A New Lightweight Machine Learning Attack Resistant Multi-PUF Design

Qingqing Ma[1], Chongyan Gu[2], Neil Hanley[2], Chenghua Wang[1], **Weiqiang Liu[1]**, Maire O'Neill[2]

[1] CEIE, Nanjing University of Aeronautics and Astronautics
[2] ECIT, Queen's University Belfast

content

- ➢ Physical Unclonable Function (PUF)
- ➢ Modeling Attacks on PUF
  - ✓ Logistic Regression (LR)
  - ✓ Evolution Strategies (ES)
- ➢ Design of Multi-PUF
  - ✓ Proposed Multi-PUF Constructions
  - ✓ Attack Results using LR and ES
  - ✓ Uniqueness and Uniformity

Physical Unclonable Function (PUF)

Physical unclonable functions (PUF) is a promising technique to provide unclonable authentication and online random key generation for Internet of Things (IoT) Devices.

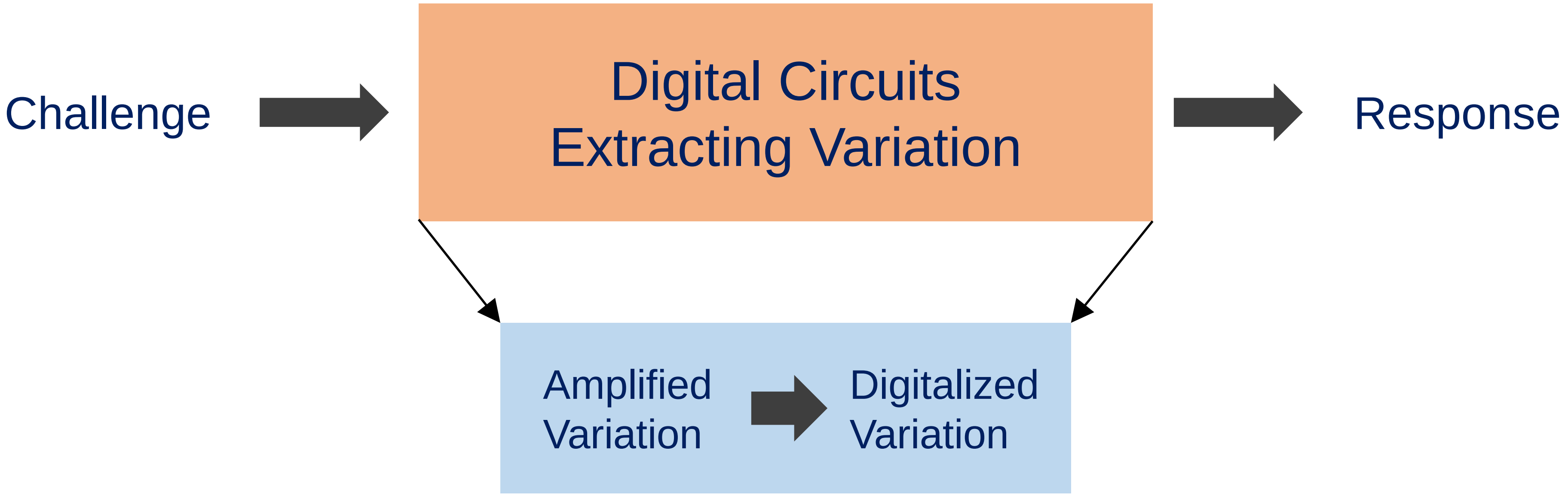
PUF can distinguish between inconsistencies in ICs that occur during fabrication.

PUF uses this feature to compute a unique tag response to uniquely identify each IC.

Even if an IC with PUF is cloned, the cloned IC response will differ from that of the genuine IC.

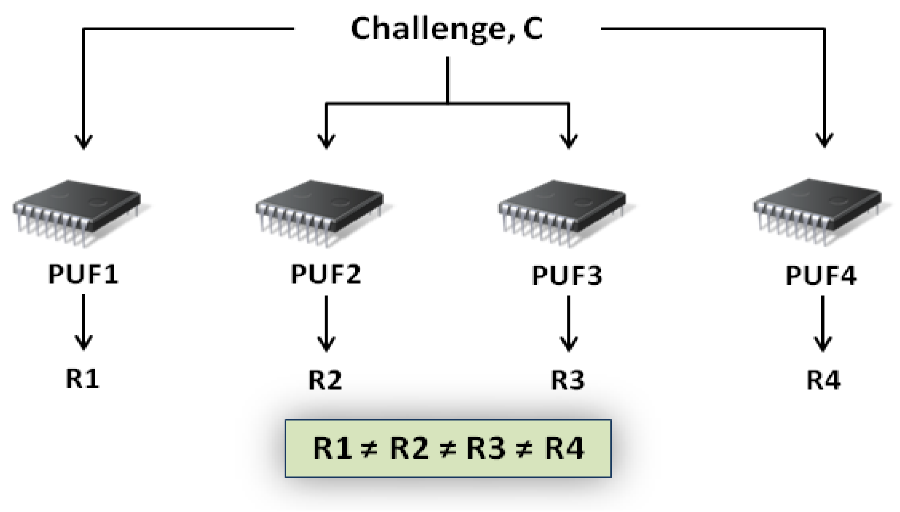
Physical Unclonable Function (PUF)

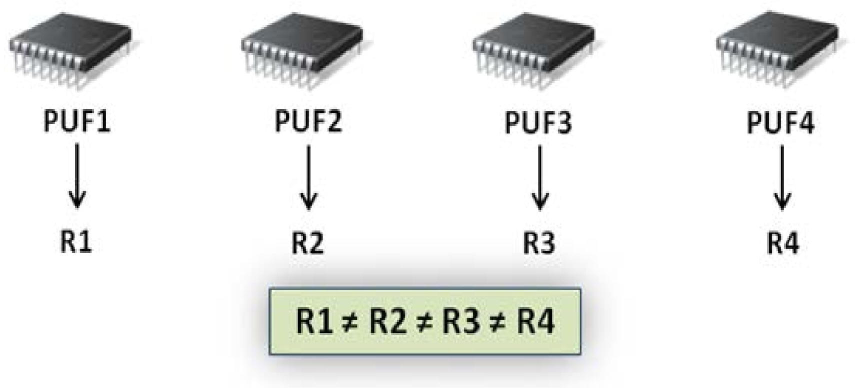
- PUF circuit takes **challenge** as the input and extracts the variation to produce the **response**.
- One PUF circuit can have several or lots of **challenge–response pairs (CRPs)** .



Physical Unclonable Function (PUF)

Two types of PUFs:





Strong PUF

- ✓ With large number of CRPs
- ✓ For device authentication

Weak PUF

- ✓ With small number of CRPs
- ✓ For key/ID generation

PUF Security Analysis

PUF could be attacked by the following methods:

- **Modeling attacks**
  - ✓ Only works for Strong PUF
  - ✓ Effective, and easy to implement
  - ✓ But strongly depends on the type of PUF
- **Side channel analysis**
  - ✓ Works for both Strong PUF and Weak PUF
  - ✓ Can improve the performance of modeling attack
  - ✓ High cost, and difficulty to operate
- **Exploiting the vulnerability of algorithms or protocols**
  - ✓ Without accessing to PUF
  - ✓ A threaten for both protocols and PUFs

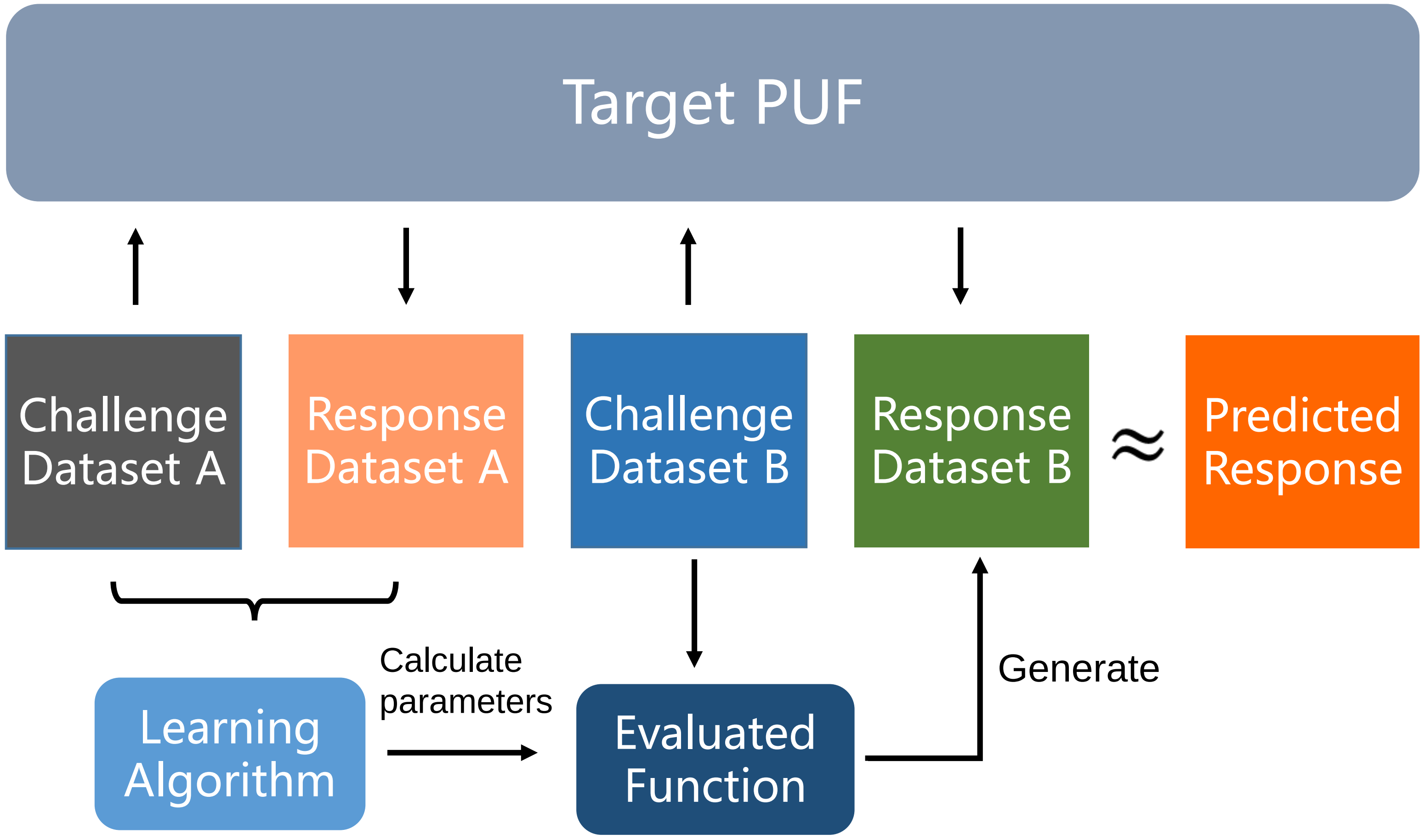
Modeling Attacks on PUF

Step 1:
Collecting adequate CRPs

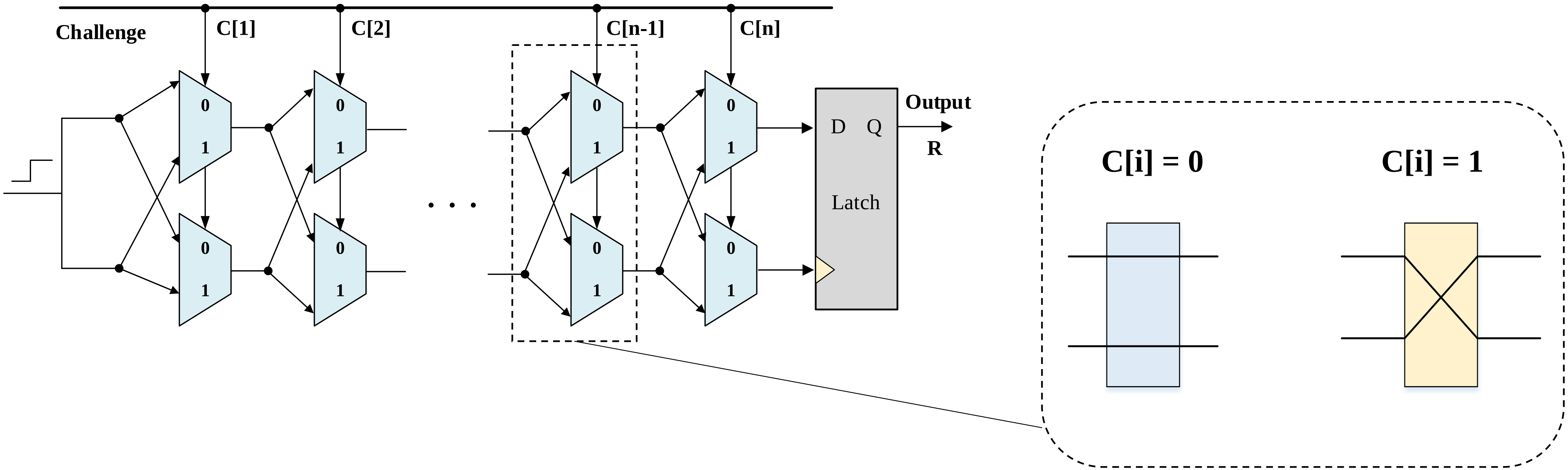
Step 2:
Choosing an appropriate model or algorithm (usually machine learning algorithms)

Step 3:
Calculating the parameters with the provided CRPs

Step 4:
With the trained model and a new challenge, one can predict the response without the PUF circuit.

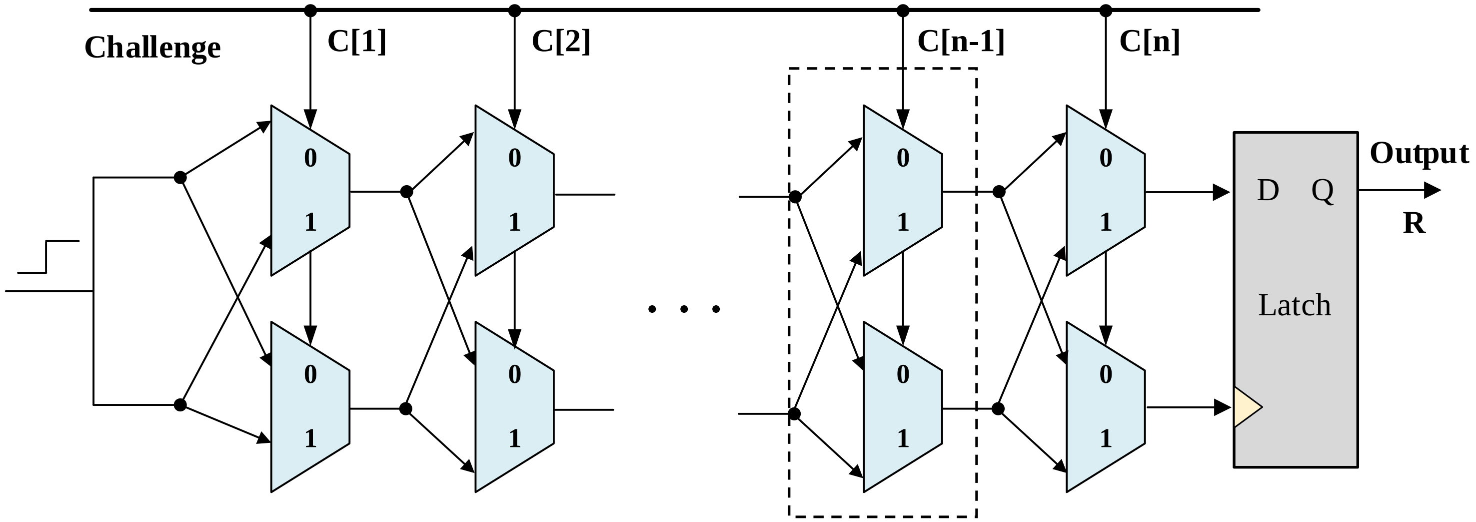


Arbiter PUF as An Example



The circuit has a multiple-bit input C (challenge) and computes a 1-bit output R (response) based on the delay difference between two paths with the same wire length.

Delay Model and Analysis of Arbiter PUF



1. $\mathbf{R} = \begin{cases} 0 & \Delta < 0 \\ 1 & \Delta \geq 0 \end{cases}$

2. $\Delta = \boldsymbol{\varphi}(\boldsymbol{C}) \cdot \mathbf{W}$
$= \varphi(c_1) \cdot \omega_1 + \varphi(c_2) \cdot \omega_2 + \cdots + \varphi(c_n) \cdot \omega_n + \omega_{n+1}$

3. $\mathbf{W} = (\omega_1, \omega_2 \ldots, \omega_n, \omega_{n+1})$

4. $\mathbf{C} = (c_1, c_2 \ldots, c_n)$,

$\boldsymbol{\varphi}(\mathbf{C}) = (\varphi(c_1), \varphi(c_2) \ldots, \varphi(c_n), 1)$

$\varphi(c_i) = \prod_i^n (1 - 2c_i)$, $c_i \in \{0,1\}$

R : response
W: delay parameters of Arbiter PUF
C : challenge

Known to us: $\mathbf{C}$, *i.e.* $\boldsymbol{\varphi}(\mathbf{C})$

Unknown: **W**

Goal: To calculate the vector **W**

Others: The delay difference Δ is linearly added by the delay parameters(ω_i); so Arbiter PUF is an additive delay model.

Algorithms for Modeling Attacks

Logistic Regression

Logistic Regression (LR) is a classification model, which estimates the probability of a binary response based on one or more variables (features) .

Input:

The training dataset T = $\{(\boldsymbol{x}^1, \boldsymbol{y}^1), (\boldsymbol{x}^2, \boldsymbol{y}^2), \ldots, (\boldsymbol{x}^m, \boldsymbol{y}^m)\}$

Output:

The weight vector $\mathbf{W} = (\omega_1, \omega_2 \ldots, \omega_n, \mathrm{b})$; the trained model $f(\boldsymbol{x}) = \frac{1}{1+e^{-\mathbf{W}^T\boldsymbol{x}}}$

When $\mathbf{W}^T\boldsymbol{x} > 0$, $f(\boldsymbol{x}) \approx 1$, and $\mathbf{W}^T\boldsymbol{x} < 0$, $f(\boldsymbol{x}) \approx 0$

Algorithms for Modeling Attacks

Evolution Strategies

A evolution strategies (ES) is an optimization technique based on the idea of evolution.

ES steps:

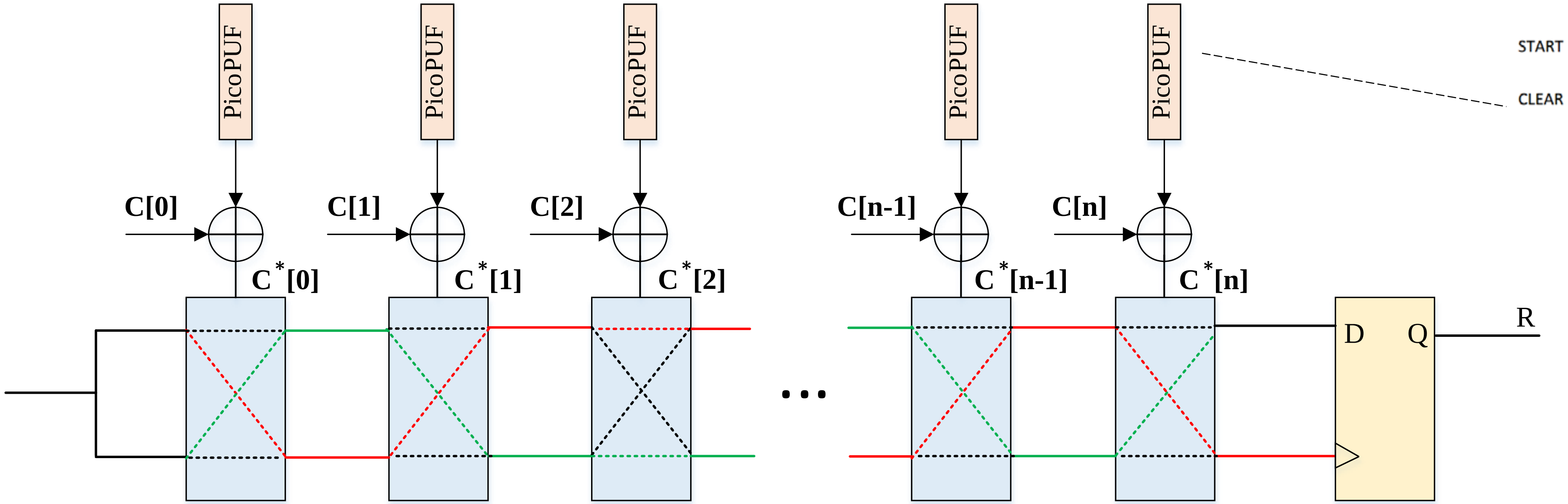
1. Generate the offspring
2. Evaluate the offspring according to the parents
3. Select the fittest offspring
4. Repeat until the termination criterion is met

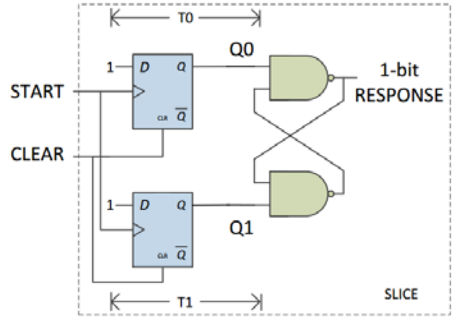
CMA-ES* stands for Covariance Matrix Adaptation-Evolution Strategy. Compared with ES, it’s an advanced method by introducing the covariance matrix adaptation.

CMA-ES can solve many problems and better than LR in some situation; however, it is more time consuming compared with LR.

* N. Hansen, “The CMA evolution strategy: a comparing review,” *Towards a new evolutionary computation*, pp. 75-102, 2006.

Design of Multi-PUF (MPUF)



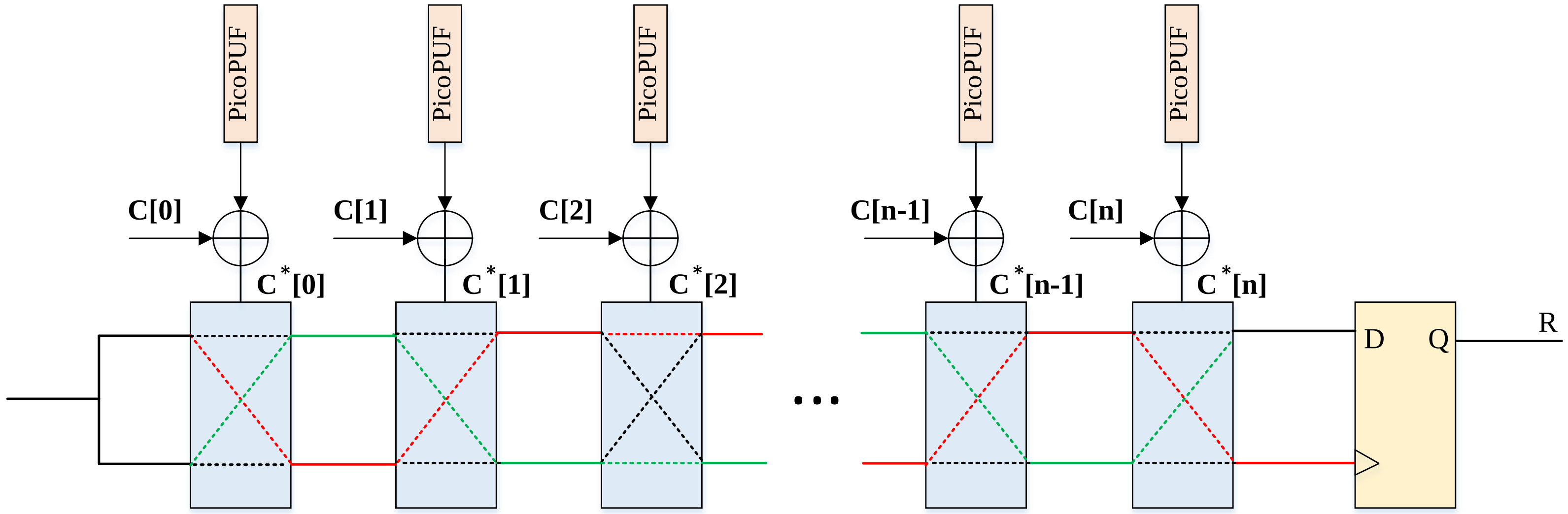


PicoPUF*:
- ➢ Weak PUF
- ➢ High reliability
- ➢ High uniqueness

- ➢ The proposed 1-bit MPUF design is composed of *n* PicoPUF design and a *n*-stage Arbiter PUF design.
- ➢ The response of ***i*** th PicoPUF is XORed with the challenge bit C[***i***] to mask the original challenge bit and a new challenge bit C [***i***] is generated.

* C. Gu, N. Hanley, and M. O'neill, "Improved reliability of FPGA-based PUF identification generator design," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 10, pp. 20:1-20:23, May 2017.

Design of Multi-PUF (MPUF)



For every two stages:

$$\varphi(c_{k-1}^*) \cdot \omega_{k-1} + \varphi(c_k^*) \cdot \omega_k$$
$$= \left(\prod_{i=k-1}^{n}(1-2c_i^*)\right) \cdot \omega_{k-1} + \left(\prod_{i=k}^{n}(1-2c_i^*)\right) \cdot \omega_k$$
$$= \left(\prod_{i=k}^{n}(1-2c_i^*)\right) \cdot [\omega_{k-1} - 2 \cdot c_{k-1} \oplus x_{k-1} + \omega_k]$$

a non-linear formula

1. $\mathbf{R} = \begin{cases} 0 & \Delta < 0 \\ 1 & \Delta \geq 0 \end{cases}$

2. $\Delta = \boldsymbol{\varphi}(\boldsymbol{C}^*) \cdot \mathbf{W}$
$= \varphi(c_1^*) \cdot \omega_1 + \varphi(c_2^*) \cdot \omega_2 + \cdots + \varphi(c_n^*) \cdot \omega_n + \omega_{n+1}$

3. $\mathbf{W} = (\omega_1, \omega_2 \ldots, \omega_n, \omega_{n+1})$

4. $\mathbf{C} = (c_1, c_2 \ldots, c_n)$, $\mathbf{C}^* = (c_1^*, c_2^* \ldots, c_n^*)$

$\mathbf{c_i^*} = c_i \oplus x_i$

$\boldsymbol{\varphi}(\boldsymbol{C}^*) = (\varphi(c_1^*), \varphi(c_2^*) \ldots, \varphi(c_n^*), 1)$

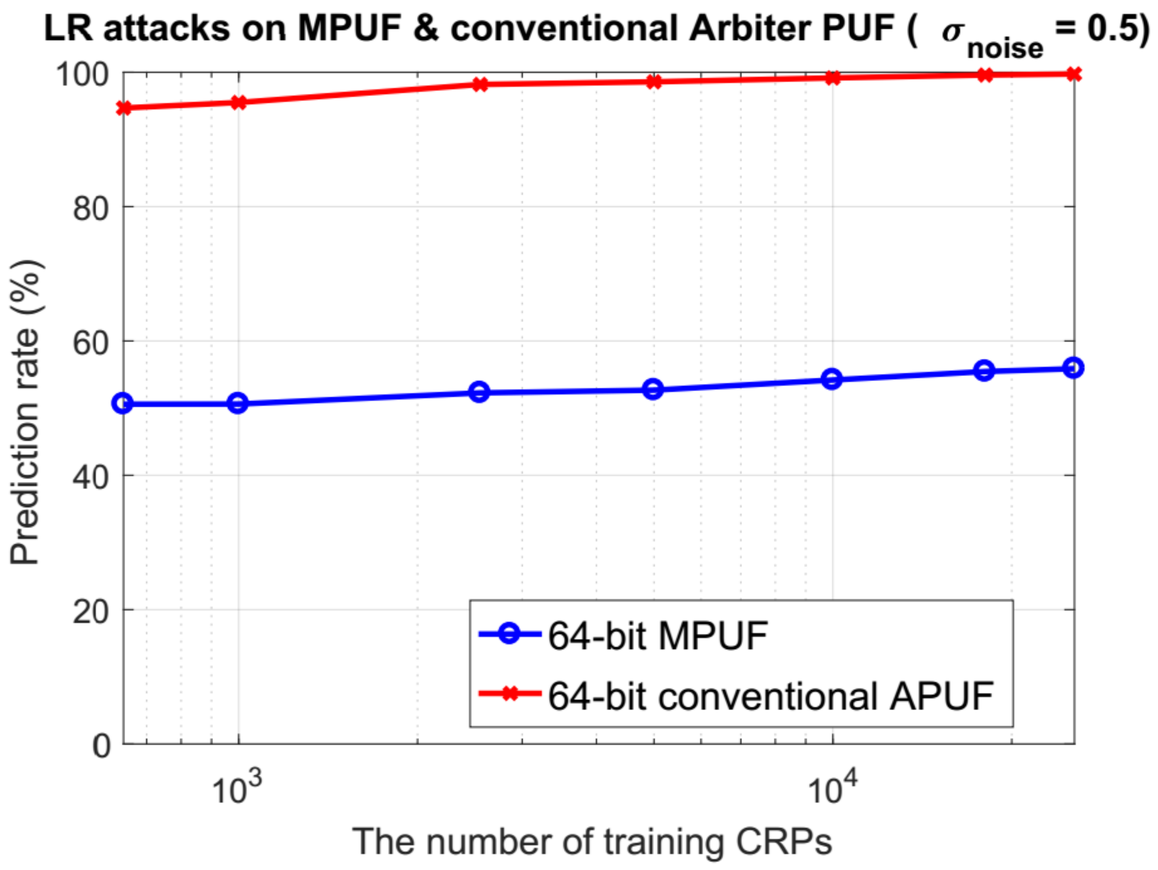
$\varphi(c_i^*) = \prod_i^n (1 - 2c_i^*)$, $c_i^* \in \{0,1\}$

R : response
W: delay parameters of Arbiter PUF
C : challenge
x_i: the output of the i^{th} picoPUF

Analysis of Multi-PUF

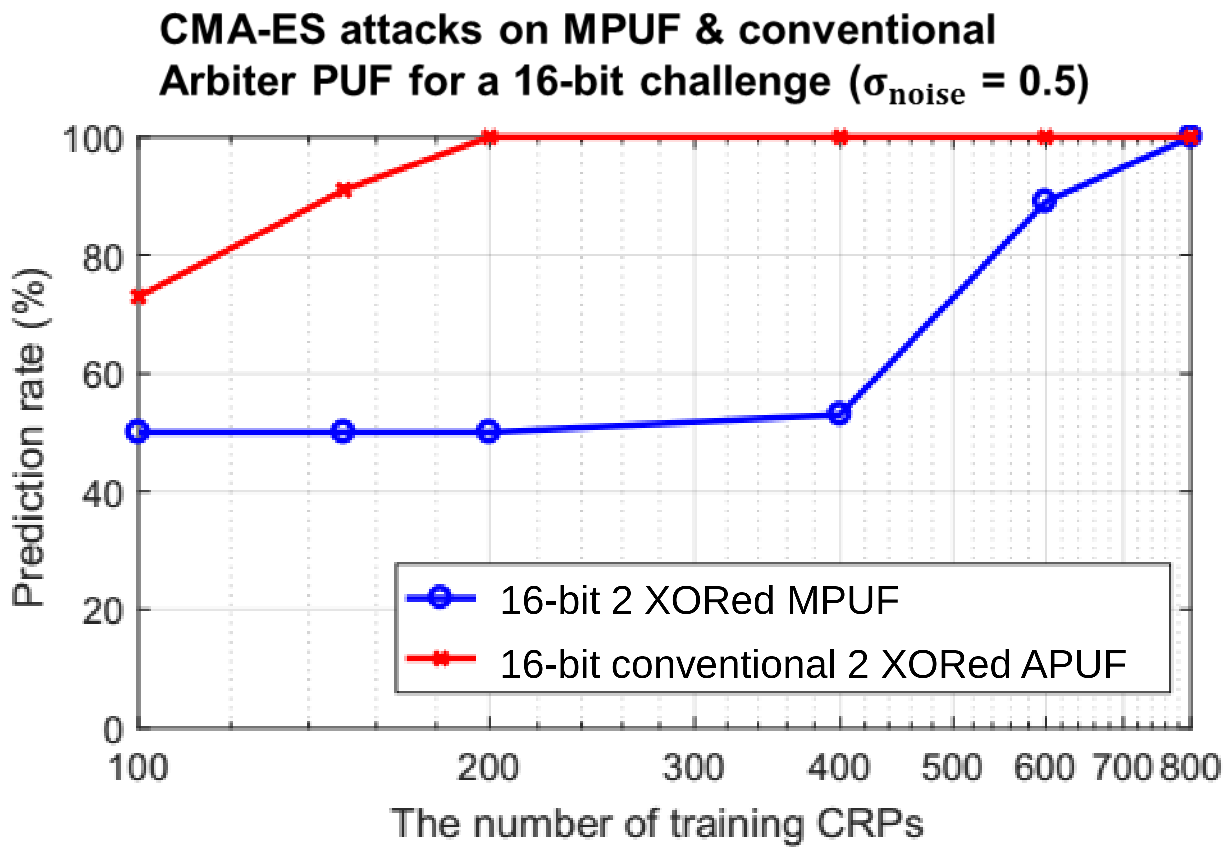
- The conventional Arbiter PUF is **an additive linear function**. The modeling attacks can easily break it by building linear models.
- Obfuscating CRPs is an efficient method to make mathematical modeling more complex.
- The MPUF demonstrates **higher complexity** than the conventional APUF since the outputs of PicoPUF designs are **obfuscated** and **masked**.

LR Attacks on Multi-PUF



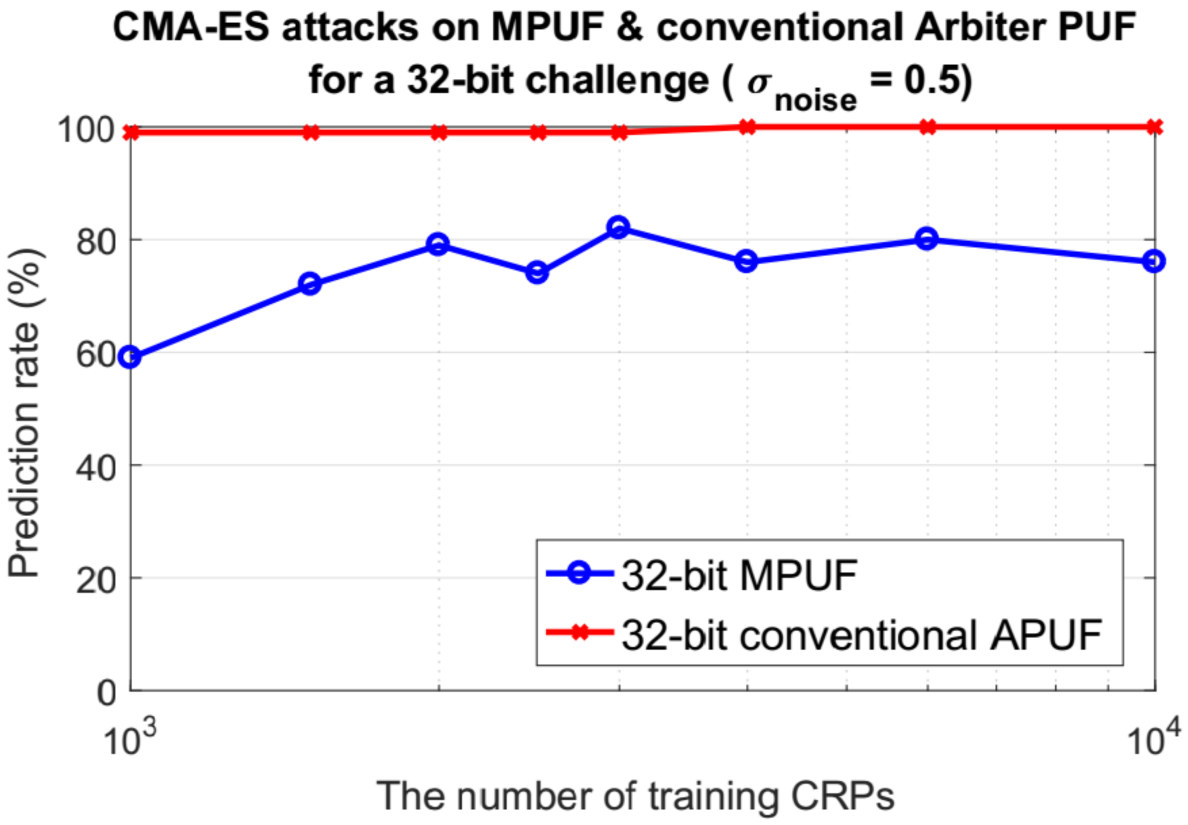
- The size of training sample sets is 3,000, 5,000, 10,000, and 20,000.
- The size of test sample data is the same size as the training one.
- With about 10,000 samples, the prediction rate of Arbiter PUF can reach up to 99%, while prediction rate of MPUF is still around 55%.

CMA-ES Attacks on Multi-PUF



- Using CMA-ES attacks, the conventional 16-bit APUF can be successfully predicted by using 200 training samples
- At least 800 samples are needed to predict the proposed MPUF design.

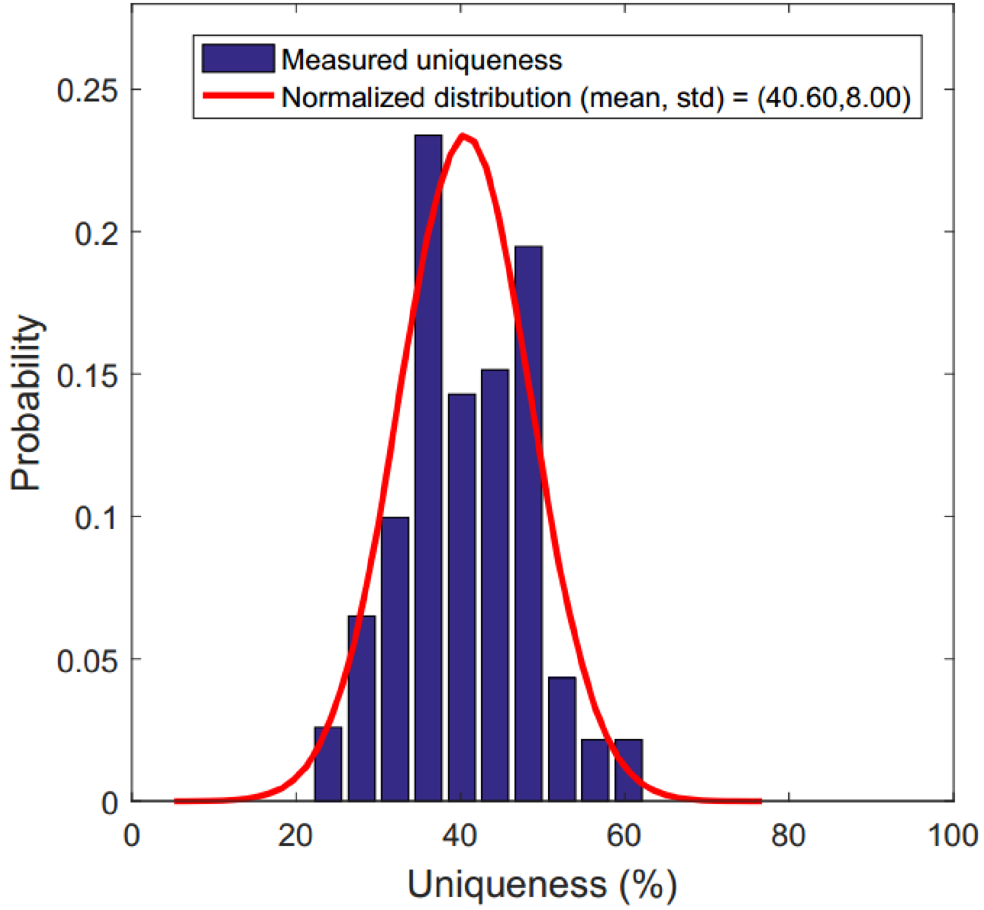
Machine Learning Attack on Multi PUF



- For a 32-bit design, the prediction rate is less than 80% with a large sample set of 10,000 CRPs.
- This means MPUF design will be significantly harder to attack than the conventional APUF for larger number of CRPs.

Performance Evaluation

Uniqueness



- Its empirical **mean** of MPUF: 40.6%

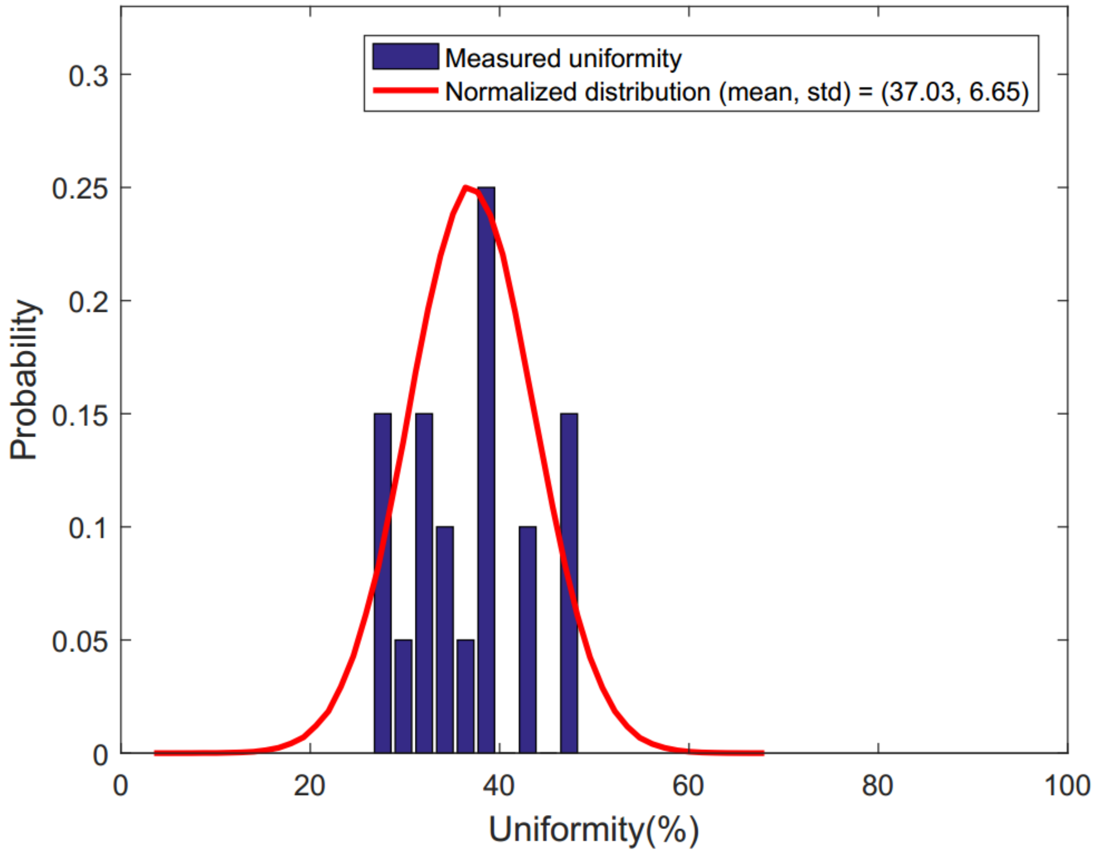
  Compared to the uniqueness results achieved by previous work on multi-PUF*: 5.44%~10.82%

- Its **standard deviation** (STD): 8 %

* D. P. Sahoo, S. Saha, D. Mukhopadhyay, R. S. Chakraborty, and H. Kapoor, “Composite PUF: A new design paradigm for physically unclonable functions on FPGA,” in *Proc. IEEE HOST*, pp. 50-55, May 2014.

Performance Evaluation

Uniformity



- Its empirical **mean** of MPUF: 37.03%

  The result is similar as the uniformity result of previous work on multi-PUF*

- Its **standard deviation** (STD): 6.65 %

* D. P. Sahoo, S. Saha, D. Mukhopadhyay, R. S. Chakraborty, and H. Kapoor, Composite PUF: A new design paradigm for physically unclonable functions on FPGA," in *Proc. IEEE HOST'*, pp. 50-55, May 2014.

Conclusion

- The proposed MPUF design uses a Weak PUF to obfuscate the challenge of a Strong PUF to resist to modeling attacks.
- The MPUF shows good resistance to the LR attack compared with the conventional Arbiter PUF design.
- Although the MPUF can be successfully predicted for designs with small bit-width by using CMA-ES, it is more difficult compared with conventional Arbiter PUF.
- The proposed MPUF design has good uniqueness and uniformity results.

THANKS !

Any Questions?