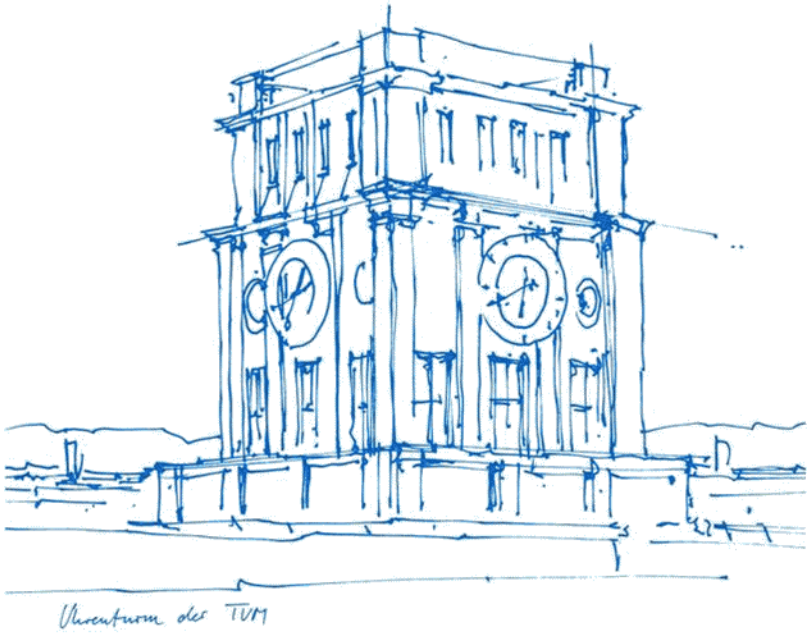
Chair of Security in Information Technology
Department of Electrical and Computer Engineering
Technical University of Munich

# Learning PUFs from Unsigned Stability: A Warning on Tolerable Leakage




**Niklas Stein**

**Michael Pehl**

Cascade Conference 2026

# Physical Unclonable Functions (PUFs)

Hardware security based on device fingerprints

Some properties like propagation delay are

- randomly distributed in each chip
- mostly constant
- difficult to read from outside the chip
- difficult to reproduce onto another chip

# PUFs as secure key storage

Challenges (bit vectors) configure each PUF output

Hidden output bits reconstruct secret keys

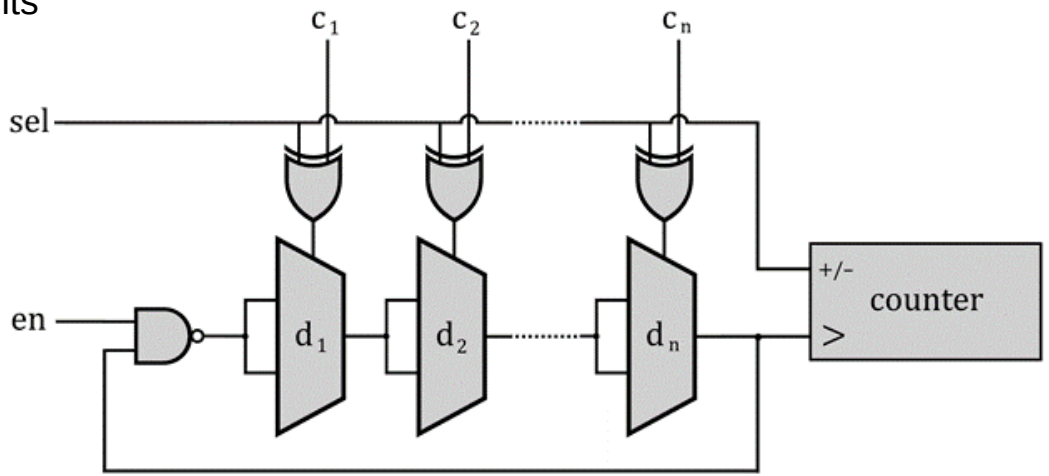
# Contribution

- PUFs are prone to Side-Channel and Modeling Attacks
- Various side-channel hardened designs proposed

**But how much side-channel leakage is tolerable when combined with modeling?**

# Loop PUF

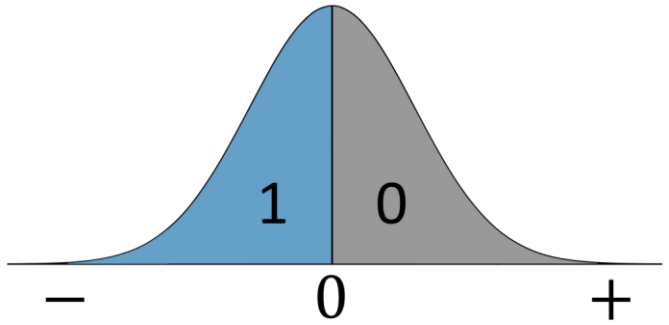
- Oscillators use difference of 2 measurements
- Delay path configured by challenge



# Quantization of PUFs

- Measured Values follow a normal distribution
- We need one (or multiple) decision boundaries

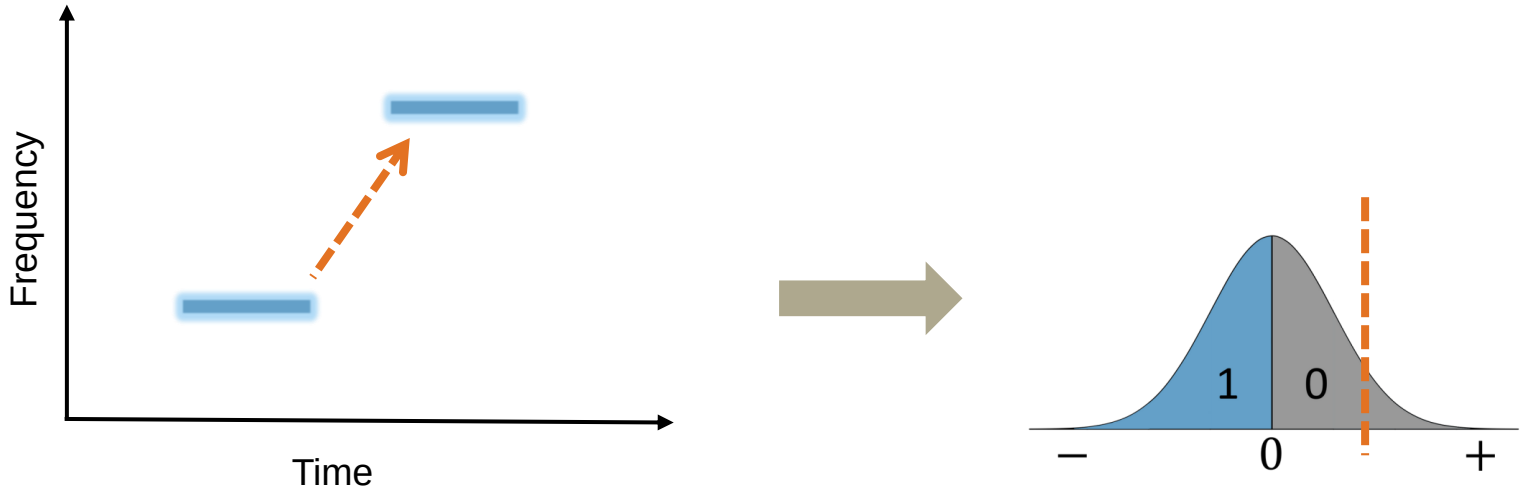
  ⟶ Sign function
- Larger distance means higher stability



# Side Channel Leakage

SCA of unprotected Loop PUF is simple: frequencies and order visible in power spectrum

→ attacker can obtain both sign and stability

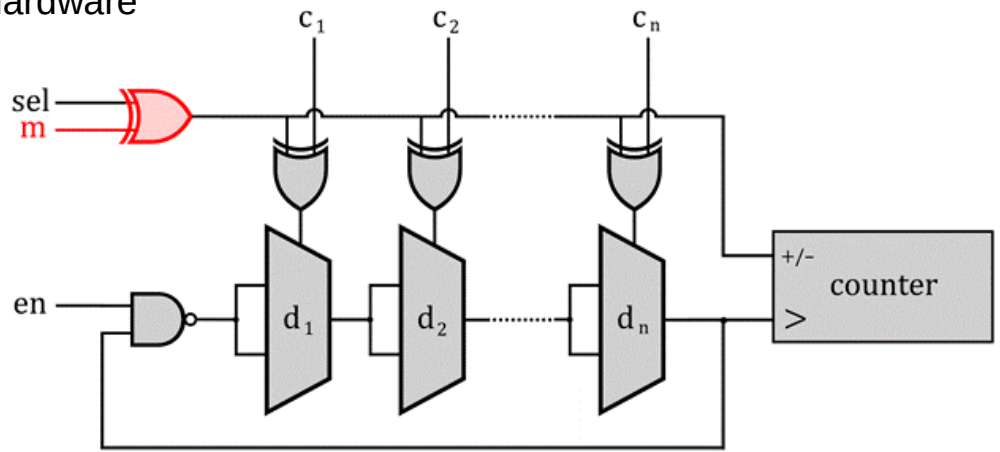


# Temporal Masking Loop PUF

Problem: Shuffling of challenges is expensive in hardware

Idea: only switch order of the 2 measurements

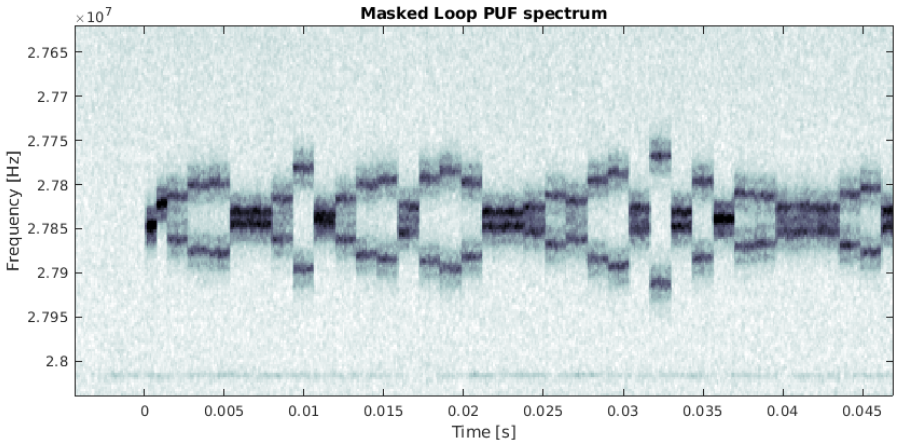
(L. Tebelmann et al.)

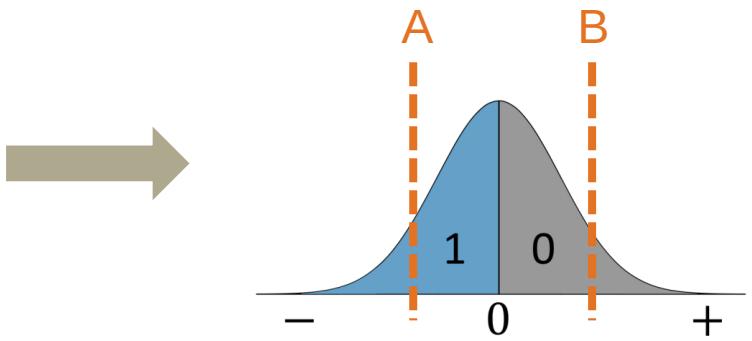


# Side Channel Leakage

Temporal masking still leaks stability, but not sign

→ two discrete values with equal probability



# Learning PUFs from Unsigned Stability

How can we train a model from only absolute values?

Complex PUF structures often learned with DNN

→ more CRPs needed than available in key storage, poor convergence

Our work:

- two-staged maximum-likelihood approach
- entropy reduction even for small number of CRPs

# Modeling of linear PUFs

Common PUF primitives can be modeled by an additive delay model (D. Lim et al.):

Dot product of (transformed) challenge vector $\mathbf{C}$ and stage-specific delay $\mathbf{d}$ equals analog secret $\mathbf{s}$

$$\vec{\mathbf{C}} \cdot \vec{d} = s$$

Model can be estimated from multiple noisy response values $\tilde{s}$

$$\vec{d} = [\mathbf{C}]^{-1} \cdot \vec{\tilde{s}}$$

# Maximum Likelihood Models

We can include knowledge about the distributions:

- delays $\mathbf{d}$ normal distribution with variance $\boldsymbol{\sigma_d}$ known from design, mean $\boldsymbol{\mu}$ assumed $\mathbf{0}$
- measurement noise $\mathbf{e}$ normal distribution with variance $\boldsymbol{\sigma_e}$ known from traces

$$\min_{d \in \mathbb{R}^n} \left\| \begin{bmatrix} \frac{1}{2{\sigma_e}^2} \\ \frac{1}{2{\sigma_d}^2} \end{bmatrix} \circ \left( \begin{bmatrix} \tilde{s} \\ \mu_0 \end{bmatrix} - \begin{bmatrix} \mathbf{C} \\ \mathbf{I} \end{bmatrix} \cdot \vec{d} \right)^2 \right\|$$

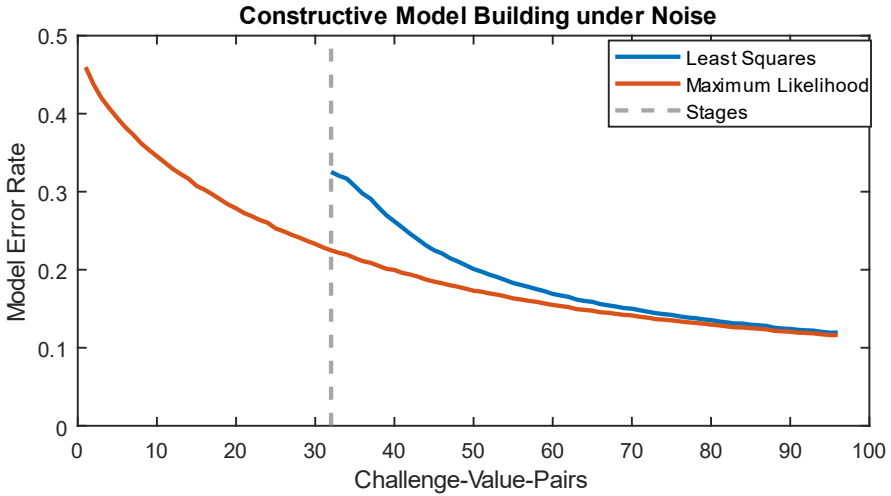
With this assumption: sum of squared errors equals likelihood of observations

→ weighted least squares problem

# Maximum Likelihood Models

With this ML estimator:

- Better noise rejection
- Definition for underdetermined systems



# Likelihood of secret outputs

Solution gives a model of the PUF

But also a residual error value of the mismatch

This error describes the likelihood of observations being consistent with physical device

for a hypothesis of secret analog values, this gives a likelihood of being correct

→ possible to rank decode all permutations of secret bits

# Likelihood of secret outputs

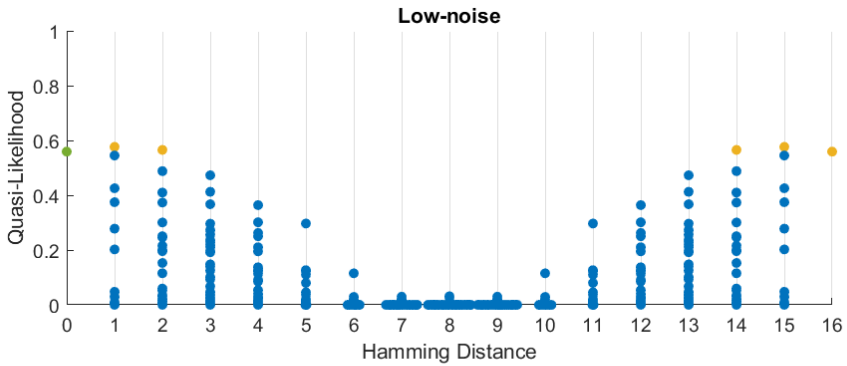
Example device: 8 stages, 16 random challenges

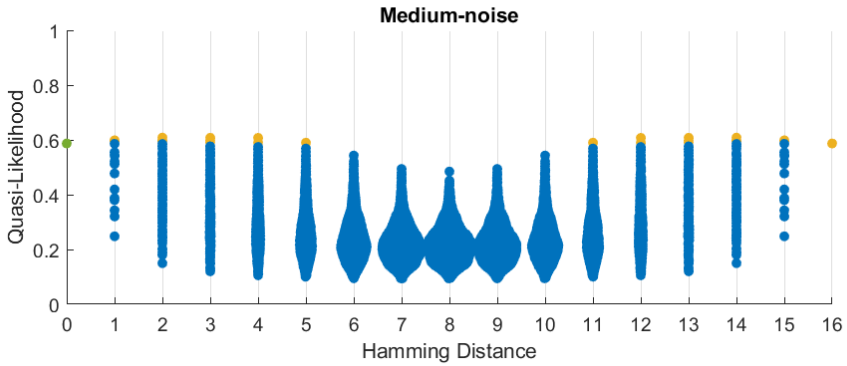
In total: $2^{16}$ hypotheses (blue)

2% noise: rank 5 (yellow)

50% noise: rank ~300 (yellow)

Due to symmetry one bit of entropy remains

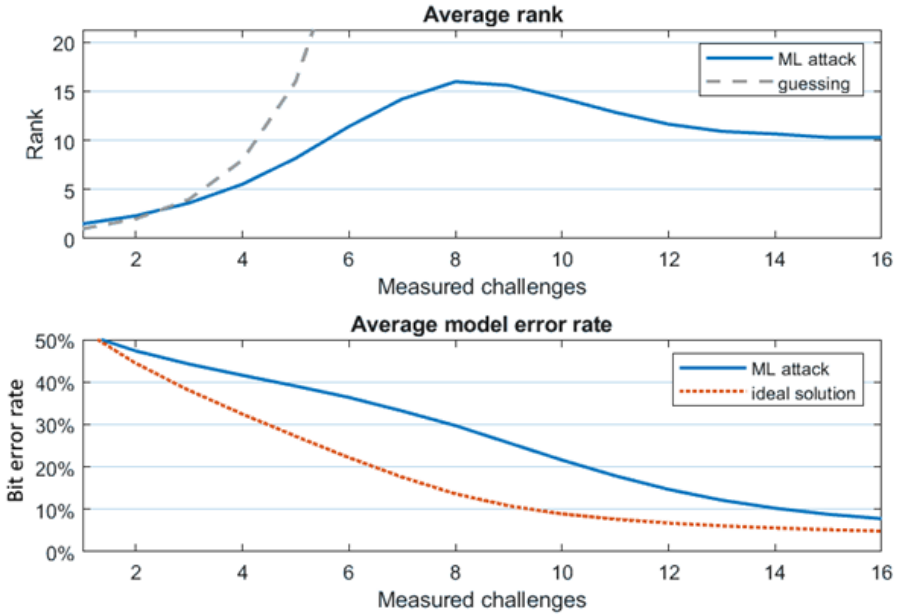


# Number of challenges

We can already reduce entropy with only a few measurements

But model will only become accurate after a larger number of challenges



# High-entropy challenge sets

We can maximize the entropy of a challenge set

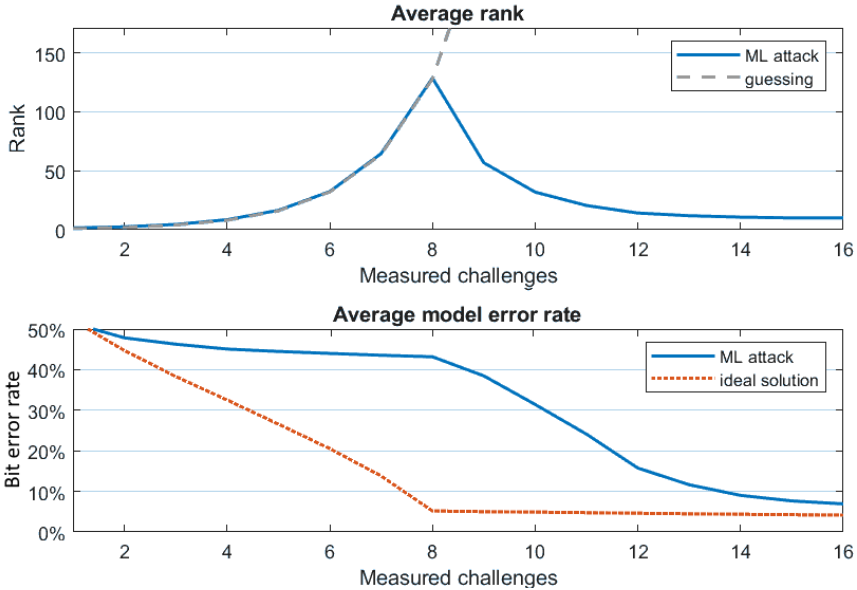
by picking orthogonal challenges (O. Rioul et al.)

Only n = 8 fully orthogonal challenges

→ resistant to this attack

Beyond that only pairwise orthogonal challenges exist

→ system breaks immediately



# Attack on large PUFs

Complexity polynomial to PUF size but exponential to number of measurements

Common designs can have hundreds of bits

Solution:

Find a high-likelihood combination with a Genetic Algorithm

- Population of 3000 random bit vectors
- Crossover of good scores keeps equal bits
- Near-optimum after 100 iterations for up to 256 bits

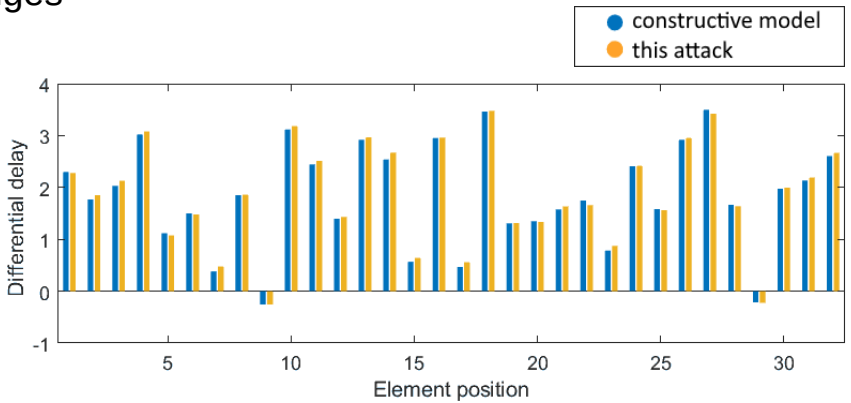
# Experimental results

Masked Loop PUF on FPGA with 32 stages, 128 challenges

- 200 power traces at 1.25 GSa/s
- ~1 min computation

→ only 2 bit errors in predicted secret

→ method not affected by bias in delays



# Conclusion

We can still train models on leakage which does not reveal the secret bit

→ multi-challenge designs leaking the unsigned stability are insecure

Orthogonal challenge sets are resistant but have a hard limit

Preventing amplitude leakage is resistant but expensive (Shuffling, ICLooPUF)