [GMK16]
- Various schemes available!



Industry

"Oh no... what should I do?"



Academia

"OK...that is trickier, but still do-able"

- Look-up table based approaches
- Global look-up table
- Re-computation method [Coron14]



Industry

"Emm... I can update codes, but not revoke all devices..."



Academia

"How about bit-sliced masking?"

- Utilise small gadgets (eg. AND2)
- Moderate memory cost, flexible
- Difficult for chaining mode (eg. CBC-ENC)



Industry

"Actually my application is quite memory-tight. Any other suggestions?"



Academia

"OK. Here are some results:"

- ISW multiplication [ISW03]
- Multiplication in bounded-moment model [BDF+17]
- Proof + some codes on Github
- Performance on ARM [GR17,GJRS18]



Industry

"Fair enough. Let us do this!"



Academia

"But please be careful with your implementations:"

- Pitfalls (eg. bad randomness)
- Model v.s. Practice
 - "order reduction theorem" [BGGRS14]



Industry

"Brilliant! I will implement one of this."



Academia

On a code-level, a d -share scheme:

- is seldom $(d - 1)$ -order secure
 - Few would do the full "diagnose-and-cure" cycle
- Even if it is $d - 1$ -order secure...
 - Weak protection when d is small



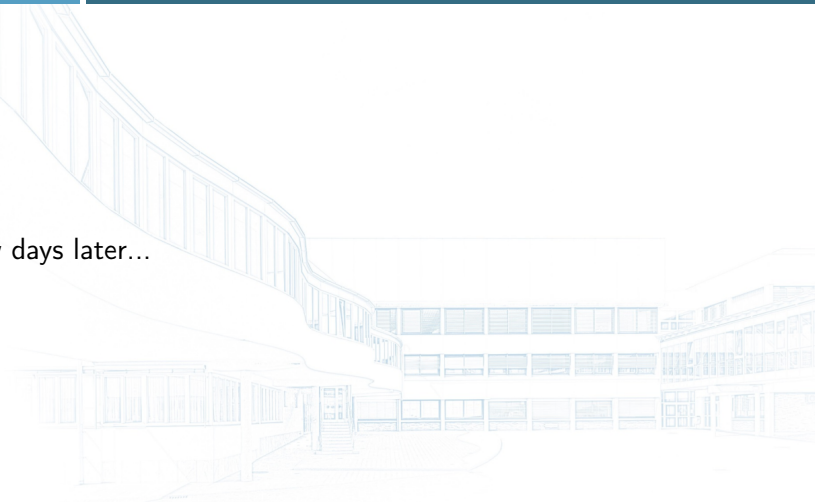
Industry

"Alright...I will keep that in mind."



Masking: theory to practice

A few days later...



Academia

"OK... you sure it is working properly?"



Industry

"Professor, I have implemented my 4-share secure AES!"

- Barthe et al.'s secure multiplication [BDF+17]
 - Parallel share processing —> efficiency
 - **Share-slicing: all shares in one register**



Academia

"Emm...maybe you are right?"



Industry

"Should be OK, I guess"

- Only claiming 1st order secure
 - "order reduction theorem"
- Previous study said so [JS17]
 - if we ignore physical coupling [CEM18,LBS19]



Academia

"Or is it really correct?"



Outline

- 1 Intro
- 2 Evaluations in practice
- 3 Read pass the "headlines"...
- 4 Discussion

Setups

ARM M3 (NXP LPC1313) & M0 (NXP LPC1114)

- Working at 12 MHz
- Scope sampling at 250 MSa/s
- Code written in Thumb assembly
- **Unused bit-width**
 - Constants: all 0-s (trivial yet waste)
 - Randomise: worst for the attacker (costly)
 - Repetition: same unshared value

Target secure AND2 tailored:

- Transition-leakage reduced to minimal

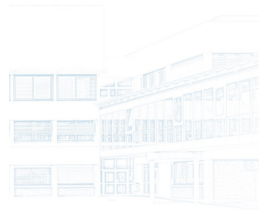
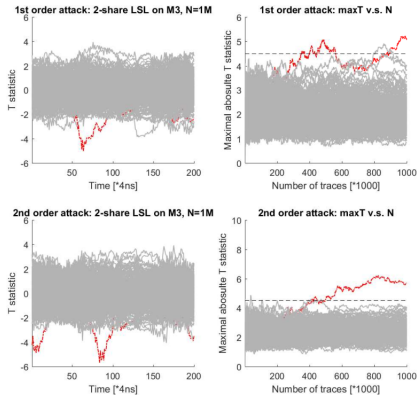
```
//rotation of b by 1
mov     r8, r8           //Clear HD
lsls   r6, r1, #1       //r6=b<<1
mov     r8, r8           //Clear HD

//The following computation has been commented out
//ands   r6, r5           //r6=(b<<1)&0xffffffff
//lsrs   r7, r1, #3       //r7=(b>>3)
//bics   r7, r5
//eors   r6, r7           //r6=(b<<<1)
//...
```


Evaluations

Barthe et al's multiplication on M3: 2-share version

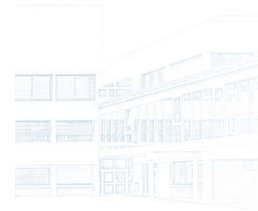
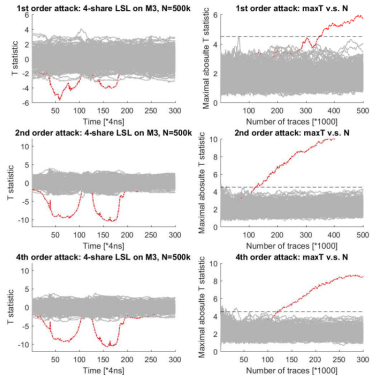
- All other 30 bits random
- correct key(red)/incorrect keys(gray)
- 1st order \approx 2nd order, not a big deal



Evaluations

Barthe et al's multiplication on M3: 4-share version

- All other 28 bits random
- correct key(red)/incorrect keys(gray)
- 2nd order is better/1st order still exists



Academia

"Have you checked the model **assumptions**?"



Industry

"Wait...how can it be? "



Academia

"Err..."



Industry

"I only checked the 'implementation defaults' section:"

- Mostly hardware perspective
- **What does it mean in software?**



Outline

- 1 Intro
- 2 Evaluations in practice
- 3 Read pass the "headlines"...
- 4 Discussion

Independent assumption

"Each share leaks independently": specifically,

- Each share has its own leakage function
- No interaction/cross-talk

this larger power consumption is the following linear model:

$$\mathbf{L}_c = \alpha_c^1 \cdot \mathbf{G}_c^1(\mathcal{Y}_c(1)) + \alpha_c^2 \cdot \mathbf{G}_c^2(\mathcal{Y}_c(2)) + \dots + \alpha_c^{n_c} \cdot \mathbf{G}_c^{n_c}(\mathcal{Y}_c(n_c)) + \mathbf{R}_c. \quad (2)$$

with all α_c^j 's $\in \mathbb{R}$. Contrary to the additive noise assumption that is only used

In hardware masking

Such assumption is usually supported by:

- Parallel separated sub-circuits (motivated by MPC)
 - No logical crosstalk
- “Keep Hierarchy”
 - No cross-talk from the synthesiser

this larger power consumption is the following linear model:

$$\mathbf{L}_c = \alpha_c^1 \cdot \mathbf{G}_c^1(\mathcal{Y}_c(1)) + \alpha_c^2 \cdot \mathbf{G}_c^2(\mathcal{Y}_c(2)) + \dots + \alpha_c^{n_c} \cdot \mathbf{G}_c^{n_c}(\mathcal{Y}_c(n_c)) + \mathbf{R}_c. \quad (2)$$

with all α_c^j 's $\in \mathbb{R}$. Contrary to the additive noise assumption that is only used

Software with share-slicing

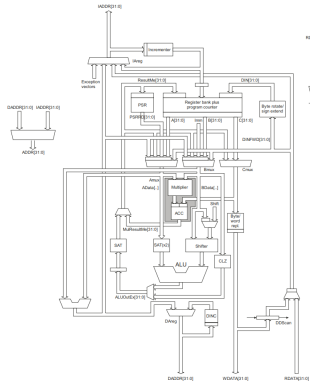
Independent assumption becomes

- following the same level of scrutiny
- **each gate in the ALU connects with only 1 bit of the register**

Independent assumption: in software

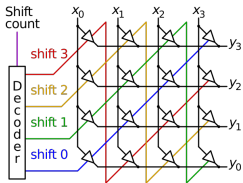
Software with share-slicing

But is that even possible?



Zoom into the shifter

Shifter can be our first headache:



Other parts of the ALU (eg. adders) can also contribute

Verifying independent assumption

Testing on shift alone

Already illustrates the issue:

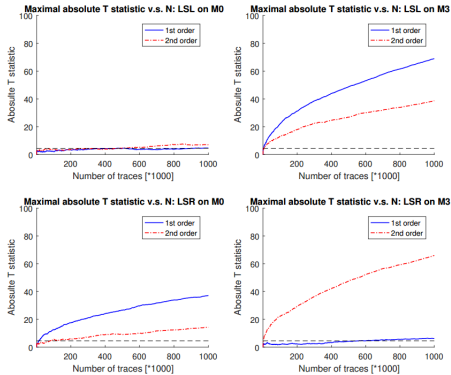


Figure 6: Instruction-wise leakage analysis: LSL/LSR

Verifying independent assumption

Academia

"Well..."



Industry

"But did not the previous study verify this already? "



Let us read pass the "headlines"

"Very high order masking: Efficient implementation and security evaluation"[JS17]:

- TVLA on one specific instance, **NOT** the assumption itself
 - Only 2/4 bits are used
 - Conservative interpretation: assuming $d/2 = 15$ order security
- Fair for their purpose, but should not be taken out of the context

Verifying independent assumption

Academia

"Well..."



Industry

"O.K...then how about the order reduction theorem? "



Let us read pass the "headlines"

"On the cost of lazy engineering for masked software implementations"[BGGRS14]:

- Security reduction for "transition-based leakage"
 - Implicitly assumed shares stores in different registers
- Do not apply at the first place
 - Which has been said in [JS17]

Theorem 1. *An d th-order secure implementation against value-based leakage functions is $\lfloor \frac{d}{2} \rfloor$ th-order secure against transition-based leakage functions.*

Outline

- 1 Intro
- 2 Evaluations in practice
- 3 Read pass the "headlines"...
- 4 Discussion

Our results suggest...

- Independent assumption should not be taken for granted on software platforms

They do not suggest...

- Share-slicing should be forbidden
 - A weaker assumption (say, SNR-based)?
 - Proof does not guarantee everything...
 - Platform-dependent
- Shifter is the (only) source of interaction
 - Various components can contribute
 - Cannot locate the exact source (unless the CPU is completely open-sourced)

What does model assumption mean in practice...

Academia

- Offer schemes in security model



Industry

- Needs the connecting to practice



Who should be the "interpreter"?

Questions?

Thank you!

- NRR06** Nikova, S., Rechberger C., Rijmen V.: Threshold Implementations Against Side-Channel Attacks and Glitches. Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006
- GMK16** Groß, H., Mangard, S., Korak, T.: Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order. Proceedings of the ACM Workshop on Theory of Implementation Security, TIS @ CCS 2016 Vienna, Austria, October, 2016
- Coron14** Coron, J.S.: Higher Order Masking of Look-Up Tables. In Nguyen, P.Q., Oswald, E., eds.: Advances in Cryptology EUROCRYPT 2014

ISW03 Ishai, Y., Sahai, A., Wagner, D.: Private Circuits: Securing Hardware against Probing Attacks. In Boneh, D., ed.: Advances in Cryptology, CRYPTO 2003

BGGRS14 Josep Balasch, Benedikt Gierlichs, Vincent Grosso, Oscar Reparaz, and François-Xavier Standaert. On the cost of lazy engineering for masked software implementations, CARDIS 2014.

BDF+17 Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel implementations of masking schemes and the bounded moment leakage model. In Advances in Cryptology - EUROCRYPT 2017

- JS17** Anthony Journault and François-Xavier Standaert. Very high order masking: Efficient implementation and security evaluation. In Cryptographic Hardware and Embedded Systems - CHES 2017
- GJRS18** Dahmun Goudarzi, Anthony Journault, Matthieu Rivain, and FrançoisXavier Standaert. Secure multiplication for bitslice higher-order masking: Optimisation and comparison. COSADE 2018
- GR17** Dahmun Goudarzi and Matthieu Rivain. How fast can higher-order masking be in software? Advances in Cryptology – EUROCRYPT 2017