Towards Fresh Re-Keying with Leakage-Resilient PRFs: Cipher Design Principles and Analysis

S. Belaïd[1], **F. De Santis**[2,3], J. Heyszl[4], S. Mangard[3], M. Medwed[5], J.-M. Schmidt[6], F.-X. Standaert[7], S. Tillich[8]

[1] Ecole Normale Supérieure and Thales Communications, France.
[2] Institute for Security in Information Technologies, Technical University of Munich.
[3] Infineon Technologies AG, Neubiberg, Germany.
[4] Fraunhofer Research Institution AISEC, Munich, Germany.
[5] NXP Semiconductors, Graz, Austria.
[6] IAIK, Graz University of Technology, Austria.
[7] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium.
[8] Department of Computer Science, University of Bristol, UK.

24.08.2013
PROOFS Workshop

Outline

Intro

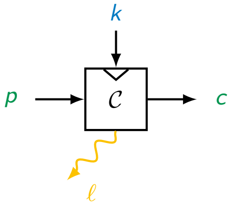
Efficient Leakage-Resilient PRFs

Fresh Re-Keying with Efficient Leakage-Resilient PRFs

Conclusion

Side-Channel Information Leakage

- Cryptographic implementations leak information over side-channels

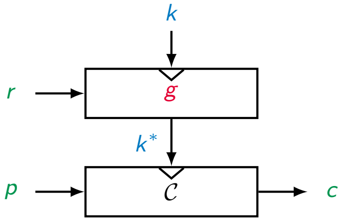


- Implementation countermeasures:
  - ➥ Protected logic styles, masking schemes, re-keying schemes, ...

- Focus on: re-keying schemes for symmetric cryptography

Re-Keying Schemes [AB00, MSGR10]

- The success probability of many (physical) attacks depends on the amount of cryptographic operations which are observable under the same key
- Idea: generate fresh keys from a master key using a re-keying function g



- Requirements:
  - ➥ g is DPA/SPA secure
  - ➥ $\mathcal{C}$ is SPA secure
  - ➥ r is a public *random* nonce

Re-keying Functions

Re-keying functions in the literature:

- Modular multiplication [MSGR10]

$$g\colon (\mathrm{GF}(2^8)[x]/(x^d+1))^2 \to \mathrm{GF}(2^8)[x]/(x^d+1)\colon (k,r) \to k\cdot r$$

Our proposal:

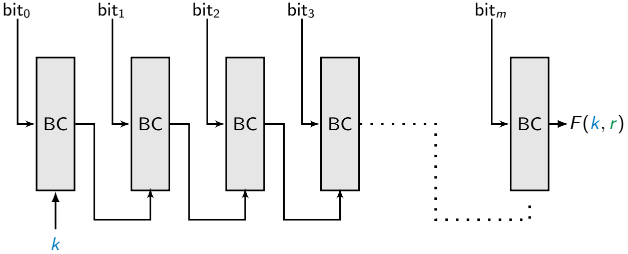
- Leakage resilient pseudo-random function [SPY$^+$09]

Informally:

- A pseudo-random function (PRF) is a function which is computationally indistinguishable from a *truly* random function
- A leakage resilient pseudo-random function (LRPRF) is a PRF which preserves "some" security, even in presence of leakages

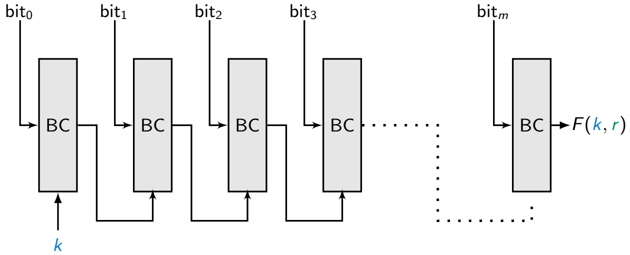
Instantiating Block Cipher based PRFs

From *classical* construction [GGM86], $r = \text{bit}_0 \| \text{bit}_1 \| \text{bit}_2 \|, \text{bit}_3 \| \ldots \| \text{bit}_m$

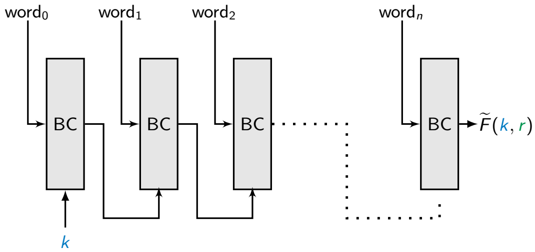


Instantiating Block Cipher based PRFs

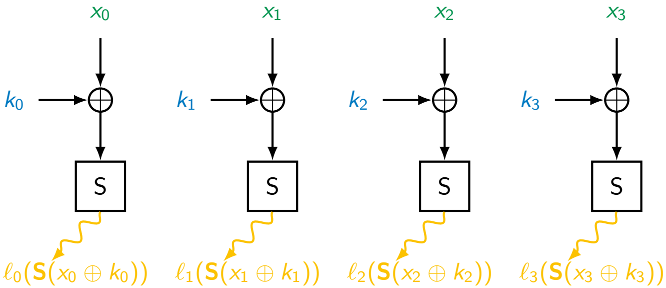
From *classical* construction [GGM86], r=$\text{bit}_0\|\text{bit}_1\|\text{bit}_2\|,\text{bit}_3\|\ldots\|\text{bit}_m$



From *efficient* construction [SPY⁺09], r=$\text{word}_0\|\text{word}_1\|\text{word}_2\|\ldots\|\text{word}_n$

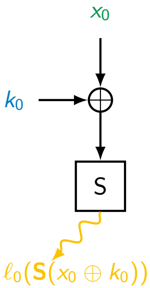


Classical DPA Attack Scenario



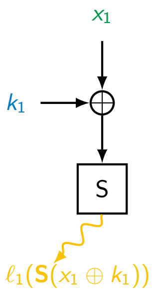
Divide et Impera: attack each S-box output independently

Classical DPA Attack Scenario



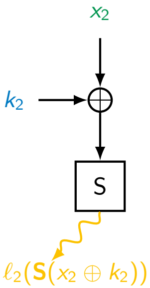
Divide et Impera: attack first S-box output

Classical DPA Attack Scenario



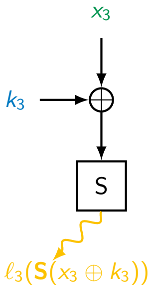
Divide et Impera: attack second S-box output

Classical DPA Attack Scenario



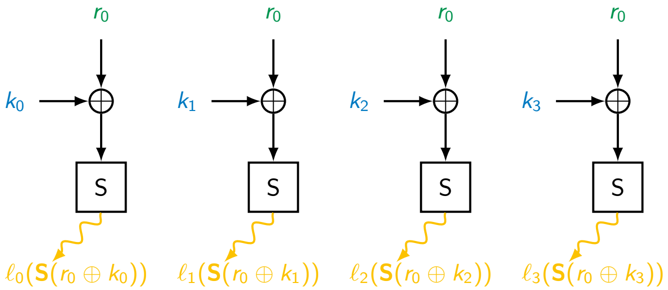
Divide et Impera: attack third S-box output

Classical DPA Attack Scenario



Divide et Impera: attack fourth S-box output ...

BC-based PRF DPA Attack Scenario [MSJ12]



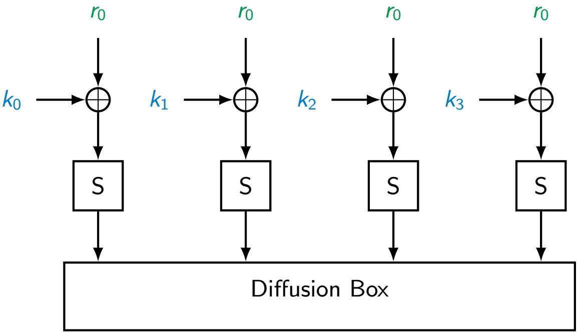
- The implementation is parallel
- The leakage functions ℓ_i are all equal
- The subkey words k_i are successfully recovered

$\Rightarrow$ Still there is a super-exponential time complexity of an enumeration over N_s to recover the full key, in case of AES: $16! = 2^{44}$ time complexity

Contributions

1. Which block cipher best suits a leakage resilient PRF in hardware?
2. Which performance can be achieved for re-keying applications?
3. Is it possible to mount classical DPA attacks in a localized EM setting?

Efficient Leakage-Resilient PRFs: Block Cipher Design Principles



SP-networks:

1. Define the round structure
2. Define the key schedule

Efficient Leakage-Resilient PRFs: Block Cipher Design Principles

- Design Parameter: number of S-boxes N_s and S-box size b
- Design Criteria: best security vs performance trade-off

N_s	16	32
$b = 4$	2^{39}	2^{95}
$b = 8$	2^{44}	2^{116}

Table: Time complexity in the 1^{st} round

N_s	16	32
$b = 4$	$2^{13.4}$	$2^{15.5}$
$b = 8$	$2^{28.8}$	$2^{38.1}$

Table: Time complexity in the 2^{nd} round

N_s	16	32
$b = 4$	432	1051
$b = 8$	1060	2954

Table: # Tr. CPA VS data complexity

N_s	16	32
$b = 4$	64	128
$b = 8$	128	256

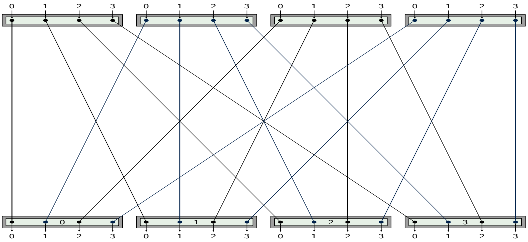
Table: Datapath size $N_s b$

$\Rightarrow$ Our Choice: 4-bit PRESENT S-box with $N_s = 32$

Efficient Leakage-Resilient PRFs: Block Cipher Design Principles

- Design Parameter: Diffusion layer
- Design Criteria: Efficient in hardware and not leaking intermediate values

First option: SMALL-PRESENT pLayer

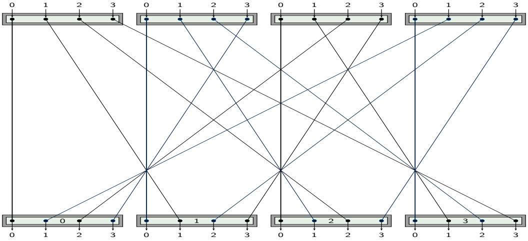


Issue: HD leaks the relative position of nibbles ...

Efficient Leakage-Resilient PRFs: Block Cipher Design Principles

- Design Parameter: Diffusion layer
- Design Criteria: Efficient in hardware and not leaking intermediate values

Our proposal: SINGLE-PATTERN



The relative offset of inputs bits must be preserved after the permutation

⇒ Our Choice: SINGLE-PATTERN

Efficient Leakage-Resilient PRFs: Block Cipher Design Principles

- Design Parameter: Number of rounds
- Design Criteria: Full diffusion (minimum property for re-keying)

- ≥ 3 rounds for $N_s = 32, b = 4$

⇒ Our Choice: 5 rounds

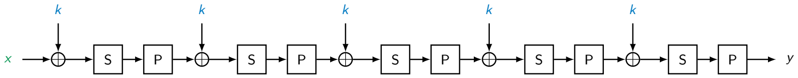
- Design Parameter: Key schedule
- Design Criteria: Efficient and not leaking intermediate values

⇒ Our Choice: No key schedule, simple key addition

Efficient Leakage-Resilient PRFs: Block Cipher Design Principles

To sum up:

- S-box layer: 32 × 4-bit PRESENT S-boxes
- Diffusion layer: SINGLE-PATTERN wire crossing with improved "regularity"
- Key schedule: Simple key addition as for the LED block cipher
- Number of rounds: 5
- Iterations: 32 for 128-bit nonces



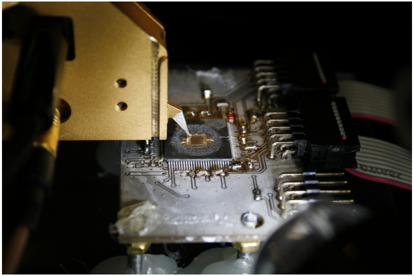
Note: intended for re-keying application only !

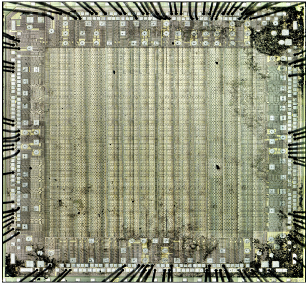
Fresh Re-Keying with Efficient Leakage-Resilient PRFs: Implementation Results

g	BC	Area [kGE]	Latency [Clock Cycles]
[MSGR10]	8-bit AES [FWR05]	10.7	562
Our PRF	8-bit AES [HAHH06]	7.19	324
	Threshold AES [MPL+11]	10.8	266
Our PRF	PRESENT(ser) [RPLP08]	4.09	643
Our PRF	PRESENT(par) [RPLP08]	4.47	131
	Threshold PRESENT [PMK+11]	3.59	578

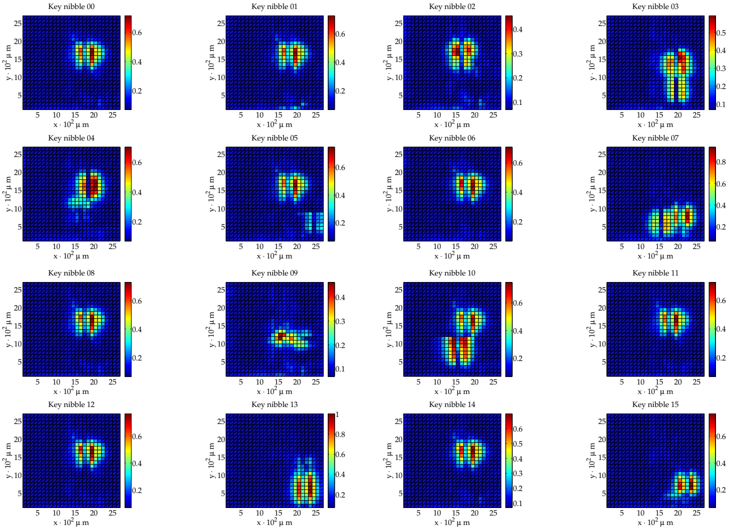
Fresh Re-Keying with Efficient Leakage-Resilient PRFs: Localized EM Attacks

- Analysis conducted on a depackaged (VQ100) Xilinx Spartan FPGA 3
- EM activity measured on the frontside
- Univariate profiled CPA attacks

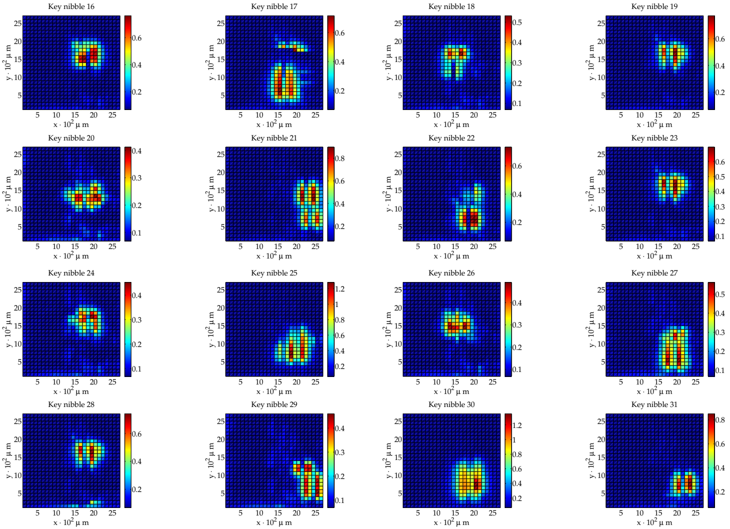




Fresh Re-Keying with Efficient Leakage-Resilient PRFs: Localized EM Attacks

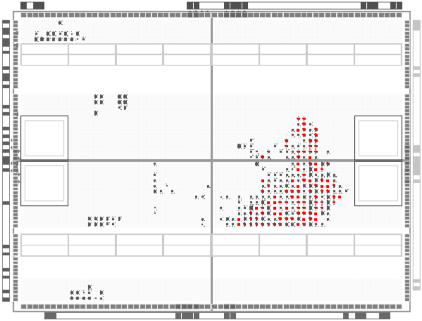


Fresh Re-Keying with Efficient Leakage-Resilient PRFs: Localized EM Attacks



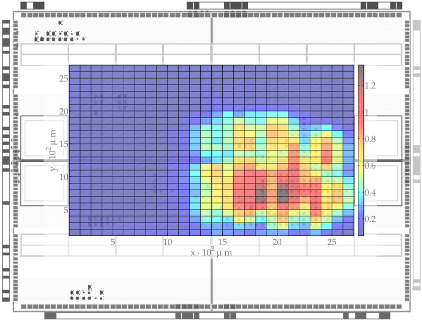
Fresh Re-Keying with Efficient Leakage-Resilient PRFs: Localized EM Attacks

- An optimal key enumeration algorithm [VCGRS13] was used to evaluate the remaining time complexity after localized EM attacks
- Yet experimental results suggest security bounds $> 2^{80}$ time complexity



Fresh Re-Keying with Efficient Leakage-Resilient PRFs:
Localized EM Attacks

- An optimal key enumeration algorithm [VCGRS13] was used to evaluate the remaining time complexity after localized EM attacks
- Yet experimental results suggest security bounds $> 2^{80}$ time complexity



Conclusion

1. We provided block cipher design principles to best suit an efficient leakage-resilient PRF in hardware
   - ➥ Security should be considered at **all** abstraction levels

2. We showed that efficient leakage resilient PRFs are valid alternatives for fresh re-keying in hardware

3. We showed that the key-dependent algorithmic noise is still hard to exploit, even in a localized EM setting (univariate)

Future work:

- Full specification of our BC-like proposal
- Multivariate attacks
- Randomization countermeasure to thwart localized EM attacks

References I

Michel Abdalla and Mihir Bellare.
Increasing the lifetime of a key: A comparative analysis of the security of re-keying techniques.
In *Advances in Cryptology*, ASIACRYPT '00, pages 546–559, London, UK, UK, 2000. Springer-Verlag.

Martin Feldhofer, Johannes Wolkerstorfer, and Vincent Rijmen.
Aes implementation on a grain of sand.
Information Security, IEE Proceedings, 152:13 – 20, 2005.

Oded Goldreich, Shafi Goldwasser, and Silvio Micali.
How to construct random functions.
J. ACM, 33(4):792–807, August 1986.

P. Hamalainen, T. Alho, M. Hannikainen, and T.D. Hamalainen.
Design and implementation of low-area and low-power aes encryption hardware core.
In *DSD 2006. 9th EUROMICRO Conference*, pages 577–583, 2006.

Amir Moradi, Axel Poschmann, San Ling, Christof Paar, and Huaxiong Wang.
Pushing the limits: A very compact and a threshold implementation of aes.
In KennethG. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 69–88. Springer Berlin Heidelberg, 2011.

Marcel Medwed, François-Xavier Standaert, Johann Großschädl, and Francesco Regazzoni.
Fresh re-keying: Security against side-channel and fault attacks for low-cost devices.
In Daniel J. Bernstein and Tanja Lange, editors, *AFRICACRYPT*, volume 6055 of *LNCS*, pages 279–296. Springer, 2010.

Marcel Medwed, François-Xavier Standaert, and Antoine Joux.
Towards super-exponential side-channel security with efficient leakage-resilient prfs.
In Emmanuel Prouff and Patrick Schaumont, editors, *CHES 2012*, volume 7428 of *LNCS*, pages 193–212. Springer Berlin Heidelberg, 2012.

Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling.
Side-channel resistant crypto for less than 2, 300 ge.
J. Cryptology, 24(2):322–345, 2011.

References II

Carsten Rolfes, Axel Poschmann, Gregor Leander, and Christof Paar.
Ultra-lightweight implementations for smart devices - security for 1000 gate equivalents.
In Gilles Grimaud and François-Xavier Standaert, editors, *CARDIS*, volume 5189 of *LNCS*, pages 89–103. Springer, 2008.

François-Xavier Standaert, Olivier Pereira, Yu Yu, Jean-Jacques Quisquater, Moti Yung, and Elisabeth Oswald.
Leakage resilient cryptography in practice.
IACR Cryptology ePrint Archive, 2009:341, 2009.

Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert.
An optimal key enumeration algorithm and its application to side-channel attacks.
In Lars R. Knudsen and Huapeng Wu, editors, *SAC*, volume 7707 of *LNCS*, pages 390–406. Springer Berlin Heidelberg, 2013.