On Implementation Security and ISAP v2.0

Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas and Thomas Unterluggauer

NIST Lightweight Cryptography Workshop 2020

Motivation

- Prime applications of LWC are:
  - Constraint devices (low computing power and memory)
  - Industry use cases (require protection from physical attacks)
- Many NIST LWC submissions use lightweight building blocks
  - Reduce cost of masking
- But the story doesn't quite end here . . .

Talk Outline

- Passive Implementation Attacks
  - DPA, **SPA**
- Fast & Compact Co-Processor for Ascon and Isap
  - Decent implementation security from unprotected building blocks
- **Active Implementation Attacks**
  - DFA, SFA, SIFA
- Summary

Passive Implementation Attacks

Differential Power Analysis (DPA)

- Requires varying inputs or outputs [KJJ99]
- Exploits differences in power consumption
- Countermeasures:
  - Implementation-level: (Higher-order) masking [GP99]
  - Mode-level: GGM-like constructions [GGM84]

Plaintext Recovery Attacks

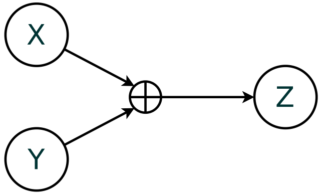
- DPA attack on plaintext during authenticated decryption
  - Without knowledge of the key
- Keys are not the only asset $\rightarrow$ plaintext could contain:
  - Keys
  - Firmware
  - Intellectual Property
- Countermeasures:
  - Implementation-level: Masking (careful with leveled implementations)
  - Mode-level: Verify authenticity of ciphertext/nonce before decryption

Simple Power Analysis (SPA)

- Similar: Profiled Attacks, Template Attacks
- Can be super powerful!
- How they work:
  1. Characterize target device $\rightarrow$ build templates
  2. Look at few/many different intermediate steps of a computation
  3. Match templates and get noisy leakages
  4. Combine leakage using multivariate PDFs/belief propagation

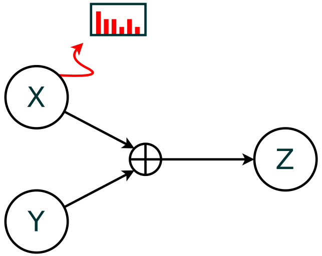
Belief Propagation

- Two types of nodes
  - Variable nodes
  - Factor nodes (here: XOR)



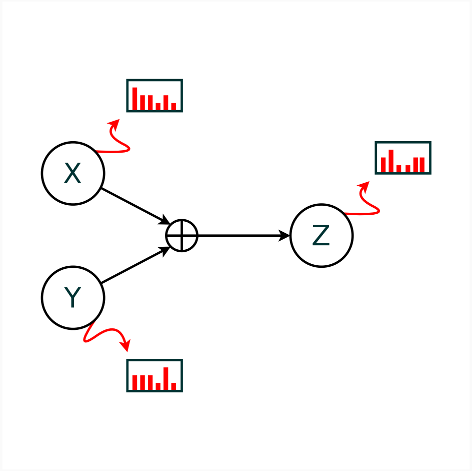
Belief Propagation

- Two types of nodes
  - Variable nodes
  - Factor nodes (here: XOR)



Belief Propagation

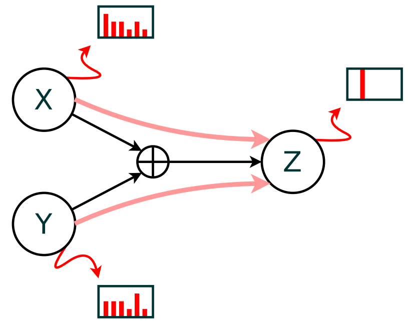
- Two types of nodes
  - Variable nodes
  - Factor nodes (here: XOR)



Belief Propagation

- Two types of nodes
  - Variable nodes
  - Factor nodes (here: XOR)
- Passes information between both types of nodes

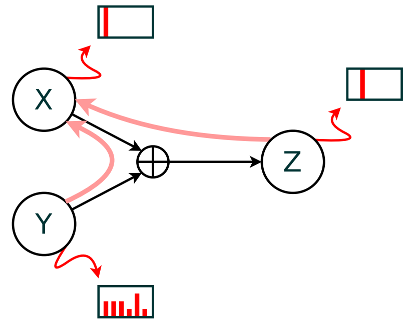
$\Rightarrow$ Combine leakage information



Belief Propagation

- Two types of nodes
  - Variable nodes
  - Factor nodes (here: XOR)
- Passes information between both types of nodes

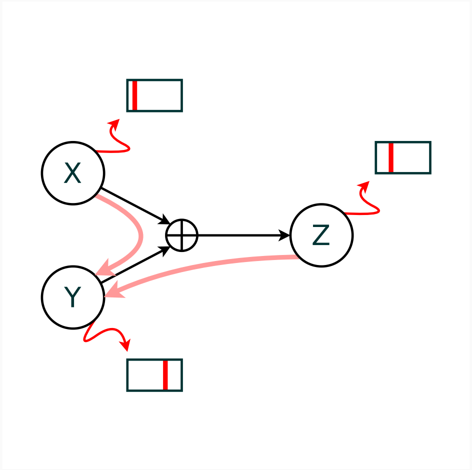
⇒ Combine leakage information



Belief Propagation

- Two types of nodes
  - Variable nodes
  - Factor nodes (here: XOR)
- Passes information between both types of nodes

$\Rightarrow$ Combine leakage information



Single-Trace Attacks on Keccak

- Presented at CHES 2020 [KPP20]
- Accumulate and combine **hundreds/thousands** of leakage points
  - State recoveries possible for 8, 16, (32)-bit devices
- Similar results in other publications [VGS14; PP19; Bel+20]
  - Also work against masked implementations
- Downside?
  - Attacker requires precise knowledge about algorithm execution

Countermeasures against SPA

- Masking alone is not very effective:
  - SPA attacks on 1st-order masking [OM07; HTM09]
  - SPA attacks on higher-order masking [PPM17]
- Hiding is more effective:
  - Uncertainty about when computations occur (belief propagation [Rav+20])
  - Increased SNR on low-end devices

Fast&Compact Co-Processor for Ascon and ISAP

Fast&Compact Co-Processor

- To be presented at CARDIS 2020 [SP20]
- Tight coupling of ASCON-*p* to processors register file
  - No additional negisters needed
  - No interface needed
  - ⇒ Half the area of dedicated co-processor designs
- Mode remains entirely in software → flexible:
  - ASCON, ASCON-HASH, ASCON-XOF
  - ISAP-A-128A, ISAP-A-128
- Speed-ups by a factor of about 50 to 80

Fast&Compact Co-Processor - Performance

Table 1: Runtime and code size comparison of ASCON and ISAP, with/without 1-round ASCON-*p* hardware acceleration (`HW-A`)

Implementations	Cycles/Byte			Binary Size (B)
	64 B	1536 B	long	
`Ascon-C (-O3)`	164.3	110.6	108.3	11 716
`Ascon-C (-Os)`	269.7	187.1	183.5	2 104
`Ascon-ASM + HW-A`	4.2	2.2	2.1	888
`AsconHash-ASM + HW-A`	4.6	2.6	2.5	484
`ISAP-A-128a-C (-O3)`	1 184.3	386.9	352.3	11 052
`ISAP-A-128a-ASM + HW-A`	29.1	5.2	4.2	1 844

Fast&Compact Co-Processor - Area

Table 2: Area comparison of the RISC-V RI5CY core and various co-processor designs

Design	kGE	
	Standalone	Integration
RI5CY base design	45.6	-
This work	4.2	0.5
ASCON co-processor [Gro+15]	7.1	?
ASCON co-processor [Gro]	9.4	?
ISAP co-processor (estimated) [Dob+19]	$\leq$ 12.8	?

Fast&Compact Co-Processor

- With respect to passive attacks, ISAP provides protection against ...
    - DPA protection (incl. plaintext recovery)
- Co-processor provides basic protection against SPA attacks ...
    - Attacker is essentially forced to perform localized EM analysis
    - Simple steps to further improve SPA protection are discussed [SP20]
- ISAP also offers protection against active attacks ...

Active Implementation Attacks

Differential Fault Attacks (DFA)

- Requires constant (unknown) inputs
- Exploits differences between correct/faulty computations [Bar+06]
- Generally applicable to AEAD if either:
  - Nonce is repeated (Enc)
  - Unverified plaintext is released (Dec)
  - Tag verification before decryption (Dec)
- Countermeasures:
  - Implementation-level: Redundant computation (temporal/spatial)
  - Mode-level protection possible

Statistical Fault Analysis (SFA)

- Exploits faulty computations only [Fuh+13]
- Requires varying (unknown) inputs
- Applicable to AEAD if:
  - Key addition occurs before output (e.g. tag or ciphertext)
- Countermeasures:
  - Implementation-level: Redundant computation (temporal/spatial)
  - Mode-level protection possible

Statistical Ineffective Fault Attacks (SIFA)

- Exploits faulted but correct computations only [Dob+18]
- Requires varying (known[1]) inputs
- Countermeasures:
    - Implementation-level: Specific combination of redundancy and masking [Dae+20]
    - Mode-level protection possible

[1]When considering attacks on AEAD.

Summary

In a Nutshell...

Dealing with typical implementation attacks purely on algorithmic level:

Attack Type	**Primary Countermeasure**	**Comment**
DPA	Masking	Low-end: Higher-order
SPA	Hiding	Shuffling, Dummy Ops.
DFA/SFA	Redundancy	Spatial/Temporal
SIFA	(Masking + Redundancy)	[Dae+20]

Least common multiple: Hiding + Masking + Redundancy

(Remark: Masking + Redundancy alone is not very effective against SPA [OM07; HTM09; PPM17])

In a Nutshell...

What can be done on mode-level (ISAP):

Attack Type	Primary Countermeasure	Comment
DPA	Masking	Low-end: Higher-order
SPA	Hiding	Shuffling, Dummy Ops.
DFA/SFA	Redundancy	Spatial/Temporal
SIFA	(Masking + Redundancy)	[Dae+20]

Least common multiple: Hiding

(Remark: Shuffling + “cheap” masking can also give an effective hiding scheme)

Final Remarks

- ISAP is fast and compact in applications that require implementation security
- Goes well with other schemes utilizing KECCAK-p[400] or ASCON-p
  - Fast AEAD without implementation security
  - Hashing functionality
  - (See e.g. compact co-processor design)

Thank you!

References

[Bar+06] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan. The Sorcerer's Apprentice Guide to Fault Attacks. In: Proc. IEEE 94.2 (2006), pp. 370–382.

[Bel+20] D. Bellizia, O. Bronchain, G. Cassiers, V. Grosso, C. Guo, C. Momin, O. Pereira, T. Peters, and F. Standaert. Mode-Level vs. Implementation-Level Physical Security in Symmetric Cryptography - A Practical Guide Through the Leakage-Resistance Jungle. In: CRYPTO (1). Vol. 12170. Lecture Notes in Computer Science. Springer, 2020, pp. 369–400.

[Dae+20] J. Daemen, C. Dobraunig, M. Eichlseder, H. Groß, F. Mendel, and R. Primas. Protecting against Statistical Ineffective Fault Attacks. In: IACR Trans. Cryptogr. Hardw. Embed. Syst. 2020.3 (2020), pp. 508–543.

[Dob+18] C. Dobraunig, M. Eichlseder, H. Groß, S. Mangard, F. Mendel, and R. Primas. Statistical Ineffective Fault Attacks on Masked AES with Fault Countermeasures. In: ASIACRYPT (2). Vol. 11273. Lecture Notes in Computer Science. Springer, 2018, pp. 315–342.

References

[Dob+19] C. Dobraunig, M. Eichlseder, S. Mangard, F. Mendel, B. Mennink, R. Primas, and T. Unterluggauer. ISAP v2.0. Submission to the NIST Lightweight Crypto Competition. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/isap-spec-round2.pdf. 2019.

[Fuh+13] T. Fuhr, É. Jaulmes, V. Lomné, and A. Thillard. Fault Attacks on AES with Faulty Ciphertexts Only. In: FDTC. IEEE Computer Society, 2013, pp. 108–118.

[GGM84] O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions (Extended Abstract). In: FOCS. IEEE Computer Society, 1984, pp. 464–479.

[GP99] L. Goubin and J. Patarin. DES and Differential Power Analysis (The "Duplication" Method). In: CHES. Vol. 1717. Lecture Notes in Computer Science. Springer, 1999, pp. 158–172.

[Gro] H. Groß. CAESAR Hacrdware API reference implementation. https://github.com/IAIK/ascon_hardware/tree/master/caesar_hardware_api_v_1_0_3/ASCON_ASCON (accessed 12/2019). (Visited on 12/2019).

[Gro+15] H. Groß, E. Wenger, C. Dobraunig, and C. Ehrenhöfer. Suit up! - Made-to-Measure Hardware Implementations of ASCON. In: DSD. IEEE Computer Society, 2015, pp. 645–652.

References

[HTM09] N. Hanley, M. Tunstall, and W. P. Marnane. Unknown Plaintext Template Attacks. In: WISA. Vol. 5932. Lecture Notes in Computer Science. Springer, 2009, pp. 148–162.

[KJJ99] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In: CRYPTO. Vol. 1666. Lecture Notes in Computer Science. Springer, 1999, pp. 388–397.

[KPP20] M. J. Kannwischer, P. Pessl, and R. Primas. Single-Trace Attacks on Keccak. In: IACR Trans. Cryptogr. Hardw. Embed. Syst. 2020.3 (2020), pp. 243–268.

[OM07] E. Oswald and S. Mangard. Template Attacks on Masking - Resistance Is Futile. In: CT-RSA. Vol. 4377. Lecture Notes in Computer Science. Springer, 2007, pp. 243–256.

[PP19] P. Pessl and R. Primas. More Practical Single-Trace Attacks on the Number Theoretic Transform. In: LATINCRYPT. Vol. 11774. Lecture Notes in Computer Science. Springer, 2019, pp. 130–149.

[PPM17] R. Primas, P. Pessl, and S. Mangard. Single-Trace Side-Channel Attacks on Masked Lattice-Based Encryption. In: CHES. Vol. 10529. Lecture Notes in Computer Science. Springer, 2017, pp. 513–533.

References

[Rav+20] P. Ravi, R. Poussier, S. Bhasin, and A. Chattopadhyay. On Configurable SCA Countermeasures Against Single Trace Attacks for the NTT - A Performance Evaluation Study over Kyber and Dilithium on the ARM Cortex-M4. In: IACR Cryptol. ePrint Arch. 2020 (2020), p. 1038.

[SP20] S. Steinegger and R. Primas. A Fast and Compact Accelerator for Ascon and Friends. In: IACR Cryptol. ePrint Arch. 2020 (2020), p. 1083.

[VGS14] N. Veyrat-Charvillon, B. Gérard, and F. Standaert. Soft Analytical Side-Channel Attacks. In: ASIACRYPT (1). Vol. 8873. Lecture Notes in Computer Science. Springer, 2014, pp. 282–296.