Tutorial on Physically Unclonable Functions

Hardware Security (NWI-IMC065)

Durba Chatterjee

3 December 2024

Radboud University

Tutorial on PUFs

REQUIREMENTS:

NEED TO INSTALL PYTHON PACKAGE **pypuf**

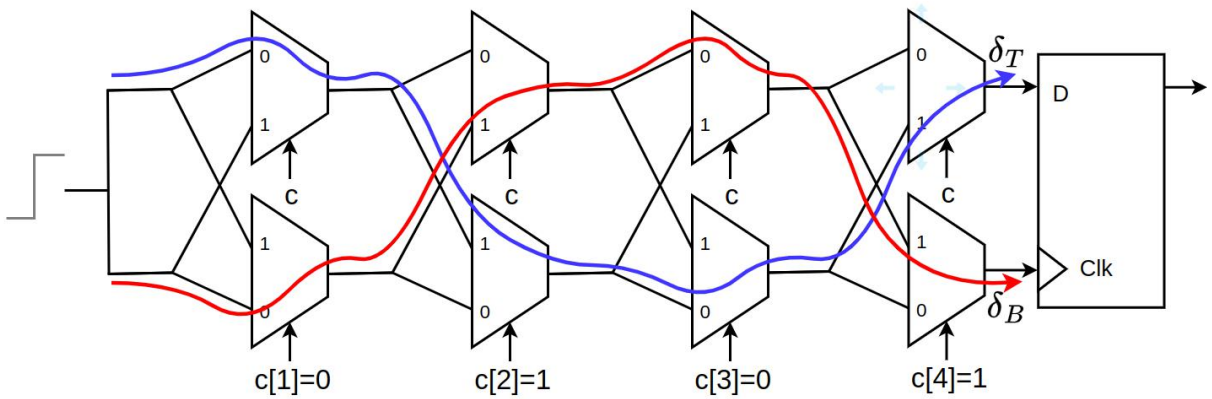
Command for Linux: pip3 install pypuf

Can also download from Github. *Link*: https://github.com/nils-wisiol/pypuf

TUTORIAL GOALS:

- Simulation of Silicon PUF in software
- Computation of PUF Quality Metrics
  - Computation of Inter-Hamming and Intra-Hamming distribution of Silicon PUFs
- Understand impact of noise on PUF behaviour

Recap: Modeling Attack on APUF



c = (c[1], c[2], c[3], c[4])

$$\delta_t(i+1) = \frac{1+c_{i+1}}{2}(p_{i+1} + \delta_t(i)) + \frac{1-c_{i+1}}{2}(s_{i+1} + \delta_b(i))$$

$$\delta_b(i+1) = \frac{1+c_{i+1}}{2}(q_{i+1} + \delta_b(i)) + \frac{1-c_{i+1}}{2}(r_{i+1} + \delta_t(i))$$

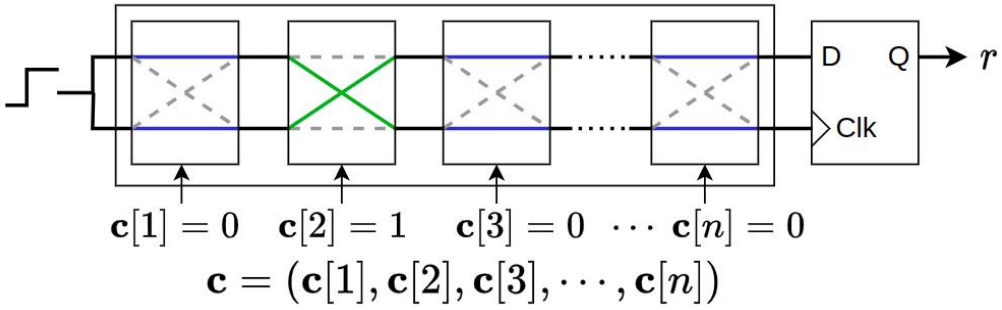
$$\Delta(i+1) = c_{i+1}.\Delta(i) + \alpha_{i+1}.c_{i+1} + b_{i+1}$$

$$\alpha_i = \frac{(p_i - q_i) + (r_i - s_i)}{2} \quad \text{and} \quad \beta_i = \frac{(p_i - q_i) - (r_i - s_i)}{2}$$

$$p_k = \prod_{i=k+1}^{n} c_i \quad k = 0, 1 \cdots, n-1 \quad p_n = 1$$

$$\begin{aligned}\Delta(n) &= \alpha_1 p_0 + (\alpha_2 + \beta_1)p_1 + \cdots + (\alpha_n + \beta_{n-1})p_{n-1} + \beta_n p_n \\ &= \langle \mathbf{w}, \phi \rangle\end{aligned}$$

PyPUF Simulation



Arbiter PUF

- Uses additive delay model to simulate APUF.
- Weights corresponding to delays α_i sampled from Gaussian distribution $\alpha_i \sim N(0,1)$

$$p_k = \prod_{i=k+1}^{n} \mathbf{c}[i] \quad k = 0, 1, \cdots, n-1, \qquad p_n = 1$$

$$\begin{aligned}\Delta(n) &= \alpha_1 p_0 + (\alpha_2 + \beta_1)p_1 + \cdots + (\alpha_n + \beta_{n-1})p_{n-1} + \beta_n p_n \\ &= \langle \mathbf{w}, \phi \rangle\end{aligned}$$

Reference: https://pypuf.readthedocs.io/en/latest/simulation/delay.html

Performance Metrics

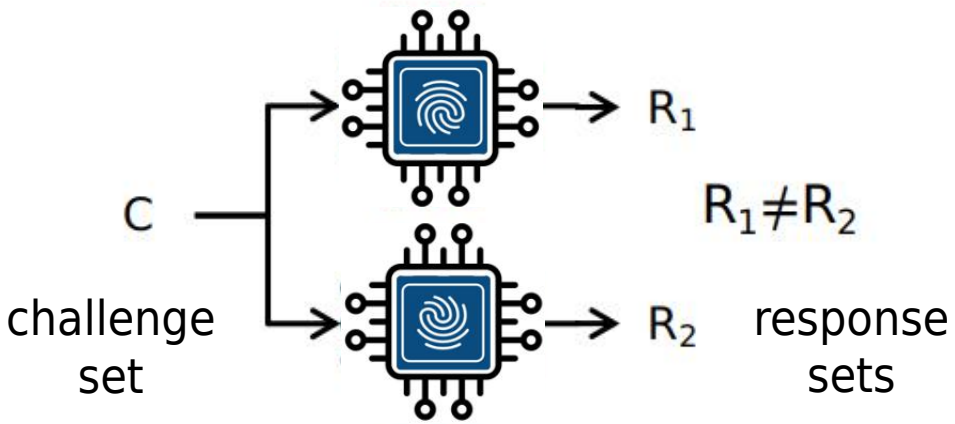
Uniformity: For a set of uniformly chosen challenges, the corresponding responses should have a uniform distribution of 0s and 1s.

$$\text{Uniformity} = \frac{\sum_{i=1}^{N} r_i}{N}$$

Ideal value = 50% or 0.5

Uniqueness

Responses of two or more instances are to be statistically independent over a set of challenges.



Also known as,
Inter Hamming distance

Metric Computation: Hamming distance of responses between all possible pairs of instances.

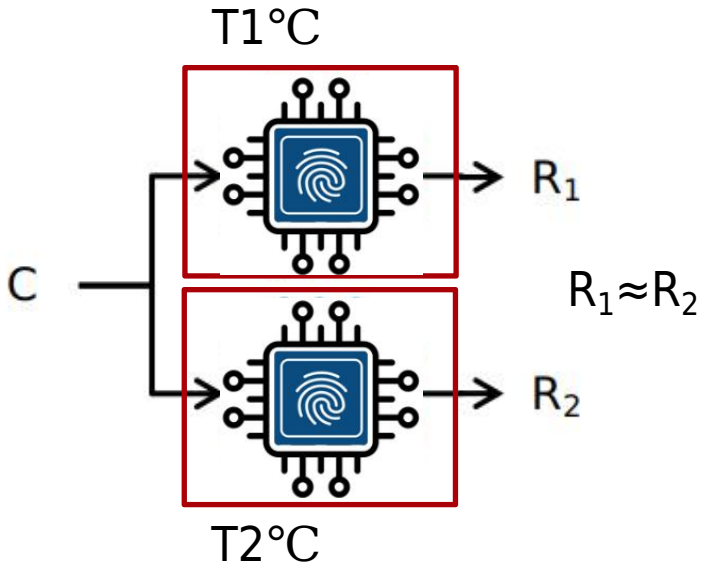
For k randomly chosen instances from the same PUF family, the uniqueness over N challenges chosen uniformly at random

$$\text{Uniqueness} = \frac{\sum_{i=1}^{k-1}\sum_{j=i+1}^{k}\frac{HD(r_{1,i},r_{2,i})}{N}}{k(k-1)}$$

Ideal Value = 0.5

Reliability

Reliability: The PUF functionality should be repeatable at different instants of time under varying environmental conditions (temperature, humidity). Also known as Intra Hamming distance.



Metric computation: Multiple measurements of responses under same and(or) different environmental conditions.

Compute Hamming distance of each measurement with a reference measurement.

(Details in Tutorial on PUFs)

THANK YOU