Physical security for cryptographic implementations with open hardware

Gaëtan Cassiers

July 11, 2023

Cryptography example: Symmetric encryption



Kerckhoff's principles:

- Enc and Dec algorithm can be public.
- Only key needs to be secret.

Side-channel leakage





- Power leakage (supply current)
- Electromagnetic leakage (near-field EM radiation)

. . .



- Add round key
- Sbox (non-linear)
- Linear mixing



- Add round key
- Sbox (non-linear)
- Linear mixing



- Add round key
- Sbox (non-linear)
- Linear mixing



- Add round key
- Sbox (non-linear)
- Linear mixing



Round:

- Add round key
- Sbox (non-linear)
- Linear mixing

10 rounds



Round:

- Add round key
- Sbox (non-linear)
- Linear mixing

Intermediate values reveal the key.

Side-channel attack on AES



DPA attack:

- Collect leakage for multiple plaintexts (same key).
- Enumerate over the 256 possible values for $k_{0,0}$.
- Classify leakage traces in two groups according the to LSB of SubBytes(k_{0,0} ⊕ a_{0,0}).
- The correct key maximizes the difference between the averages of the groups.

\rightarrow Divide-and-conquer attack

Figure of merit:

- ▶ Number of executions with leakage ("traces") needed for key recovery
- Unprotected implementation: typ. 1 to 1000.

Countermeasures

- Decrease SNR (noise generation circuits, clock jitter, dual-rail logic)
- Hide/Mask intermediate values E.g., Boolean masking: $x \to (x_0, x_1)$
 - x_0 random, $x_1 = x \oplus x_0$
 - need to adapt all computations
- Rekeying: reduce the lifetime of a single key
 - Cryptographic security proofs

- ► Released May 2023
- Beyond typical reasearch-quality: implementation, documentation, verification
- ► Dual-licensing scheme (OHL-S+commercial), eventually permissive
- Open to comments, feedback, contributions

- Arbitrary-order masking: $x = x_0 \oplus x_1 \oplus \cdots \oplus x_d$.
- Provable security properties: no bad surprise.
- ▶ Robustly secure, portable and efficient synthesis: still an open problem
- Under public evaluation with a public dataset of leakage traces (first-order, FPGA).
 - Current best attack: 390,000 traces.

Unusual (?) requirements:

- Standard cell designs
- Modern tools optimizations
- E.g., masking constraints:
 - Glitches matter.
 - Prevent retiming of some Flip-flops.
 - Set of input wires in a combinational circuit.
 - Monotonic logic.
- Custom design generation and verification steps.

As a generic IP designer: do this in a robust and portable manner!

Multiple tools to verify the security of implementations (e.g., masking).

Needs:

- Symbolic evaluation \rightarrow (abstract) netlist
- Simulation
- Additional annotations (e.g. verilog attributes)

An additional design flow step:

- Should be integrated in design flow.
- ► Ideally, run on final netlist (& also at earlier stages).
- Currently: mostly separate flows, using custom annotation schemes. Brittle.

Open vs closed countermeasures

Evaluating security is easier when the design is known.



Open hardware security: a timely proposal?

As research advances:

- the advantages of security by obscurity vanish,
- open solutions imporove.



Physical security for cryptographic implementations with open hardware

SIMPLE-Crypto association

Improving the long-term security of crypto. implementations

- Developing and maintaining open soure HW & SW
- Ensuring continuous security evaluation of the designs
- Trainings on physical security with open designs
- Complementing the existing industrial ecosystem
 - Design companies: Open-source & proprietary chips
 - Evaluation labs: continous assessment of open-source specific IP blocks
 - Standardization: maintain high-quality reference implementations with open evidence of good security.
- Collaborating with academia
 - Support developing research prototypes into reusable open-source blocks.

- Open-hardware improves security
- Open toolchain helps to build secure hardware
- Looking for feedback, collaborations...

https://simple-crypto.org