Arbiter PUF Faults: Impact, Testing, and Diagnosis

PUF = Physically Unclonable Function (hardware security primitive)


Natasha Devroye and Wenjing Rao, University of Illinois at Chicago

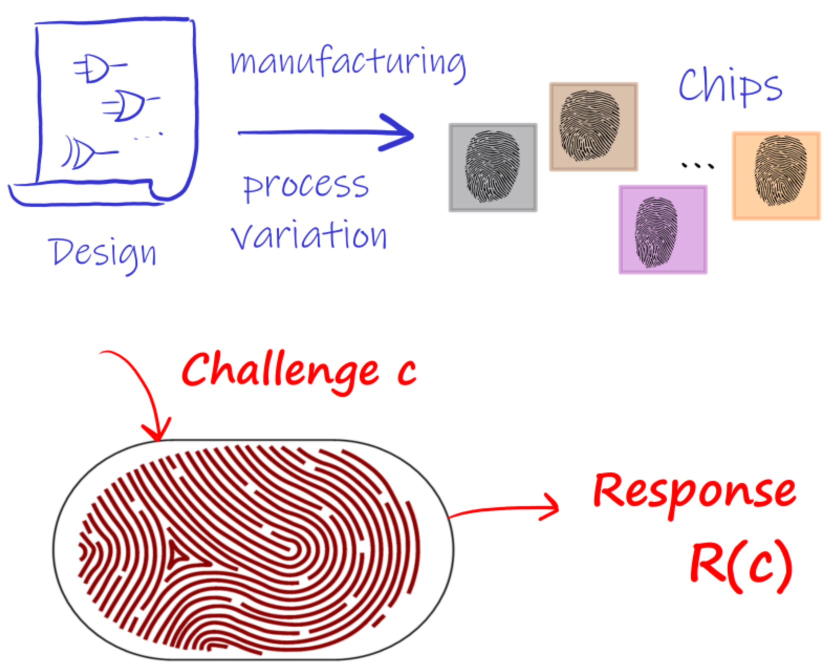
Award # 1909547

CIF: Small: Analytically Predicting Strong PUF Responses from Few Known CRPs

Y. Wei, T. Fox, V. Dumoulin, W. Rao and N. Devroye ***"Faults in Arbiter PUFs: Impact, Testing, and Diagnosis"*** *Design, Automation and Test in Europe Conference (DATE), March 2022.*

Physically Unclonable Functions (PUFs) in General

→ Hardware security primitives offering a unique "fingerprint" for each chip.



Physically Unclonable Function
$\mathbf{c} \in \{0,1\}^n \to R(\mathbf{c}) \in \{\pm 1\}$

Input: Challenge
$\mathbf{c} := \{c_1, c_2, ..., c_n\} \in \{0,1\}^n$

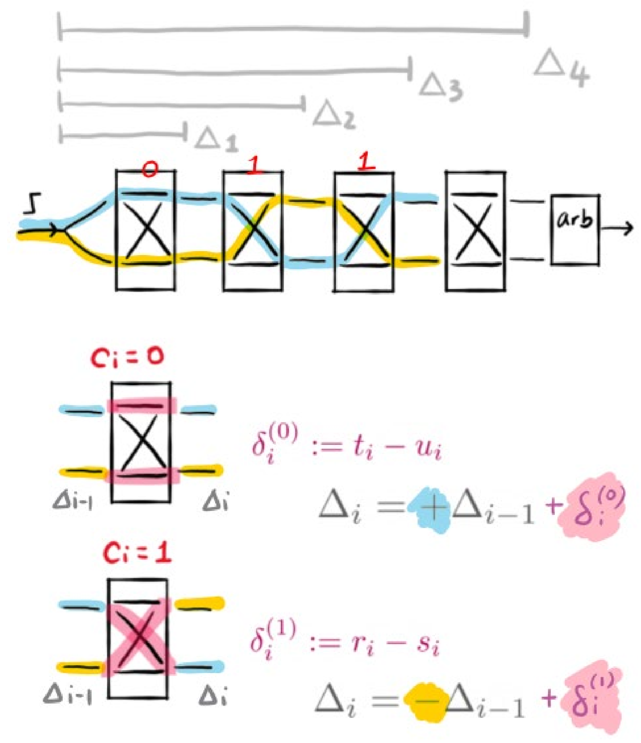
Output: Response
$R(\mathbf{c}) \in \{\pm 1\}$

Strong PUFs
- Hardware cost: $O(n)$
- Truth table size: $O(2^n)$

PUFs: Same design + manufacturing process variation = unique function per chip.

Math Model for APUF

→ Recursive notation of $\Delta_n(\mathbf{c})$ as a function of selected $\delta_i^{(x)}$ delay differences.



Challenge
$\mathbf{c} := \{c_1, c_2, ..., c_n\} \in \{0,1\}^n$

Delay difference elements:
$\delta_i^{(0)} := t_i - u_i$ (selected if $c_i = 0$)
$\delta_i^{(1)} := r_i - s_i$ (selected if $c_i = 1$)

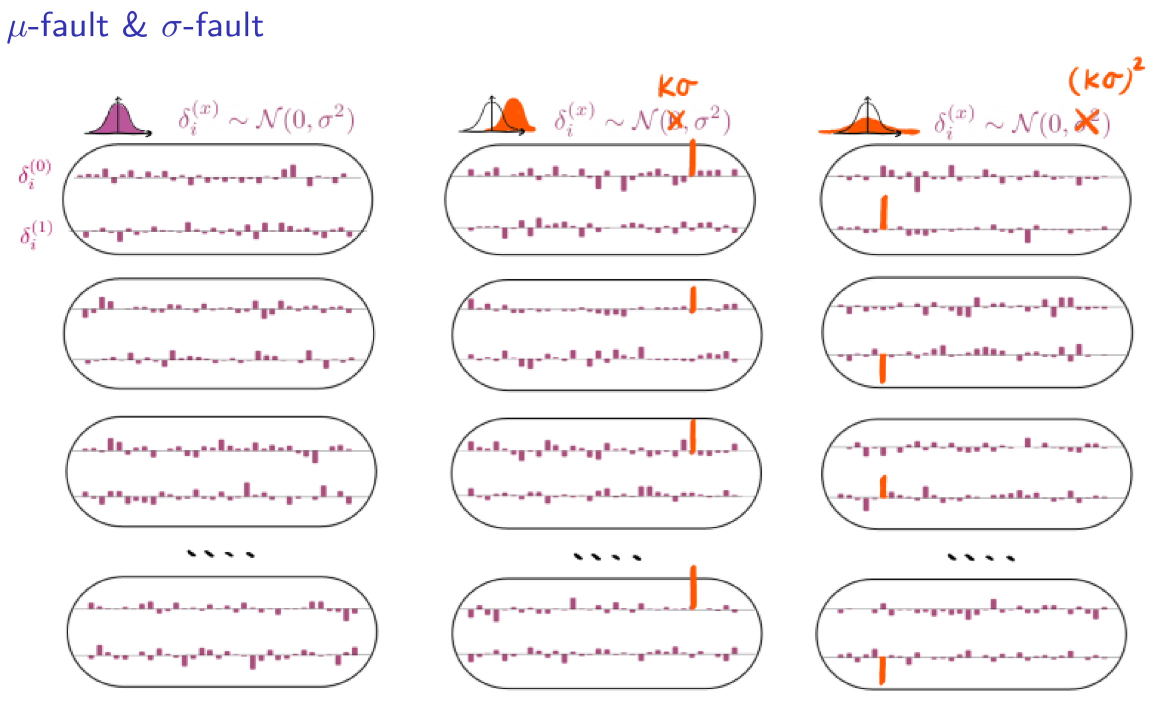
Recursive accumulated delay difference:
$$\Delta_i(\mathbf{c}) = \begin{cases} +\Delta_{i-1}(\mathbf{c}) + \delta_i^{(0)}, c_i = 0 \\ -\Delta_{i-1}(\mathbf{c}) + \delta_i^{(1)}, c_i = 1 \end{cases}$$

Response
$R(\mathbf{c}) := \text{sign}(\Delta_n(\mathbf{c})) \in \{\pm 1\}$

Challenge: how to find ``abnormal'' elements?

Fault Models for APUF Production

→ μ-fault & σ-fault

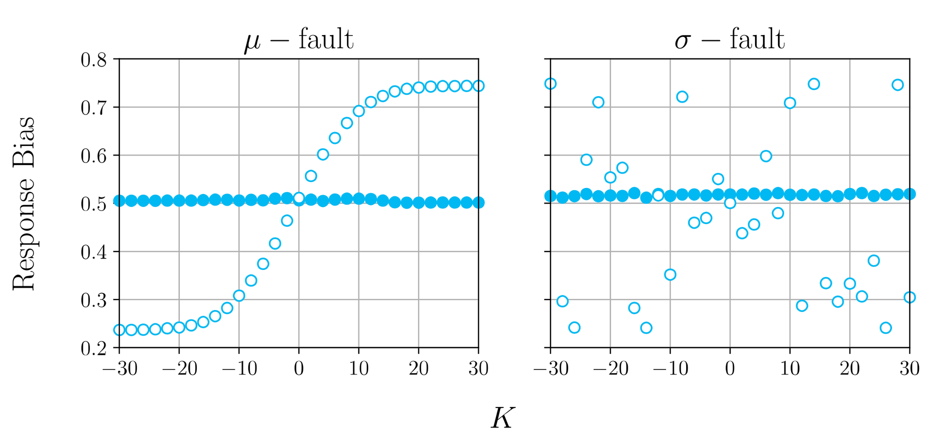


Normal δs
$\delta_i^{(x)} \sim \mathcal{N}(0, \sigma^2)$

Abnormal δ with μ-fault
$\delta_j^{(y)} \sim \mathcal{N}(K\sigma, \sigma^2)$

Abnormal δ with σ-fault
$\delta_j^{(y)} \sim \mathcal{N}(0, (K\sigma)^2)$

Why? Abnormal elements affect bias and uniqueness!

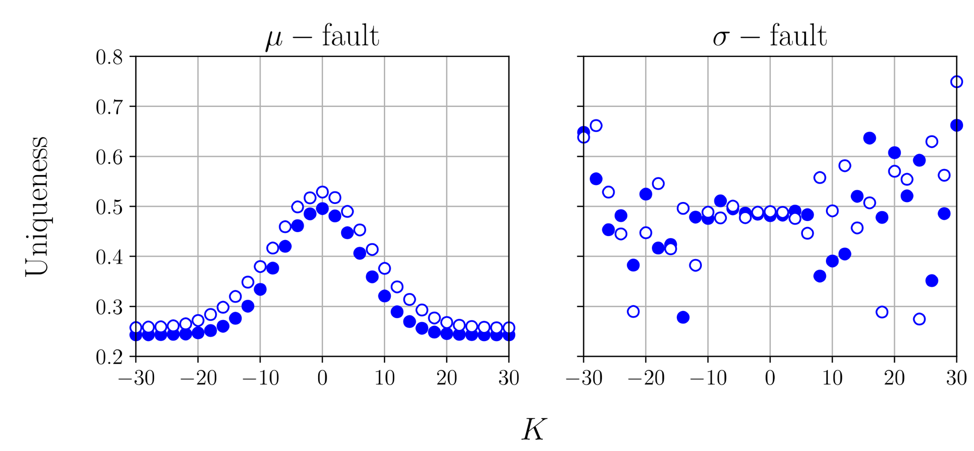


Normal δs
$\delta_i^{(x)} \sim \mathcal{N}(0, \sigma^2)$

Abnormal δ with μ-fault
$\delta_j^{(y)} \sim \mathcal{N}(K\sigma, \sigma^2)$

Abnormal δ with σ-fault
$\delta_j^{(y)} \sim \mathcal{N}(0, (K\sigma)^2)$

● : at stage $i \neq n$
○ : at stage $i = n$



Normal δs
$\delta_i^{(x)} \sim \mathcal{N}(0, \sigma^2)$

Abnormal δ with μ-fault
$\delta_j^{(y)} \sim \mathcal{N}(K\sigma, \sigma^2)$

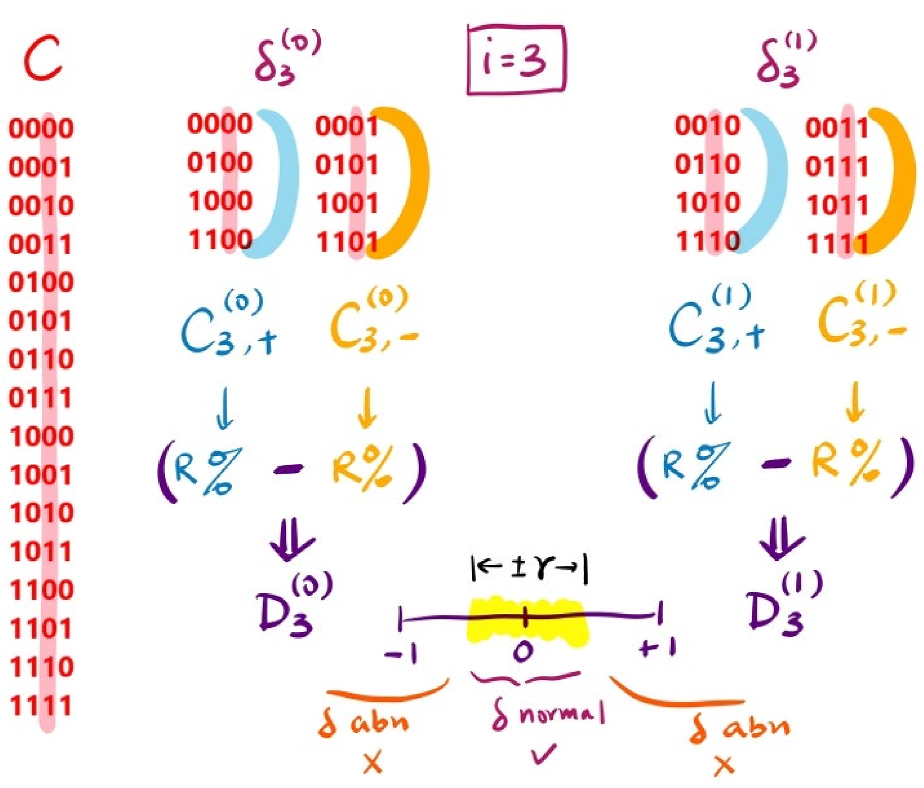
Abnormal δ with σ-fault
$\delta_j^{(y)} \sim \mathcal{N}(0, (K\sigma)^2)$

● : at stage $i \neq n$
○ : at stage $i = n$

Solution: find challenge sets that pinpoint faults

Main Algorithm for Testing and Diagnosis

→ Iteration $i = 3$: form target sets for $\delta_3^{(x)}$, find response biases, use difference scores to evaluate $\delta_3^{(x)}$.



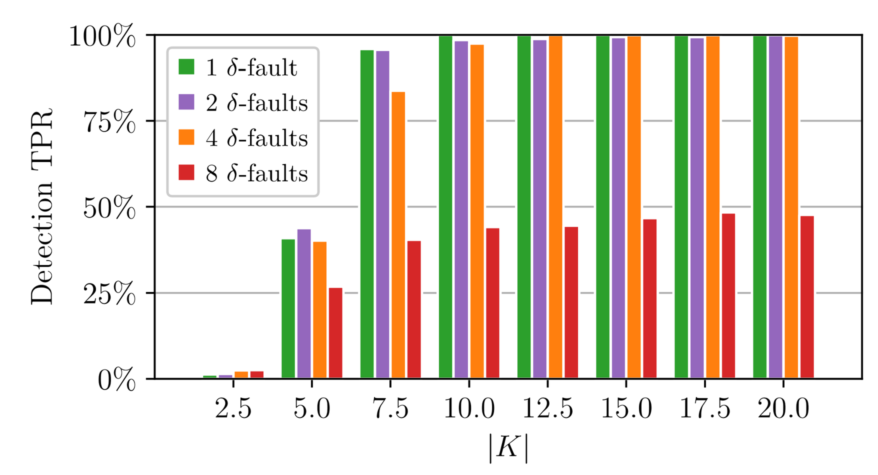
Difference Score
$D_i^{(x)} := B(\mathcal{C}_{i,+}^{(x)}) - B(\mathcal{C}_{i,-}^{(x)}) \in [-1, 1]$

Criteria: choose $\gamma \in [0, 1]$
- $D_i^{(x)} > \gamma \implies \delta_i^{(x)} > T$
- $|D_i^{(x)}| < \gamma \implies |\delta_i^{(x)}| \leq T$
- $D_i^{(x)} < -\gamma \implies \delta_i^{(x)} < -T$

Looping through $i = 1$ to n
Split $\mathcal{C} \to \mathcal{C}_{i,+}^{(0)}, \mathcal{C}_{i,-}^{(0)} \mathcal{C}_{i,+}^{(1)}, \mathcal{C}_{i,-}^{(1)}$
Collect $B(\mathcal{C}_{i,+}^{(x)})$
Compute $D_i^{(x)}$
Diagnose $\delta_i^{(x)}$

Results of Testing: Can bad APUFs be detected?

→ Yes, by the proposed algorithm: $\sim 100\%$ detected for up to 4 abnormal δs per 64-bit APUF with $K > 10$.



True Positive: TP
\# of bad APUFs correctly detected

False Negative: FN
\# of bad APUFs incorrectly identified as good

True Positive Rate: TPR $= \frac{TP}{TP + FN}$
% of bad APUFs detected

Setting: A total of 1000 64-bit bad APUFs with 1, 2, 4, or 8 δ-faults, varying fault intensity K, and randomly selected fault position.

Broader Impact (impact on society – who will care)

- simple, effective tool for manufacturers of PUFs
- Native APUF fault model

Broader Impact (education and outreach)

- ECE 464: Testing and Reliability of Digital System course at UIC
- Women in Engineering Summer Program: Devroye participates yearly as speaker.

Broader Impact and Broader Participation (quantify potential impact)

- Supported 4 excellent REUs, one is co-author on published paper
- Inter-disciplinary project hardware security, testing, and statistics / information theory