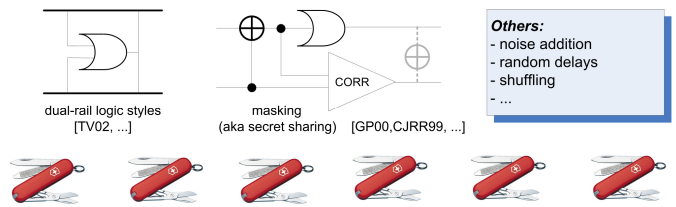
Practical Leakage-Resilience

B. Gierlichs, F.-X. Standaert
KUL COSIC & UCL Crypto Group

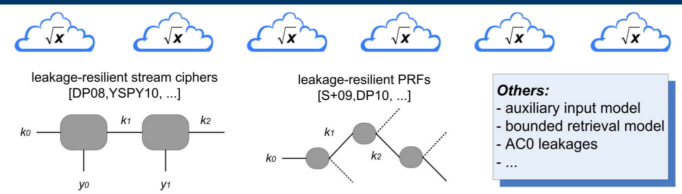
ECRYPT Workshop on Physical Attacks
Graz, Austria, November 2012

ECRYPT II

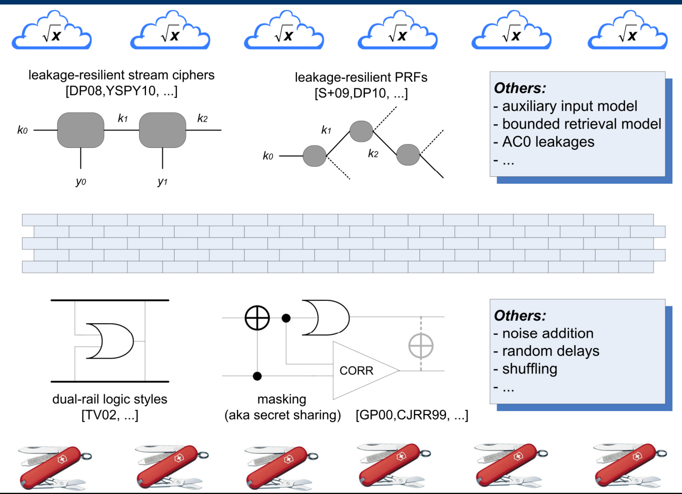
SCA security (implementation level) 1



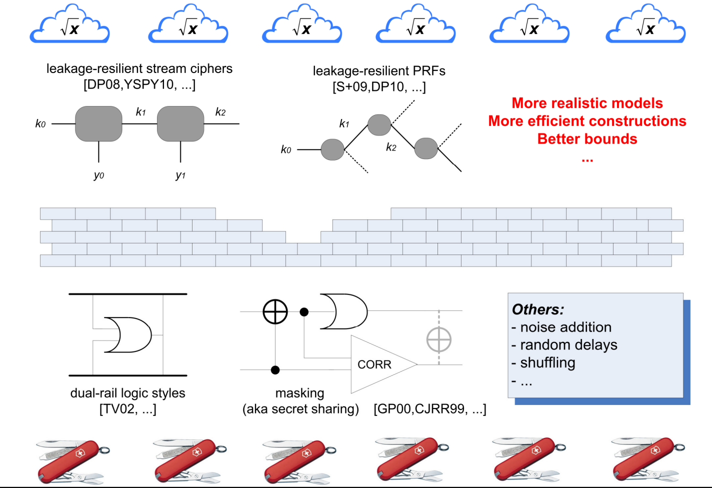
SCA security (mathametical level) 1



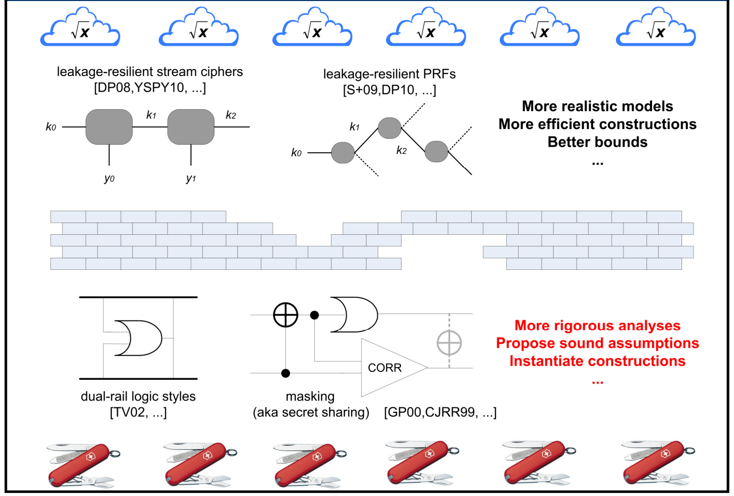
Limitations of current approaches 1



Direction for improvements #1 1



Direction for improvements #2 1



This talk: 2 ECRYPT results in these lines 2

- Leakage-resilient PRFs with parallelism
  - CHES 2012 + new results
    - S. Belaid, F. De Santis, J. Heyszl, A. Joux, S. Mangard, M. Medwed, J. Schmidt, FX Standaert, S. Tillich
- Theory and Practice of a Leakage Resilient Masking Scheme
  - ASIACRYPT 2012
    - J. Balasch, S. Faust, B. Gierlichs, I. Verbauwhede

1. Leakage-Resilient PRFs

Motivation 3

- Why PRFs (not PRPs)?
  - One of the most important primitives in symmetric cryptography (see Goldreich's book)
  - **Enough for encryption** / **authentication**
  - Needed for init. of stream ciphers
  - Stateless primitive!
  - Can be combined with fresh re-keying
  - …

Motivation 3

- Why PRFs (not PRPs)?
  - One of the most important primitives in symmetric cryptography (see Goldreich's book)
  - **Enough for encryption** / **authentication**
  - Needed for init. of stream ciphers
  - Stateless primitive!
  - Can be combined with fresh re-keying
  - …

- Main question: can leakage-resilient PRFs be
  - Secure (**super-exponential security**)?
  - Efficient (compared to other countermeasures)?

Secure – in what sense? 4

- Main focus so far: # of measurements
  - e.g. noise addition: # of measurements increases linearly with the noise variance
  - e.g. masking: # of measurements *may* increase exponentially with the number of masks
    - But requires hardware assumptions (e.g. leakage of shares must be independent)

Secure – in what sense? 4

- Main focus so far: # of measurements
  - e.g. noise addition: # of measurements increases linearly with the noise variance
  - e.g. masking: # of measurements *may* increase exponentially with the number of masks
    - But requires hardware assumptions (e.g. leakage of shares must be independent)

- Leakage-resilient PRFs approach:
  - Bound the data complexity by design
  - Try to guarantee high time complexity

Outline

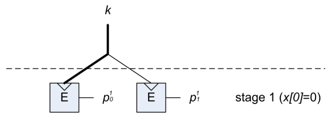
1. Tree-based PRF (GGM 86)
2. Efficiently exploiting parallelism
   a. Previous leakage-resilient PRFs
   b. Our tweak: carefully chosen plaintexts
3. Instantiation issues
   a. Power measurements
   b. Block cipher design
   c. EM radiation

Outline

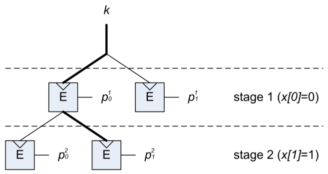
1. Tree-based PRF (GGM 86)
2. Efficiently exploiting parallelism
   a. Previous leakage-resilient PRFs
   b. Our tweak: carefully chosen plaintexts
3. Instantiation issues
   a. Power measurements
   b. Block cipher design
   c. EM radiation

Tree-based PRF (GGM 86) 5

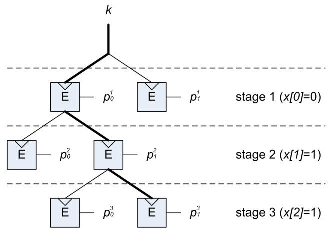
Tree-based PRF (GGM 86) 5



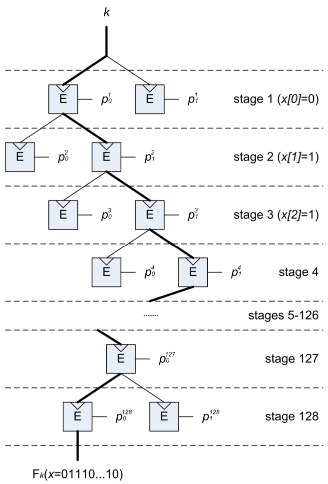
Tree-based PRF (GGM 86) 5



Tree-based PRF (GGM 86) 5



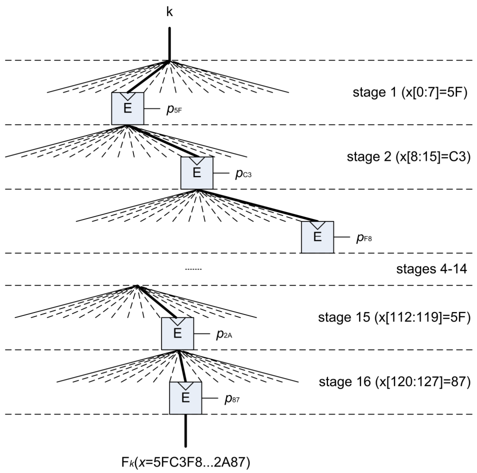
Tree-based PRF (GGM 86) 5



☺: 2-bounded data complexity

☹: 128 AES per 128-bit input

Efficiency / security tradeoff 6



☺: 16 AES per 128-bit input

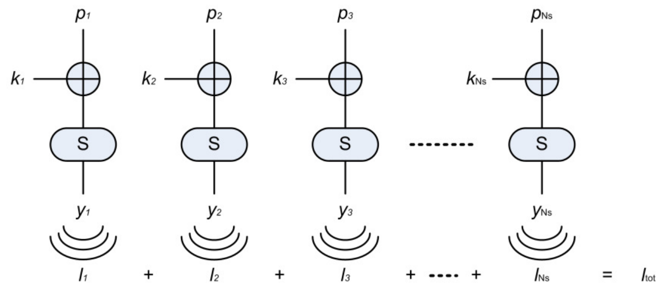
😐: 256-bounded data complexity?

Outline

1. Tree-based PRF (GGM 86)
2. Efficiently exploiting parallelism
   a. Previous leakage-resilient PRFs
   b. Our tweak: carefully chosen plaintexts
3. Instantiation issues
   a. Power measurements
   b. Block cipher design
   c. EM radiation

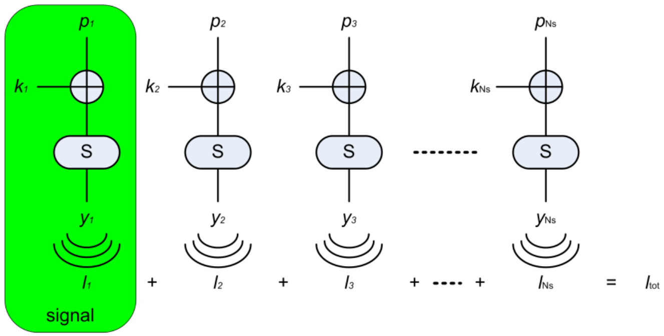
Algorithmic noise (standard DPA)

7



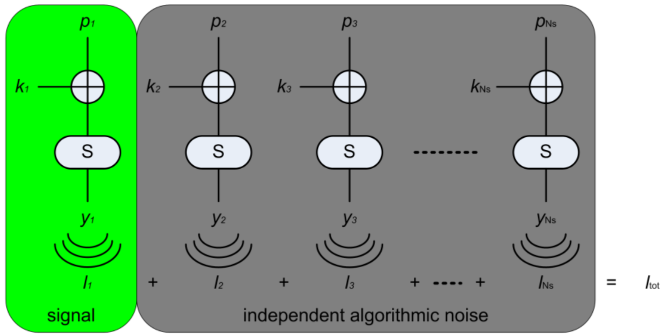
Algorithmic noise (standard DPA)

7

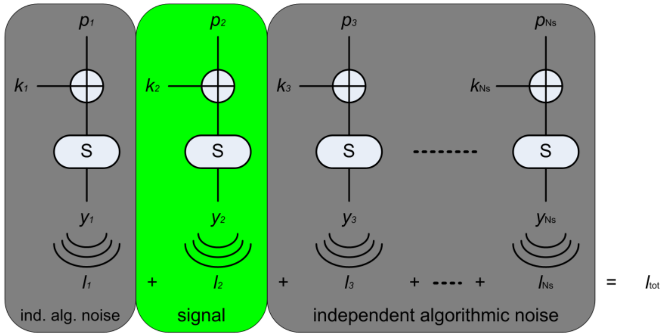


Algorithmic noise (standard DPA)

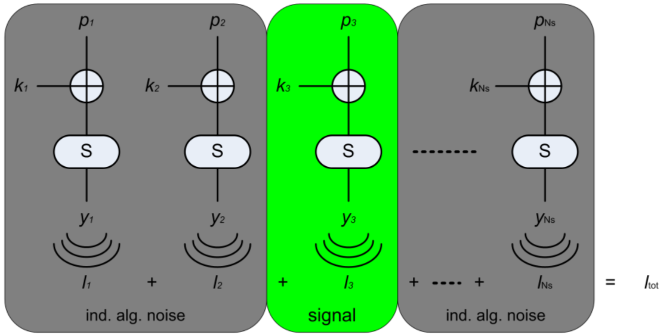
7



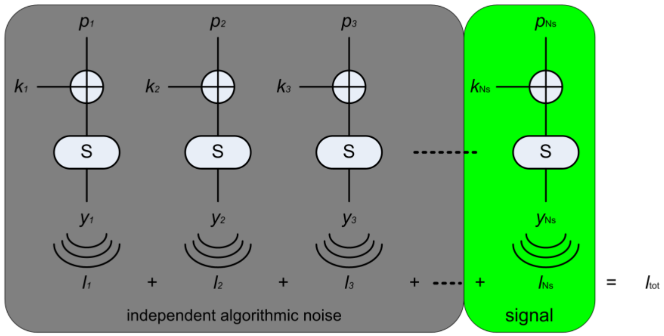
Random p_i's => divide & conquer attacks



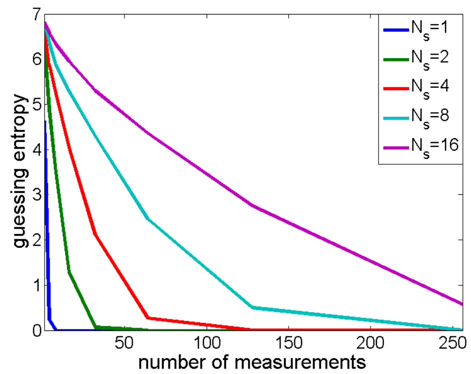
Random p_i's => divide & conquer attacks



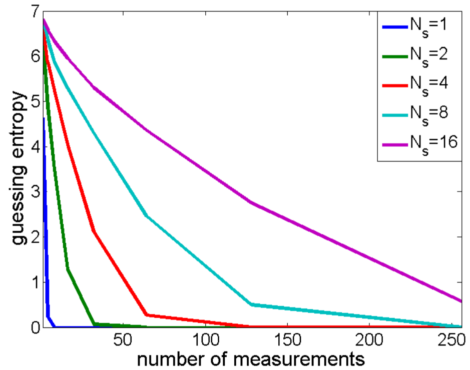
Random p_i's => divide & conquer attacks



Single S-box attack results



Single S-box attack results
9

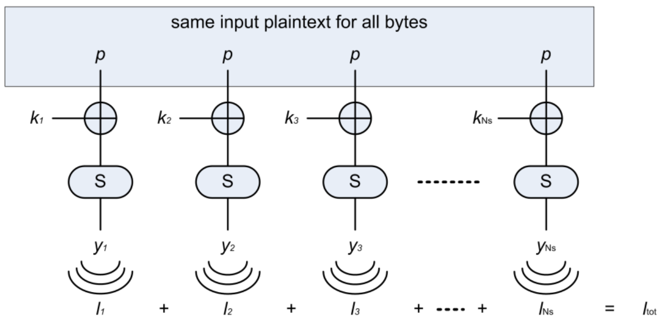


- Noise can be averaged by measuring more ☹

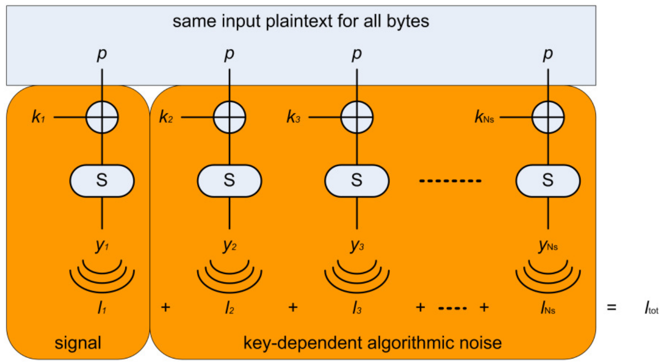
Outline

1. Tree-based PRF (GGM 86)
2. Efficiently exploiting parallelism
   a. Previous leakage-resilient PRFs
   b. Our tweak: carefully chosen plaintexts
3. Instantiation issues
   a. Power measurements
   b. Block cipher design
   c. EM radiation

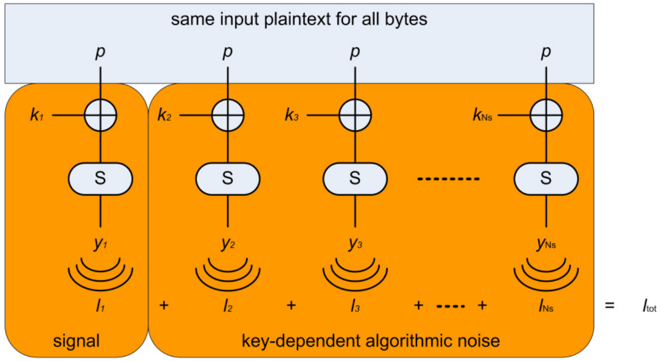
Our tweak: carefully chosen plaintexts (I)
10



Our tweak: carefully chosen plaintexts (I)
10



Our tweak: carefully chosen plaintexts (I) 10



e.g. CPA + HW model: same predictions for 16 key bytes

Our tweak: carefully chosen plaintexts (II) 11

- Intuition #1: algorithmic noise is key dependent
  => Divide & conquer attacks hardly apply

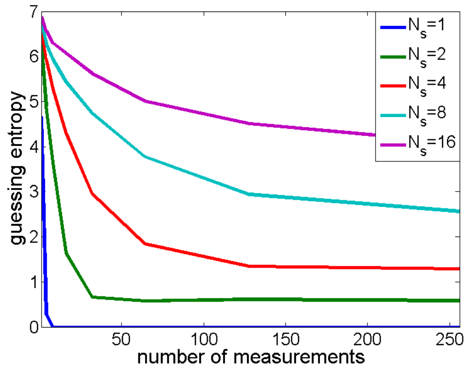
Our tweak: carefully chosen plaintexts (II) 11

- Intuition #1: algorithmic noise is key dependent
  => Divide & conquer attacks hardly apply
- Intuition #2: assume the leakage functions are (roughly) identical for all S-boxes
  - Then the models in standard DPA attacks are also identical for all S-boxes

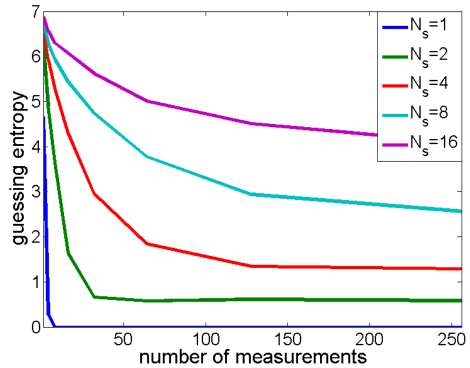
Our tweak: carefully chosen plaintexts (II) 11

- Intuition #1: algorithmic noise is key dependent
  => Divide & conquer attacks hardly apply
- Intuition #2: assume the leakage functions are (roughly) identical for all S-boxes
  - Then the models in standard DPA attacks are also identical for all S-boxes
  - Even in the (unlikely) situation where the *Ns* key bytes are rated in the first *Ns* positions by DPA, it remains to enumerate *Ns*! Permutations
    - e.g. 16!=2^44, 24!=2^79, 32!=2^117

Single S-box attack results 12



Single S-box attack results 12



- Even with 256 meas., noise cannot be averaged ☺

Outline

1. Tree-based PRF (GGM 86)
2. Efficiently exploiting parallelism
   a. Previous leakage-resilient PRFs
   b. Our tweak: carefully chosen plaintexts
3. Instantiation issues
   a. Power measurements
   b. Block cipher design
   c. EM radiation

Main question 13

- Do different S-boxes leak the same?

Main question

13

- Do different S-boxes leak the same?
- FPGA case study with two types of S-boxes

Main question

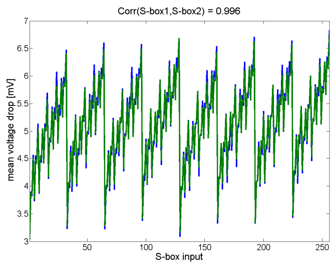
13

- Do different S-boxes leak the same?
- FPGA case study with two types of S-boxes
  - Power measurements

Main question

13

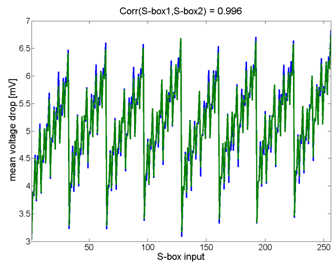
- Do different S-boxes leak the same?
- FPGA case study with two types of S-boxes
  - Power measurements
  - Using the RAM blocks of modern FPGAs

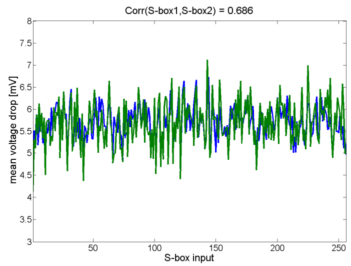


Main question

13

- Do different S-boxes leak the same?
- FPGA case study with two types of S-boxes
  - Power measurements
  - Using the RAM blocks of modern FPGAs
  - Combinatorial (from Canright, CHES 2005)



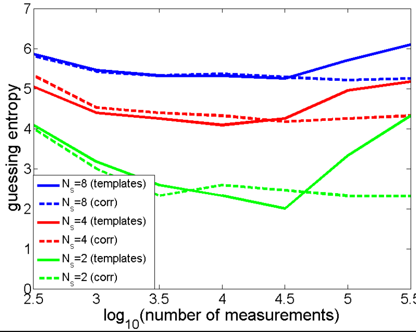


Can we exploit different leakage models? 14

- Case study using the Canright S-boxes
  - Template attacks, correlation attacks
  - Both using the *Ns* different models

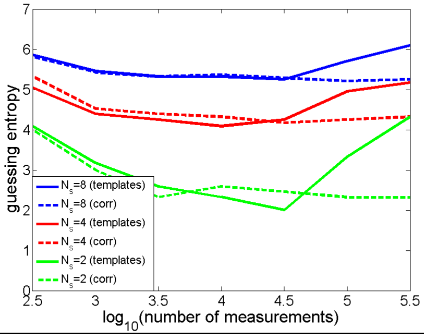
Can we exploit different leakage models? 14

- Case study using the Canright S-boxes
  - Template attacks, correlation attacks
  - Both using the *Ns* different models



Can we exploit different leakage models? 14

- Case study using the Canright S-boxes
  - Template attacks, correlation attacks
  - Both using the *Ns* different models



Main message:
the key-dependent algorithmic noise remains hard to exploit
☺

Outline

1. Tree-based PRF (GGM 86)
2. Efficiently exploiting parallelism
   a. Previous leakage-resilient PRFs
   b. Our tweak: carefully chosen plaintexts
3. Instantiation issues
   a. Power measurements
   b. Block cipher design
   c. EM radiation

Which underlying block cipher? 15

- AES not best suited for LR-PRF designs
  - MixColumn allows “easier” 2nd-round attacks

Which underlying block cipher? 15

- AES not best suited for LR-PRF designs
  - MixColumn allows “easier” 2nd-round attacks

- New candidate: PRESENT-like cipher

Which underlying block cipher? 15

- AES not best suited for LR-PRF designs
  - MixColumn allows “easier” 2nd-round attacks

- New candidate: PRESENT-like cipher

  - With 32 4-bit S-boxes (best tradeoff between time and data complexity of attacks)

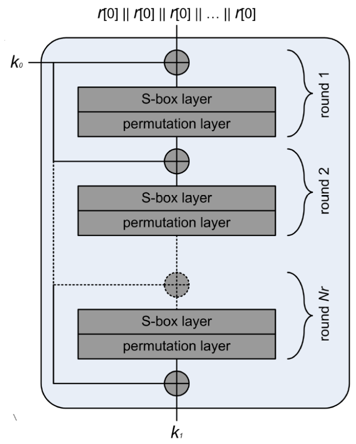
Which underlying block cipher? 15

- AES not best suited for LR-PRF designs
  - MixColumn allows “easier” 2nd-round attacks

- New candidate: PRESENT-like cipher
  - With 32 4-bit S-boxes (best tradeoff between time and data complexity of attacks)

  - Wire crossing with improved “regularity”
    - e.g. the first bits of the S-box outputs should end up in the same position after permutation

Block diagram



- Number of rounds left optional so far
- 5 rounds suggested for fresh re-keying
- How many for secure encryption?

Outline

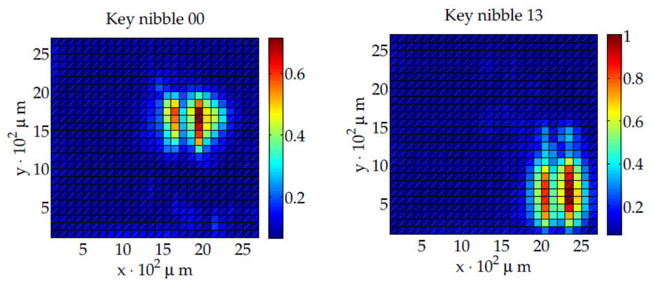
1. Tree-based PRF (GGM 86)
2. Efficiently exploiting parallelism
   a. Previous leakage-resilient PRFs
   b. Our tweak: carefully chosen plaintexts
3. Instantiation issues
   a. Power measurements
   b. Block cipher design
   c. EM radiation

Localization of S-boxes?

- Feasible with (worst-case) profiling
  - Key under control to detect “hot spots”

Localization of S-boxes?

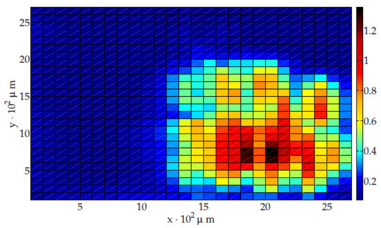
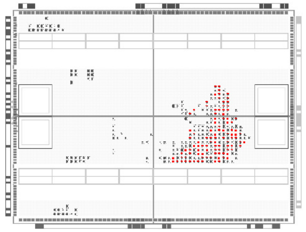
- Feasible with (worst-case) profiling
  - Key under control to detect “hot spots”
  - Leakage models indeed different ☹, e.g.



Leakage exploitation

18

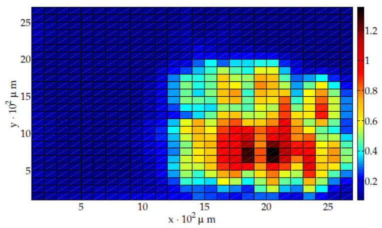
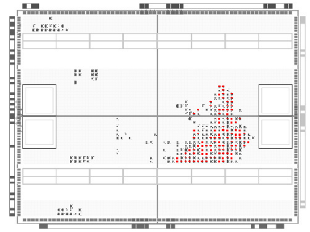
- Putting things together, key-dependent algorithmic noise still more difficult to exploit

Leakage exploitation

18

- Putting things together, key-dependent algorithmic noise still more difficult to exploit

- Current experimental results suggest security bounds around 2^80 time complexity ☺

2. Theory and Practice of a Leakage Resilient Masking Scheme

Motivation

19

Leakage resilient crypto
- Proofs
- Resist "arbitrary" adversaries
- Theoretical
- Strong, abstract requirements for physical behaviour of implementation
- Complex, impractical, large implementation overhead

Masking / blinding
- Proofs
- Resist specific attacks
- Practice oriented
- Concrete requirements for physical behaviour of implementation
- Simple, practical, efficient

Theory and Practice of a Leakage Resilient Masking Scheme

20

- Narrow the gap between theory and practice
- One masking scheme in both worlds
  - Large value of security parameter: leakage resilient
  - Small value of security parameter: feasible on 8-bit microcontroller, secure enough?
- Learn what parts make a scheme inefficient
- What parts are needed only for theoretical security

Inner-product Masking

21

- Secret value X is masked as

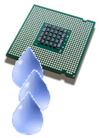
$$X = L_1 \otimes R_1 \oplus \ldots \oplus L_n \otimes R_n$$

- X, L_i, R_i are field elements, $|\mathbb{F}| \geq 2$
- L_i, R_i random, $L_i \neq 0$
- $n \geq 2$ is security parameter
- Focus on GF(2^8) to protect AES
- Closely related to boolean, multiplicative, affine, polynomial masking

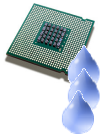
Theory side

22

$$X = L_1 \otimes R_1 \oplus \ldots \oplus L_n \otimes R_n$$



- 2 processors P_R and P_L with independent leakage



- Leakage resilient for n > 130
  - Adversary may learn up to 3n bits from each processor
  - Non-adaptive leakage: adversary may learn L and 3n bits about R
    - Example: adversary may learn L and Hamming weight of each Ri
- Impractical

Theory side

23

$$X = L_1 \otimes R_1 \oplus \ldots \oplus L_n \otimes R_n$$

- Security of operations in masked domain
  - Addition, multiplication, squaring, re-randomization
- Simplified or new, more efficient operations
- Simplified re-randomization
  - Theoretical but not practical attack
  - For proof we assume that it does not leak

Practice side

$$X = L_1 \otimes R_1 \oplus \ldots \oplus L_n \otimes R_n$$

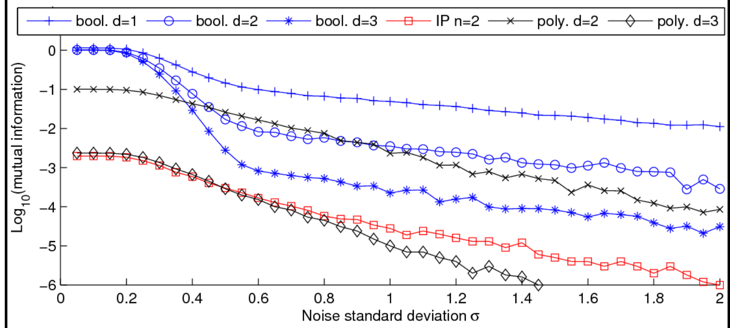
- IP masking with 2n=d+1 is secure against n-1th or (d+1)/2-1th order attacks
  - n = 2 → secure against 1st order attacks
  - 2nd order flaw appears with probability 2^{-8n}
- Complex dependency between shares and secret
- Expect higher security than from Boolean masking with same number of shares

Practice side

- Comparison of information leakage
  - IP masking n=2 (4 shares)
  - Boolean masking (2, 3 and 4 shares)
  - Polynomial masking (4 and 6 shares, including the public constants)
- Simulations
  - Hamming weight leakage of each share
  - Independent Gaussian noise
- Estimate mutual information I(leakages;secret)

Practice side

Comparison of information leakage



Practice side

- Comparison of attack success
  - Multivariate MIA attacks (using HW model)
  - Key recovery: S(p+k) with AES S-box,
  - Leakage simulation as before but no noise
- Estimate number of traces for 90% SR

Masking type	Number of traces
Boolean, 2 shares	90
Boolean, 3 shares	200
Boolean, 4 shares	600
Polynomial, 4 shares	280k
Polynomial, 6 shares	~15M
Inner product, 4 shares	~15M

Practice side 28

- Performance in 8-bit software
- Only one processor: temporal separation
- Masked AES-128 encr in assembly
  - 1536 bytes of LUTs
  - Constant time and flow, no branches
- S-box
  - Compute inverse(x) as x^{254}
  - Affine transform: polynomial over GF(2^8)

$$\mathrm{AffTrans}[X] = \{05\} \otimes X^{128} \oplus \{09\} \otimes X^{64} \oplus \{f9\} \otimes X^{32} \oplus \{25\} \otimes X^{16} \oplus \{f4\} \otimes X^{8} \oplus \{01\} \otimes X^{4} \oplus \{b5\} \otimes X^{2} \oplus \{8f\} \otimes X \oplus \{63\}$$

Practice side 29

- Performance in 8-bit software
  - Including masked key schedule

Operation	Cycle count	
AddRoundKey	8,796	
SubBytes - inverse	45,632	
SubBytes - affine	72,128	117,760
ShiftRows	200	
MixColumns	27,468	
Full AES-128 encr	1,912,000	

- Unprotected AES-128 encr: ~3,000 cycles

Conclusion and future research 30

- Provide input to theory community
  - Implement schemes, identify performance bottlenecks
  - Analyze schemes for security overkill
  - Leakage assumptions that can be practically verified

THANKS