Robust and Reliable PUF Protocol Exploiting NMQ and the Neyman-Pearson Lemma

Neelam Nasir, Julien Béguinot, Wei Cheng, Ulrich Kühne, and Jean-Luc Danger

Télécom Paris - Institut Polytechnique de Paris

2025-04-05

Overview

- Background
  - Physical Unclonable Functions
  - The Loop PUF
- Reliability Evaluation
- Security Evaluation
- Novel Protocol
  - Taking Reliability into Account
  - Protocol
- Evaluation
- Conclusion

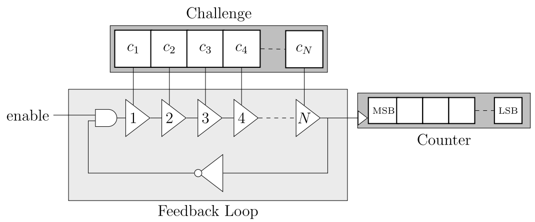
Background

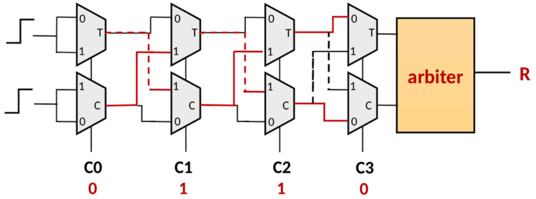
Background

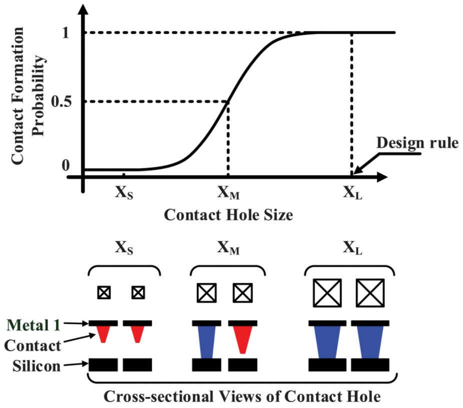
Physical Unclonable Functions

Physical Unclonable Functions

- Unique device **fingerprint**
- Low-cost security anchor
- Avoids non-volatile memory
- Based on **physical** property
  - Delay
  - Resistance
  - Process variations
  - ...
- Cannot be copied by design
- **Many** different architectures







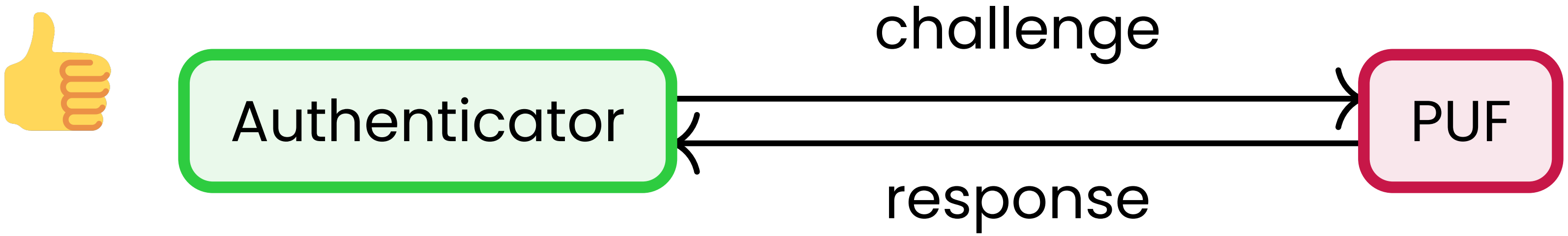
Applications

1. Key derivation

Applications

1. Key derivation

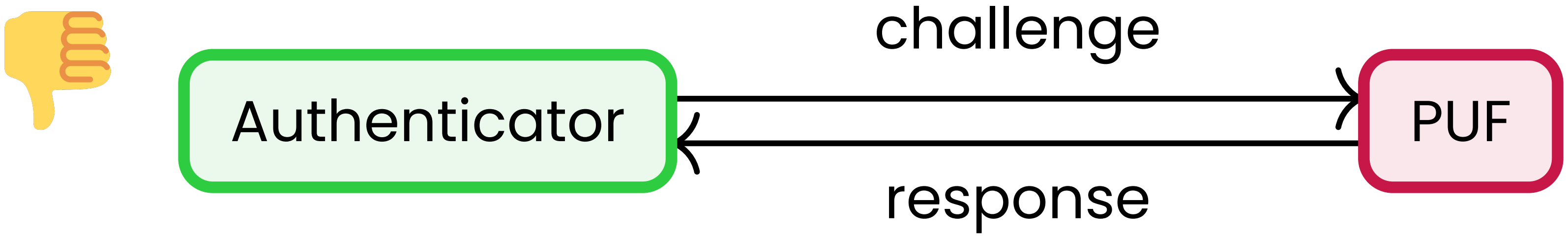
2. Challenge-response authentication



Applications

1. Key derivation

2. Challenge-response authentication



Design Objectives

(Conflicting) design objectives for PUFs:

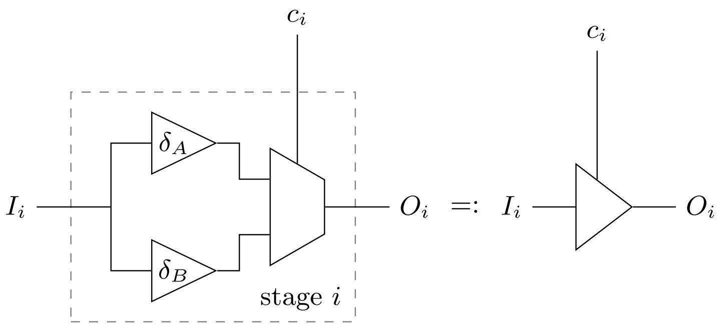
- Logic complexity
  - Should be **small**
- Performance
  - Should be **fast**
- Reliability
  - Should be **insensitive** to environmental conditions
  - Response should be **stable** over time
- Security
  - Should **resist** against attacks

Background

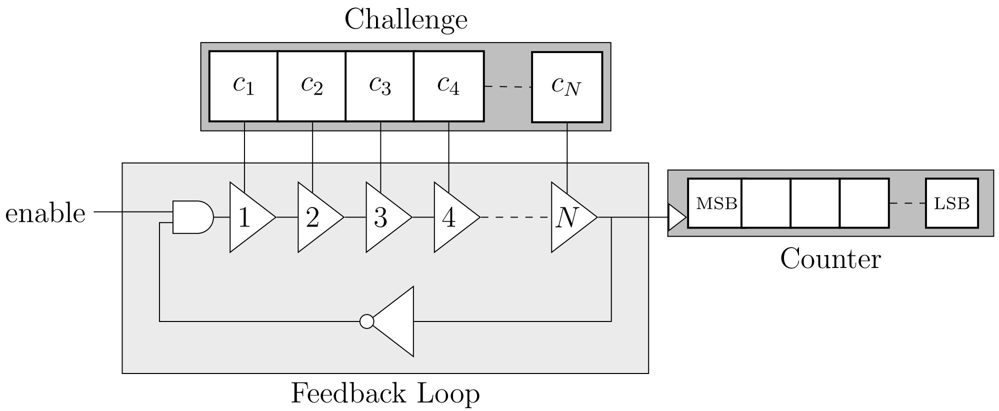
The Loop PUF

Configurable Ring Oscillator

Delay element:



Global structure:



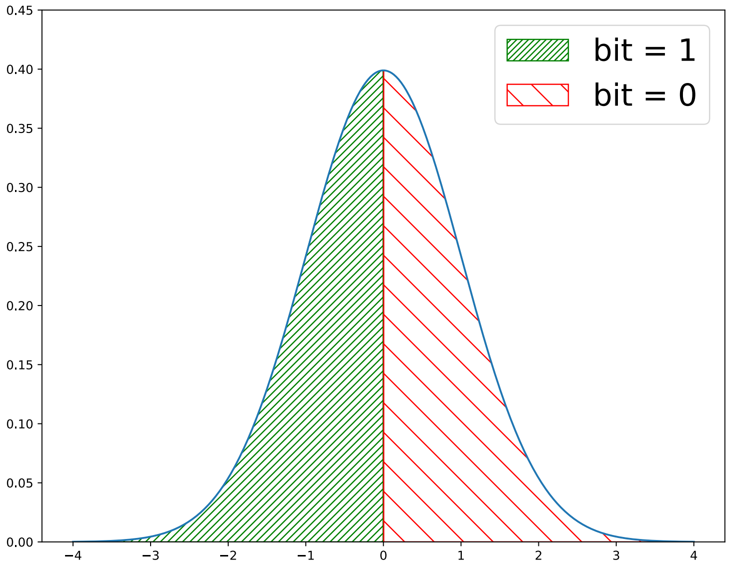
- Measure frequency/delay induced by challenge C

Model

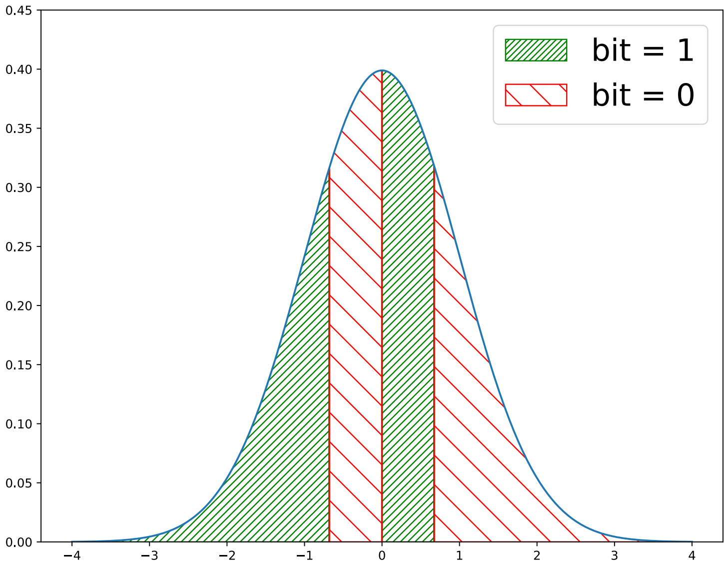
$$C \xrightarrow{\mathcal{N}(\mu, \Sigma)} \delta_C \xrightarrow{\mathcal{N}(0, \sigma)} \hat{\delta}_C \xrightarrow{b} \{0, 1\}$$

C	Challenge
Σ	Standard deviation of delay variances (fixed for each device)
δ_C	Raw response (delay difference between C and $\overline{C}$)
σ	Standard deviation of measurement noise
$\hat{\delta}_C$	Noisy response
b	Quantization function

Quantization

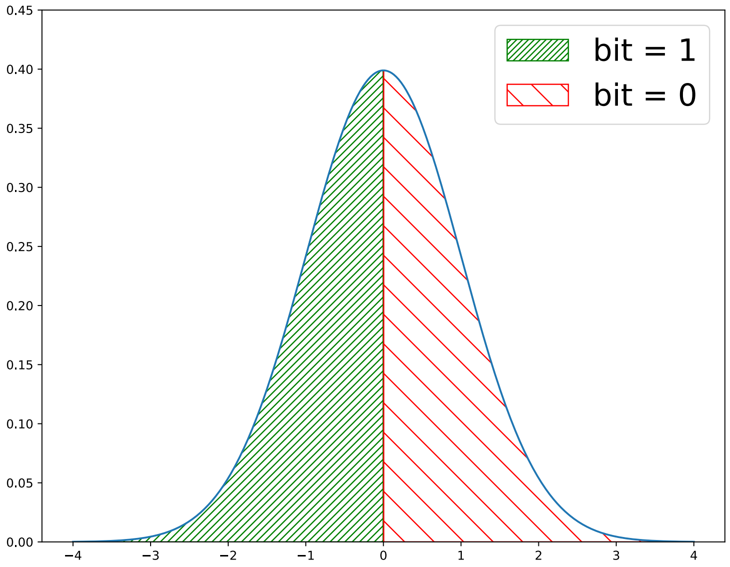


sign function



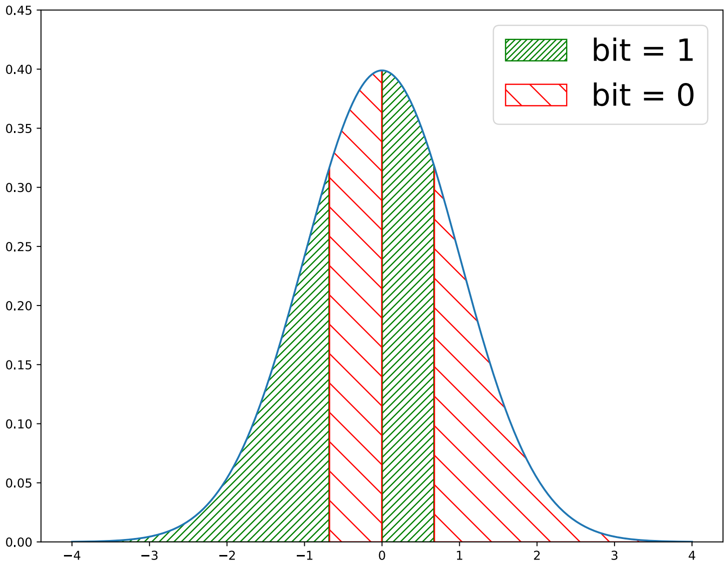
Non-monotonic quantization

Quantization



sign function

$Q = 2$



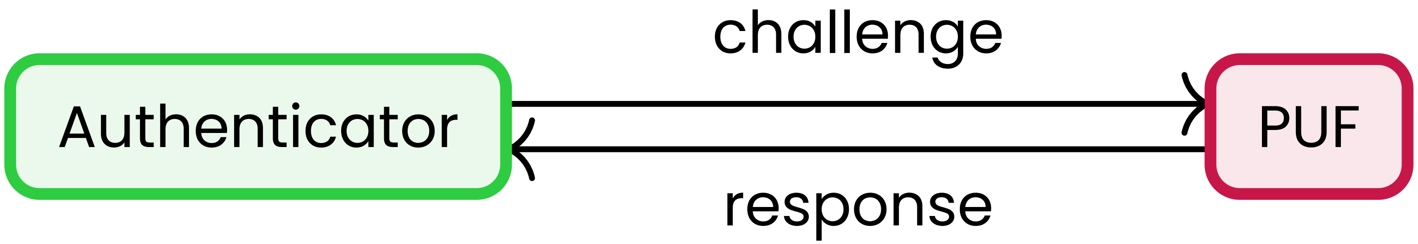
Non-monotonic quantization

$Q = 4$

Reliability Evaluation

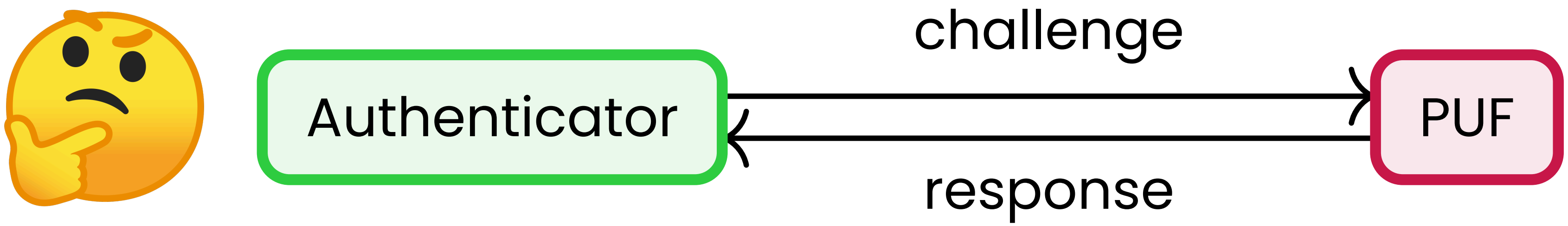
Bit-error Rate

- The PUF's raw response $\hat{\delta}_C$ is **noisy**
- The **same challenge** might give **differents reponses**
- **Bit error rate** (BER):
  - Probability that the response differs from the nominal response



Bit-error Rate

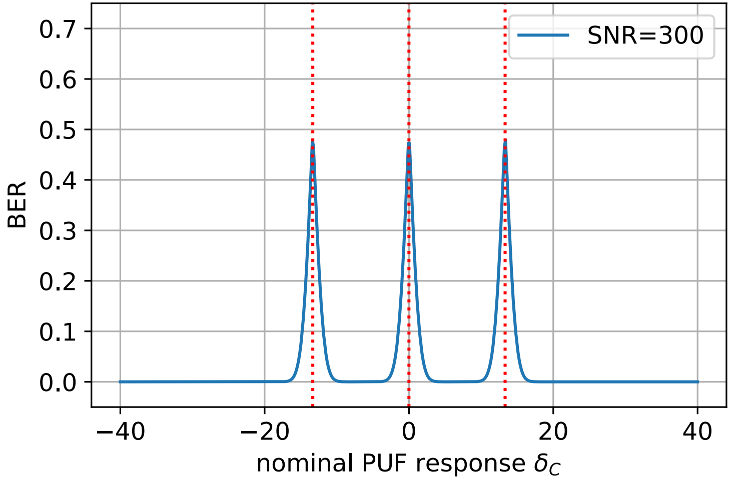
- The PUF’s raw response $\hat{\delta}_C$ is **noisy**
- The **same challenge** might give **differents reponses**
- **Bit error rate** (BER):
  - Probability that the response differs from the nominal response

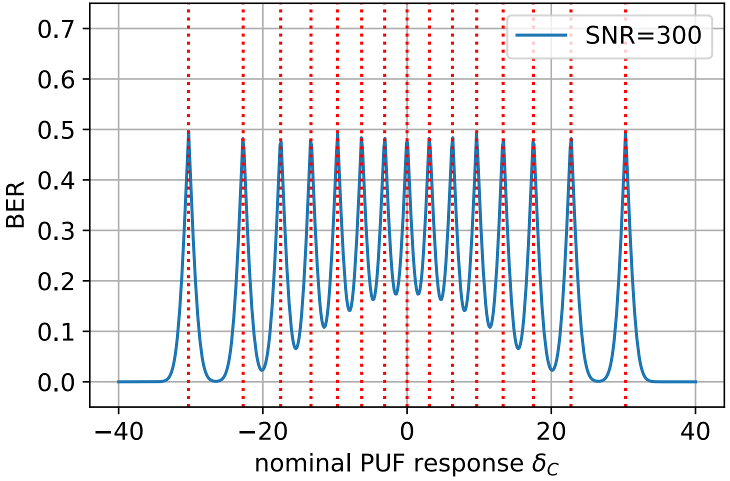


Reliability Analysis of NMQ

- Raw response close to threshold $\Longrightarrow$ **high BER**
- Higher $Q \Longrightarrow$ more thresholds $\Longrightarrow$ **higher BER**

$Q = 4$



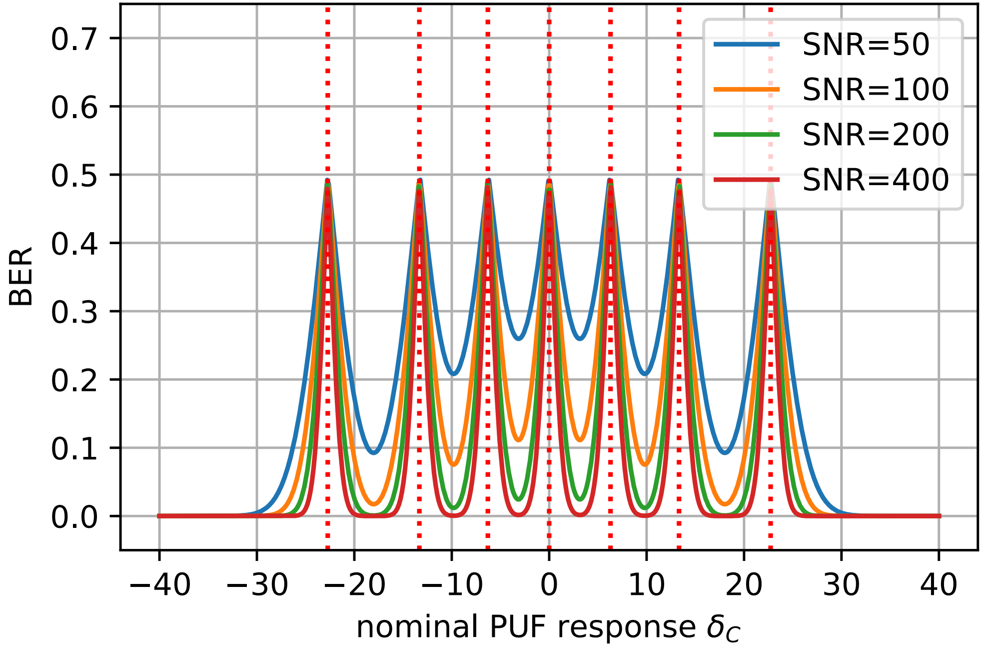


$Q = 16$

Nominal PUF response vs BER

Reliability Analysis of NMQ

- Higher **noise** level $\Longrightarrow$ **higher BER**



Nominal PUF response vs BER for $Q = 8$

Security Evaluation

Modeling Attacks

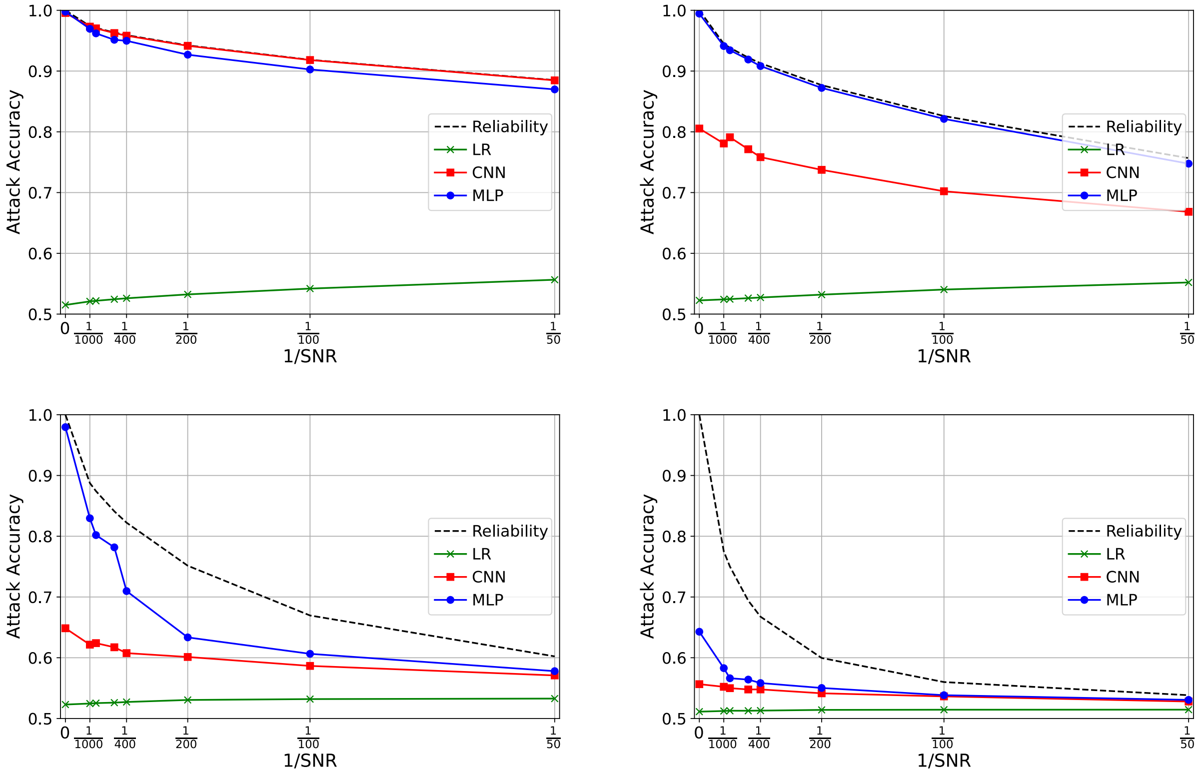
- Try to impersonate a PUF
- Learning from intercepted challenge-response pairs
- Construction of a **model**
- Different learning strategies
  - Logistic regression
  - **Deep learning**

Attack Strategy

- Attack on Loop-PUF with NMQ
  - $Q \in \{4, 8, 16, 32\}$
- Simulated PUF with different noise levels
  - $\mathrm{SNR} \in \{50, 100, ..., 1000, \infty\}$
- Real-world FPGA implementation
- Machine learning
  1. Logistic Regression (LR)
  2. Convolutional Neural Network (CNN)
  3. Multi-layer Perceptron (MLP)

Attack Results (Simulated)

$Q = 4$ $Q = 8$

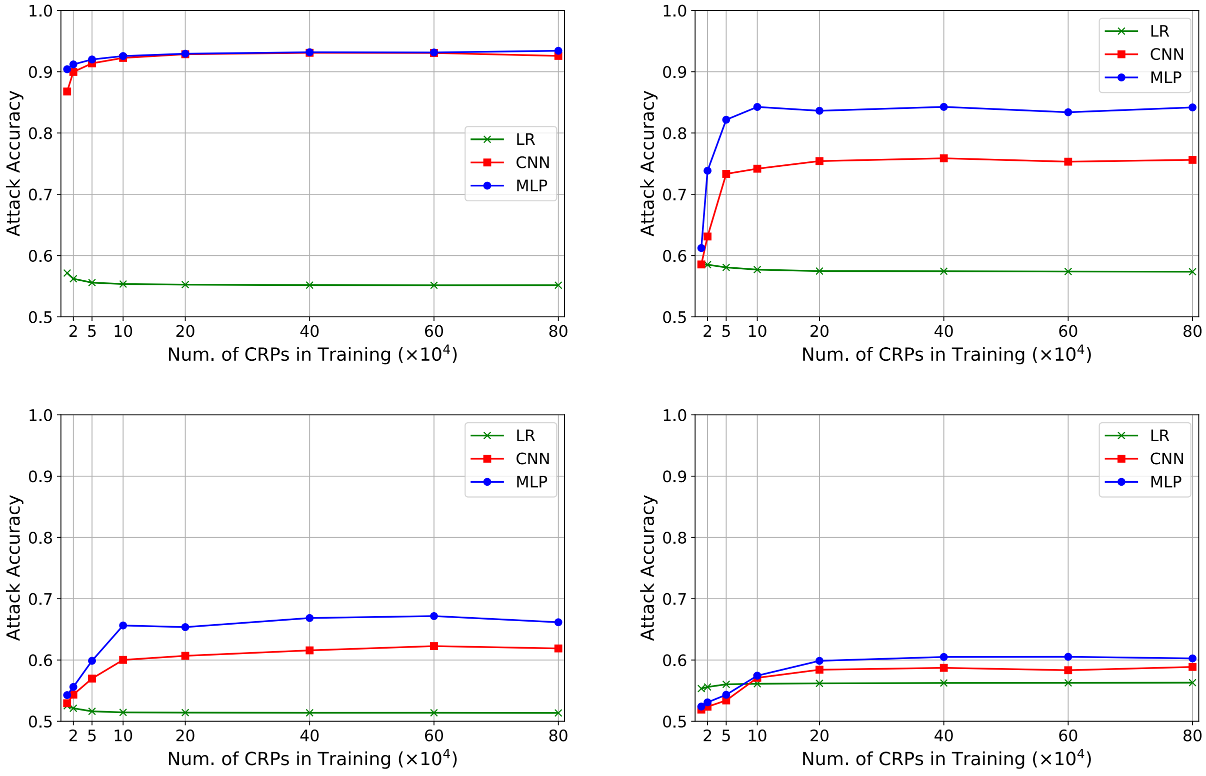


$Q = 16$ $Q = 32$

Noise level vs attack accuracy

Attack Results (FPGA)

$Q = 4$ $Q = 8$



$Q = 16$ $Q = 32$

Number of training CRPs vs attack accuracy

Security Evaluation

- Logistic Regression defeated by NMQ
- Low quantization levels ($Q \in \{4, 8\}$) **attackable** using CNN and MLP
- Higher quantization levels ($Q \in \{16, 32\}$) are reasonably **secure**
- Noise makes attacks **more difficult**

However:

The quantization levels needed for good resistance against ML attacks lead to very poor reliability ☹

How to achieve both high reliability and good security?

Novel Protocol

Problem Statement

How to authenticate a PUF wich migh sometimes give wrong responses?

Problem Statement

How to authenticate a PUF wich migh sometimes give wrong responses?

- Need for **more challenges**
  - How many?

Problem Statement

How to authenticate a PUF wich migh sometimes give wrong responses?

- Need for **more challenges**
  - How many?

Can we use reliability information?

Problem Statement

How to authenticate a PUF wich migh sometimes give wrong responses?

- Need for **more challenges**
  - How many?

Can we use reliability information?

- Only send challenges with **high reliability**
  - This gives away too much information for an attacker!

Problem Statement

How to authenticate a PUF wich migh sometimes give wrong responses?

- Need for **more challenges**
  - How many?

Can we use reliability information?

- Only send challenges with **high reliability**
  - This gives away too much information for an attacker!

Using reliability to weight responses

Novel Protocol

Taking Reliability into Account

Taking Reliability into Account

Hypotheses:

- H_0: The device is **legitimate**.
- H_1: The device is an **adversary**.

Taking Reliability into Account

Hypotheses:

- H_0: The device is **legitimate**.
- H_1: The device is an **adversary**.

For response R to n challenges C_i with **observed error** $e_i \in \{0, 1\}$:

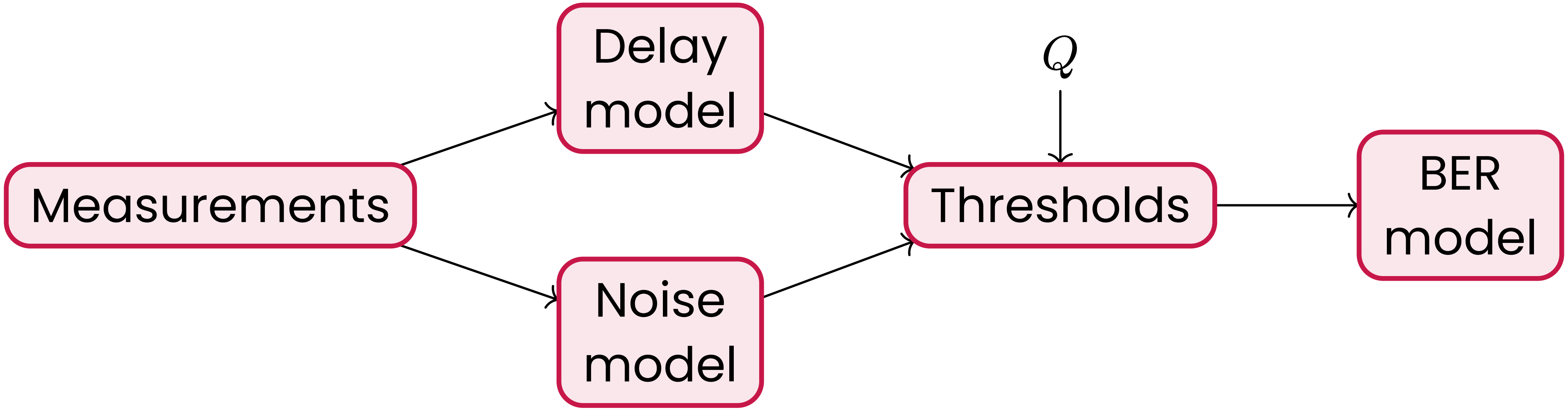
$$\alpha = \frac{L(R|H_0)}{L(R|H_1)} = \frac{\prod_{i=0}^{n} \mathrm{BER}(C_i)^{e_i} (1 - \mathrm{BER}(C_i))^{1-e_i}}{\left(\frac{1}{2}\right)^n}$$

is the **likelihood ratio** of the response coming from a legitimate device versus a random adversary.

Novel Protocol

Protocol

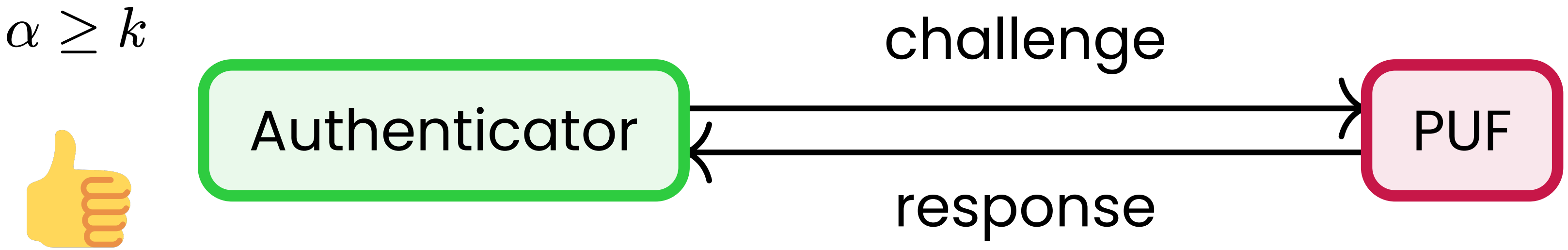
Enrollment Phase



- **Reliability model**:
  - Measure raw delays for chosen (*Hadamard*) challenges
  - Construct delay and noise models
  - Derive thresholds
  - Store reliability model on server-side

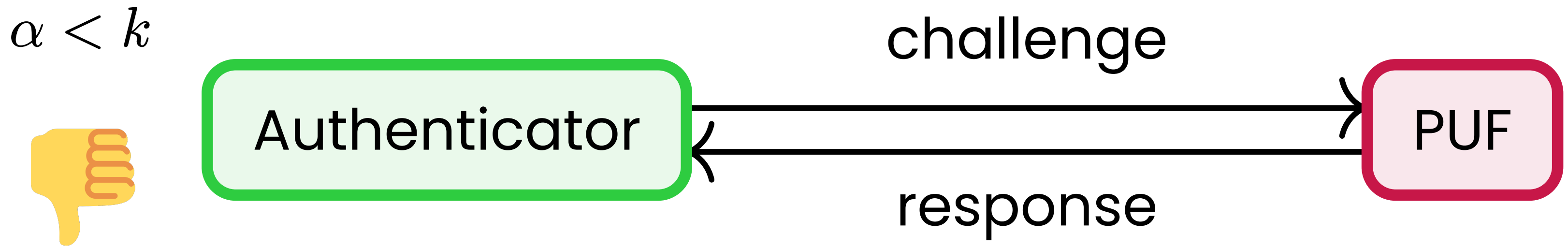
Authentication

- Server sends n challenges
- PUF replies with reponse R
- Server computes α
- Server accepts authentication if α is above chosen threshold k



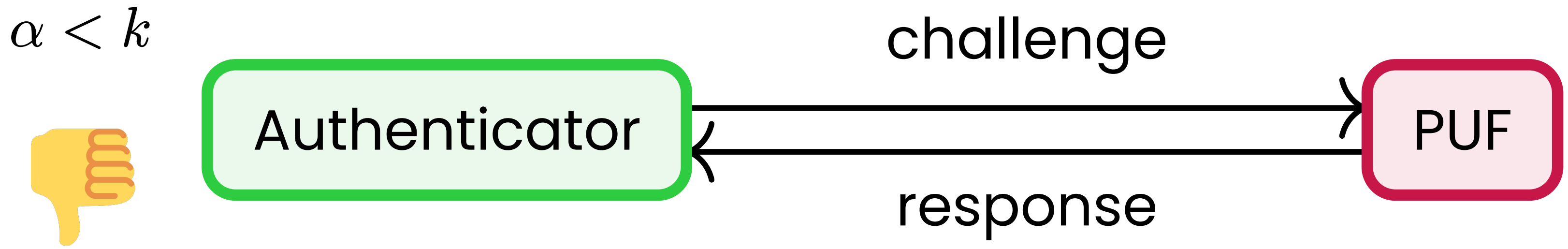
Authentication

- Server sends n challenges
- PUF replies with reponse R
- Server computes α
- Server accepts authentication if α is above chosen threshold k



Authentication

- Server sends n challenges
- PUF replies with reponse R
- Server computes α
- Server accepts authentication if α is above chosen threshold k



How to choose threshold k? $\Longrightarrow$ experiments

Evaluation

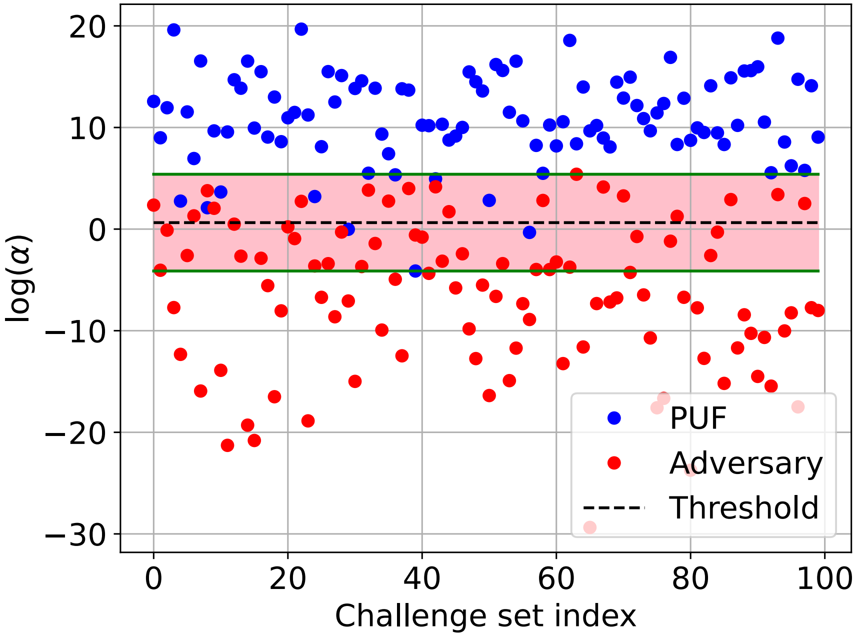
Setup and Experiments

- Loop-PUF design
  - Delay chain with 64 elements
  - 16 bit counter values
  - FPGA implementation on Basys-3 (Xilinx Artix-7 28nm)
- Experiments
  - Authentication threshold
  - **False authentication** probability

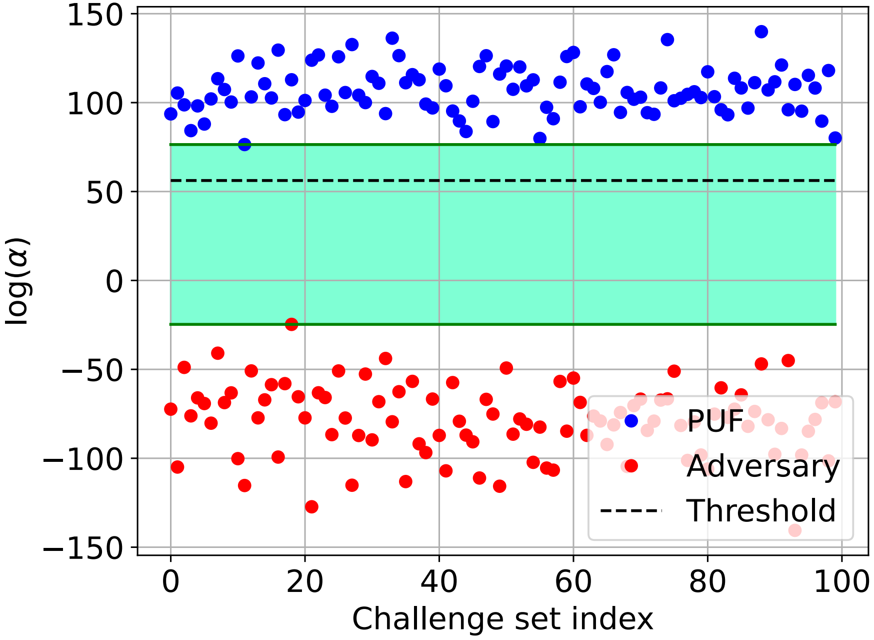
Authentication Threshold

- Extract reliability model from PUF (enrollment)
- Sample responses for sets of random challenges
- Evaluate α for different set sizes

Authentication Threshold



50 challenges



500 challenges

Safety windows for $Q = 16$

False Authentication Probability

- Authentic device vs random adversary
- Varying number of challenges
  - $N \in \{50, ..., 500\}$
- Varying quantization level
  - $Q \in \{4, 8, 16, 32\}$
- Setting authentication threshold on the safe side
  - Probability of rejecting genuine device ≈ 0
- Tested on 16 different FPGA devices

False Authentication Probability

n	$Q = 4$	$Q = 8$	$Q = 16$	$Q = 32$
50	$0.008 - 0.055$	$0.052 - 0.119$	$0.115 - 0.268$	$0.220 - 0.361$
100	$0.000 - 0.011$	$0.006 - 0.035$	$0.046 - 0.140$	$0.159 - 0.315$
150	$3.605 \times 10^{-5} - 0.001$	$0.002 - 0.012$	$0.016 - 0.104$	$0.073 - 0.255$
200	$1.345 \times 10^{-6} - 3.134 \times 10^{-4}$	$0.001 - 0.004$	$0.010 - 0.068$	$0.050 - 0.211$
250	$1.220 \times 10^{-5} - 2.509 \times 10^{-4}$	$2.527 \times 10^{-4} - 0.002$	$0.006 - 0.044$	$0.034 - 0.168$
300	$1.090 \times 10^{-9} - 2.285 \times 10^{-7}$	$1.006 \times 10^{-5} - 0.001$	$0.002 - 0.034$	$0.036 - 0.134$
350	$2.125 \times 10^{-10} - 2.484 \times 10^{-7}$	$1.250 \times 10^{-5} - 1.291 \times 10^{-4}$	$0.001 - 0.029$	$0.024 - 0.129$
400	$1.221 \times 10^{-6} - 1.781 \times 10^{-5}$	$3.727 \times 10^{-6} - 8.169 \times 10^{-4}$	$0.001 - 0.015$	$0.005 - 0.105$
450	$1.607 \times 10^{-9} - 3.337 \times 10^{-7}$	$1.478 \times 10^{-6} - 1.986 \times 10^{-4}$	$2.252 \times 10^{-4} - 0.007$	$0.006 - 0.079$
500	$1.004 \times 10^{-10} - 1.195 \times 10^{-7}$	$4.542 \times 10^{-6} - 8.002 \times 10^{-6}$	$2.105 \times 10^{-4} - 0.004$	$0.004 - 0.076$
550	$1.021 \times 10^{-13} - 4.862 \times 10^{-10}$	$1.579 \times 10^{-8} - 2.908 \times 10^{-7}$	$1.812 \times 10^{-5} - 0.001$	$0.004 - 0.065$
600	$0.0 - 1.519 \times 10^{-12}$	$1.014 \times 10^{-8} - 1.576 \times 10^{-5}$	$2.435 \times 10^{-5} - 0.001$	$0.004 - 0.052$

Conclusion

Conclusion

- Study of the **Loop-PUF** as authentication anchor
- Looking for interesting **security-reliability** trade-offs
- Evaluation of resistance to **machine learning** attacks
- Non-monotonic quantization improves security
- Compensation for poor reliability at protocol level

Questions ?