When the Decoder Has to Look Twice: Clock-Glitching a PUF Error Correction

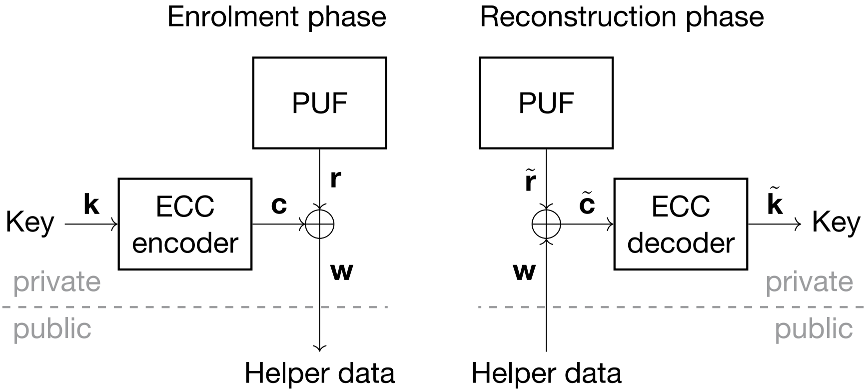
CHES 2022

Jonas Ruchti, Michael Gruber, Michael Pehl

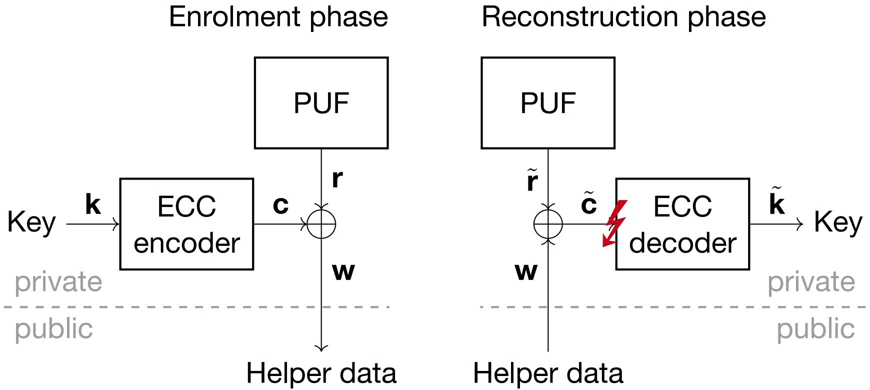
Chair of Security in Information Technology
School of Computation, Information and Technology
Technical University of Munich

2022-09-21

Overview: Fuzzy Commitment Scheme



Overview: Fuzzy Commitment Scheme



Context: Vulnerabilities of PUF Key Storage Schemes

- Physical attacks on PUF primitives: SCA, FIA
- Helper data manipulation attacks on secure sketches, e.g.
  Becker, “Robust Fuzzy Extractors and Helper Data Manipulation Attacks Revisited: Theory versus Practice”, 2019.
- Side-channel attacks on the error correction codes (ECCs) of PUF systems, e.g.
  Merli, Stumpf, and Sigl, *Protecting PUF Error Correction by Codeword Masking*, 2013.

This work: First fault injection analysis targeting ECC implementations for PUF key storage.

Outline

1. Attack
2. Experiment
3. Masking
4. PUF Noise
5. Realistic Attacker
6. Conclusion

Attack: Scenario

- Physical access to the device under attack
- Possibility to repeatedly trigger key reconstruction phases
- Pass/fail reconstruction phase result
- Serial transfer of the codeword to the ECC decoder

Attack: Scenario

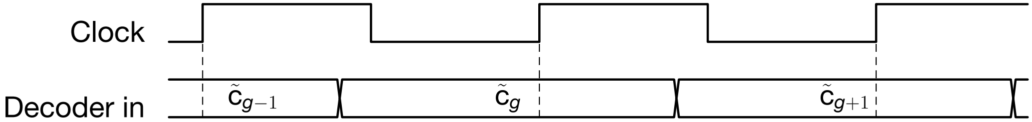
- Physical access to the device under attack
- Possibility to repeatedly trigger key reconstruction phases
- Pass/fail reconstruction phase result
- Serial transfer of the codeword to the ECC decoder

Initially: Powerful attacker

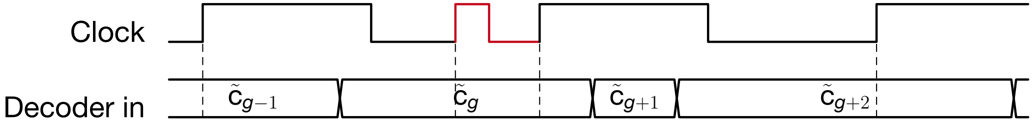
- Noise-free PUF response
- Read/write access to helper data
- Profiling and attack one the same device

Later: More realistic attacker

Attack: Fault Model



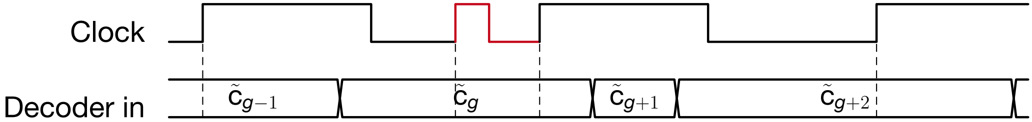
Attack: Fault Model



Clock glitch at bit position g:

- Bit $\tilde{c}_{g+1}$ is replaced by bit $\tilde{c}_g$

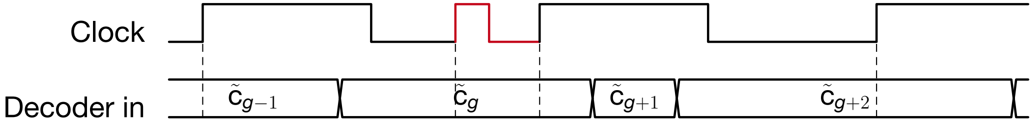
Attack: Fault Model



Clock glitch at bit position g:

- Bit $\tilde{c}_{g+1}$ is replaced by bit $\tilde{c}_g$
- Helper data modification $\Rightarrow$ ECC is at error correction limit

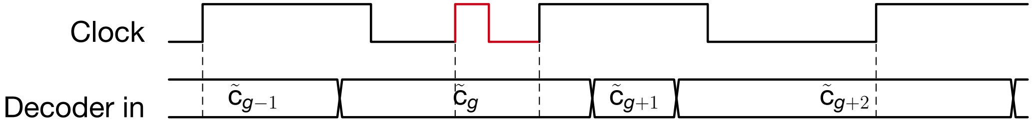
Attack: Fault Model



Clock glitch at bit position g:

- Bit $\tilde{c}_{g+1}$ is replaced by bit $\tilde{c}_g$
- Helper data modification $\Rightarrow$ ECC is at error correction limit
- Extraction of bit differences using clock glitch:
  - Reconstruction succeeds $\Rightarrow$ Bits at g and $g+1$ are the same
  - Reconstruction fails $\Rightarrow$ Bits at g and $g+1$ differ

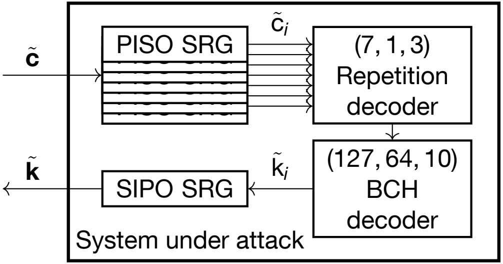
Attack: Fault Model



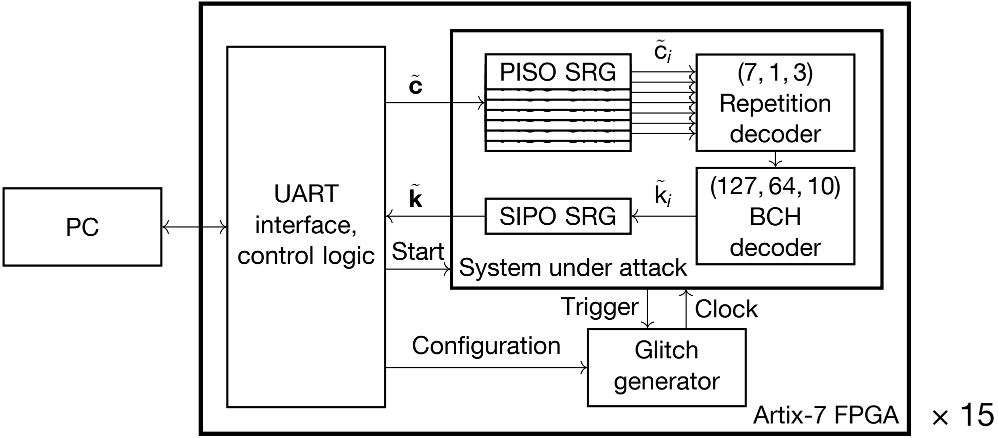
Clock glitch at bit position g:

- Bit $\tilde{c}_{g+1}$ is replaced by bit $\tilde{c}_g$
- Helper data modification $\Rightarrow$ ECC is at error correction limit
- Extraction of bit differences using clock glitch:
  - Reconstruction succeeds $\Rightarrow$ Bits at g and $g+1$ are the same
  - Reconstruction fails $\Rightarrow$ Bits at g and $g+1$ differ
- $n-1$ fault injections $\Rightarrow$ $n-1$ codeword bit differences recoverable

Experiment: Set-up



Experiment: Set-up



Experiment: Procedure

Profiling

- Estimate of observable data dependency
- One random key per board
- Random search to find best glitch parameters for each board

Experiment: Procedure

Profiling

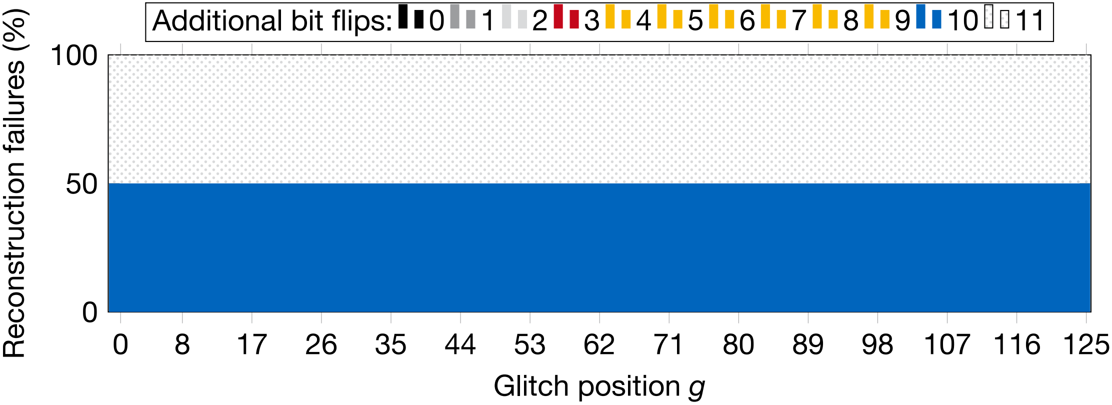
- Estimate of observable data dependency
- One random key per board
- Random search to find best glitch parameters for each board

Attack

- 15 FPGA boards
- 100 random keys attacked per board
- 250 trials with 127 clock glitches each
- Codeword extracted from average of trials

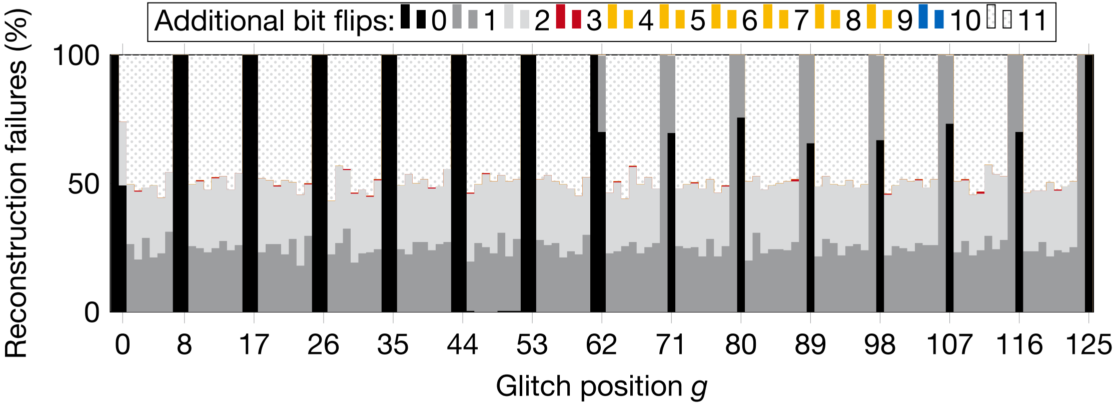
Glitch Effects: Model Prediction

Experiment: 250 random codewords with a varying number of extra bit flips



Glitch Effects: Observed

Experiment: 250 random codewords with a varying number of extra bit flips



Experiment: Results

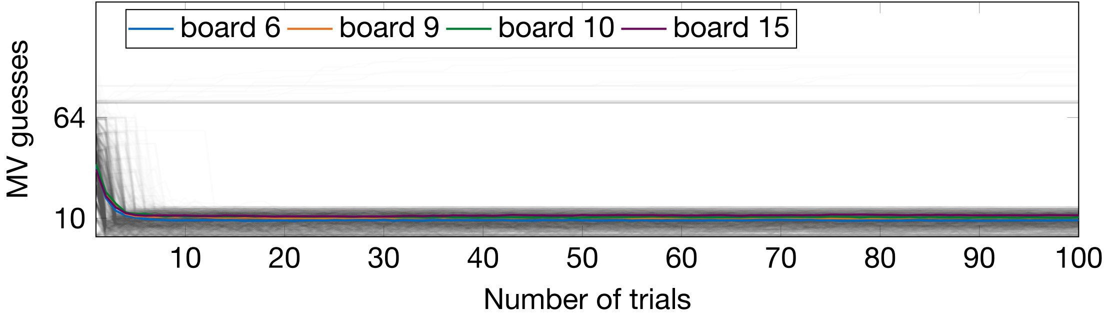
After 250 trials: 14.8 codeword bit extraction errors on average

Strategy	Guess entropy (average)
Naïve	24.7 bit

Experiment: Results

After 250 trials: 14.8 codeword bit extraction errors on average

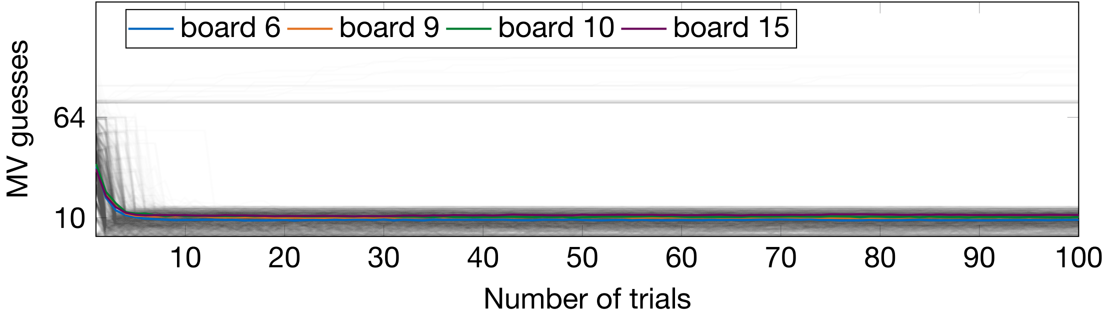
Strategy	Guess entropy (average)
Naïve	24.7 bit
Maximum-variance (MV)	10.4 bit



Experiment: Results

After 250 trials: 14.8 codeword bit extraction errors on average

Strategy	Guess entropy (average)
Naïve	24.7 bit
Maximum-variance (MV)	10.4 bit
Maximum extraction error probability	8.6 bit



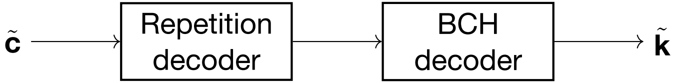
Masking

- Masking is effective against SCA with a similar scenario[1]
- Based on the fault model, masking *could* protect against the attack

[1] Merli, Stumpf, and Sigl, *Protecting PUF Error Correction by Codeword Masking*, 2013.

Masking

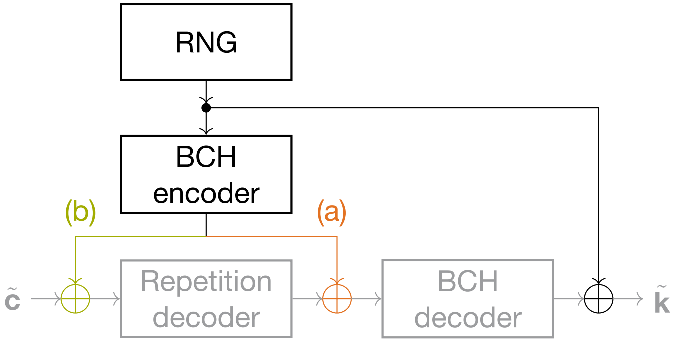
- Masking is effective against SCA with a similar scenario[1]
- Based on the fault model, masking *could* protect against the attack



[1] Merli, Stumpf, and Sigl, *Protecting PUF Error Correction by Codeword Masking*, 2013.

Masking

- Masking is effective against SCA with a similar scenario[1]
- Based on the fault model, masking *could* protect against the attack



[1] Merli, Stumpf, and Sigl, *Protecting PUF Error Correction by Codeword Masking*, 2013.

(a) Mask Applied at BCH Decoder Input

Experiment results

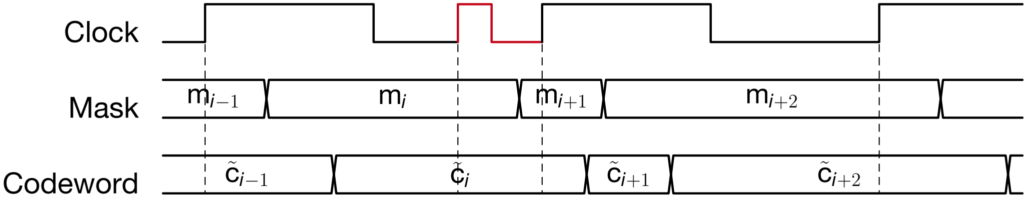
- Even better attack performance than for the unmasked case
- All 1500 tested codewords perfectly extractable

(a) Mask Applied at BCH Decoder Input

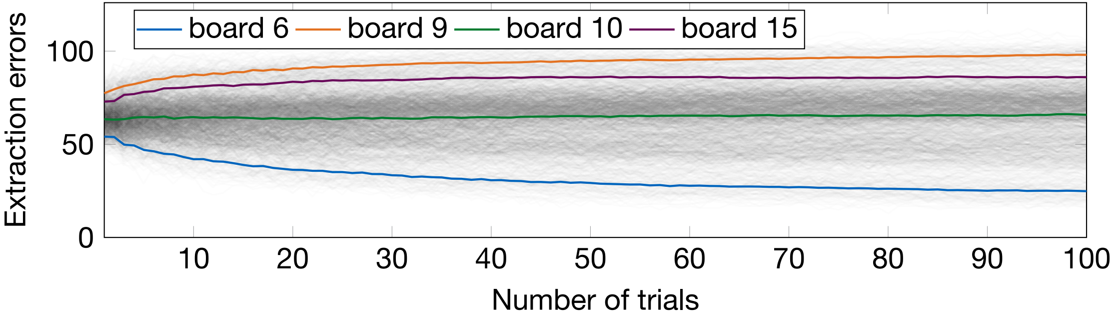
Experiment results

- Even better attack performance than for the unmasked case
- All 1500 tested codewords perfectly extractable

Why?



(b) Mask Applied at Repetition Decoder Input



Some amount of protection, but…

- Best-attackable board: 19 bit left to guess on average
- Mask/codeword propagation delay matching can be impractical

PUF Noise

Model

- Static response offset (e.g. ageing): Can be extracted and compensated
- Measurement noise: Must be compensated by averaging more trials

PUF Noise

Model

- Static response offset (e.g. ageing): Can be extracted and compensated
- Measurement noise: Must be compensated by averaging more trials

Experiment

- PUF response now has i.i.d. measurement noise
- Attack is carried out with different BERs

PUF Noise

Model

- Static response offset (e.g. ageing): Can be extracted and compensated
- Measurement noise: Must be compensated by averaging more trials

Experiment

- PUF response now has i.i.d. measurement noise
- Attack is carried out with different BERs

Results

- Repetition decoder limits noise influence:
  For $\text{BER}_{\text{PUF}} \leq 11\,\%$, fewer than 0.5 bit errors are left on average
- Attack progress is slower, but averaging can combat the remaining errors

⇒ Realistic error rates: Attack performance is nearly the same

Realistic Attacker

- Profile one device, attack a different device
- No more helper data access/manipulation
- Attacker can *increase* PUF noise

Realistic Attacker

- Profile one device, attack a different device
- No more helper data access/manipulation
- Attacker can *increase* PUF noise

Results

- Attack is significantly slower (approx. factor 10)
- Attack performs well except for 3 boards
- Best-attackable board: 9.6 bit guesses left on average after 250 trials

Summary

Conclusion

- FIA on the ECC in PUF key storage can be feasible and more powerful than SCA
- Masking is difficult to get right and can even make matters worse
- Helper data manipulation detection does not always help

Summary

Conclusion

- FIA on the ECC in PUF key storage can be feasible and more powerful than SCA
- Masking is difficult to get right and can even make matters worse
- Helper data manipulation detection does not always help

In the full paper

- Profiling, extraction of static PUF response offsets
- HD manipulation and guessing strategies for more efficient key/codeword recovery
- Error-correcting partially extracted codewords
- Other secure sketches (e.g. syndrome construction)