CycPUF: Physical Unclonable Function

Michael Dominguez and Amin Rezaei

Introduction

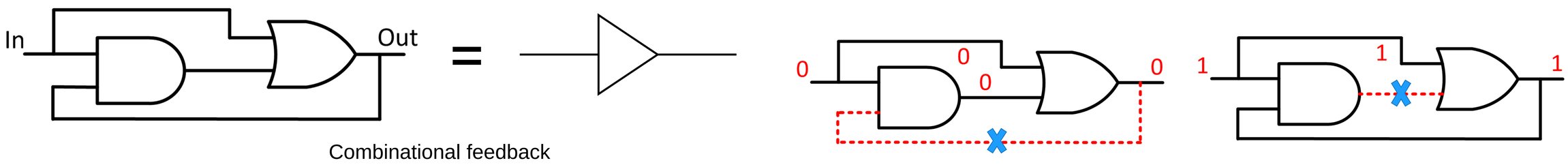
Definition: **Physical Unclonable Functions (PUFs)** leverage manufacturing process imperfections that cause propagation delay discrepancies for the signals traveling along these paths.

Use case: PUFs can be used for **device authentication** and **chip-specific key generation**.

Challenge: Strong PUFs have been shown to be vulnerable to **machine learning modeling attacks**.

Contribution: We propose **CycPUF** by introducing feedback signals into traditional delay-based PUF to give them a wider range of possible output behaviors and thus an edge against modeling attacks.

Cyclic Combinational Circuits

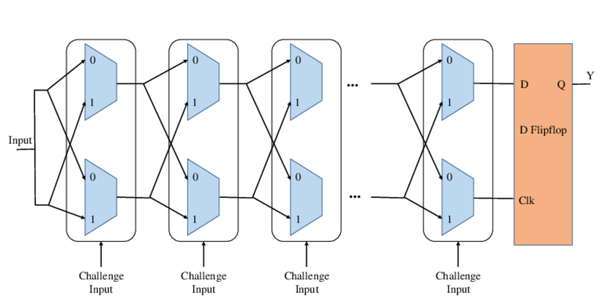


Combinational feedback

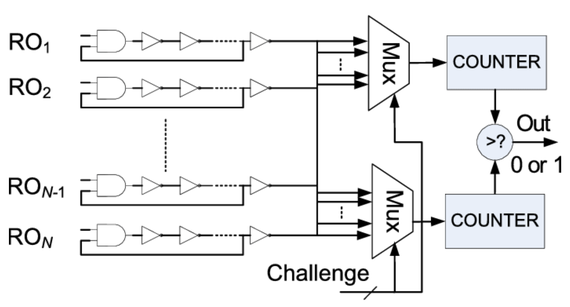
While there is a common perception that combinational circuits must be designed without any loop, cyclic combinational circuits can be designed to provide benefits to hardware security that were previously frowned upon for their difficulty to synthesize and verify using synthesis tools and FPGA emulation.

A PUF Primer

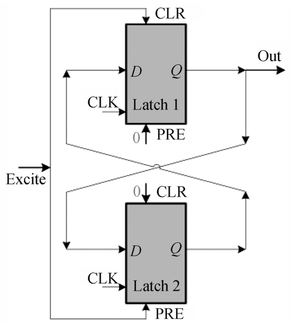
There are various types of traditional delay-based PUFs, such as Arbiter PUF (APUF) [1], Ring Oscillator PUF (ROPUF) [2], and Butterfly PUF (BPUF) [3]. Each type of PUF has its own unique properties and strengths, and the selection of the PUF category depends on the intended application and security requirements.



Arbiter PUF [1]



Ring Oscillator PUF [2]



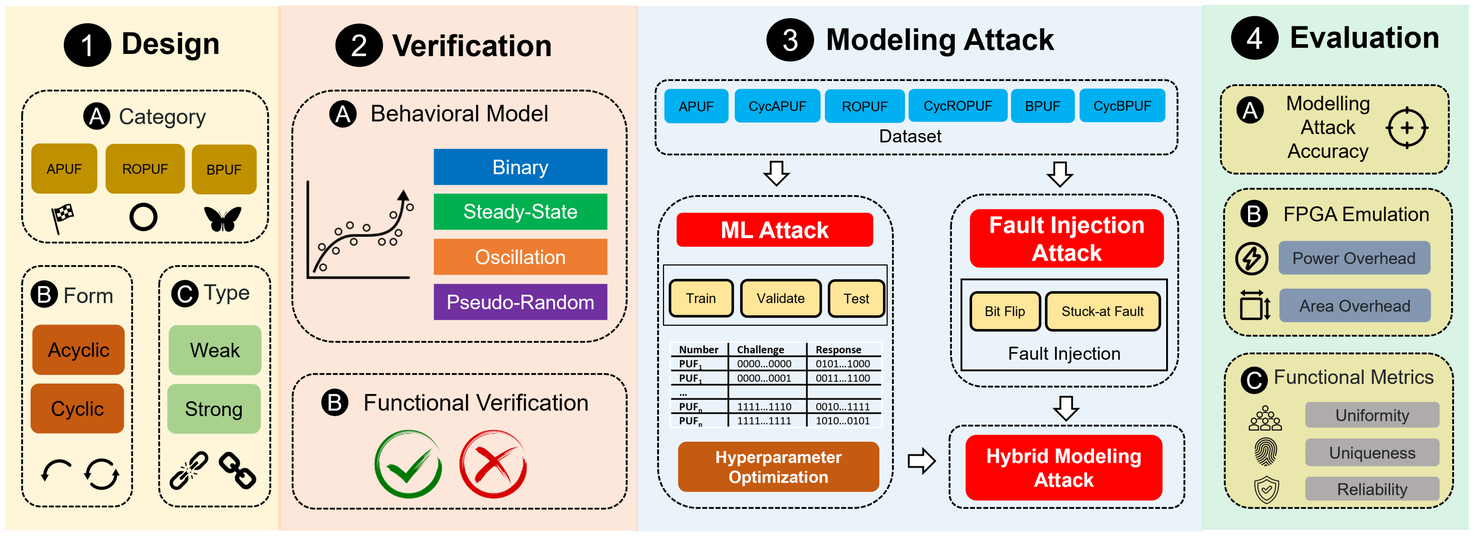
Butterfly PUF [3]

Previous Approaches & Challenges

A deception authentication protocol has been proposed by deceiving the adversary to use a training set dominated by invalid responses [4]. However, finding such a set for each instance of the device is a tedious task for the designer.

Linear-Feedback Shift Registers (LFSR) have been utilized for the purpose of obfuscating the CRPs of the proposed PUFs [5]. By feeding the response into the LFSRs, then using the outputs to obfuscate the current challenge, ML models may be thrown off and rendered inaccurate. Here however, the design increases greatly in complexity. Additionally, only reliable responses are utilizable for obfuscation of challenges.

CycPUF Framework for Developing Cyclic and Acyclic PUFs



CycPUF Behavior

PUF Behavior	Description
Binary	The CycPUF behaves the same as its acyclic counterpart
Steady-State	There are a few outputs before a final response is produced.
Oscillating	The output will switch between two or more responses, creating a discernible pattern in the output.
Pseudo-Random	The output will generate seemingly random response vectors, such that no pattern is recognizable.

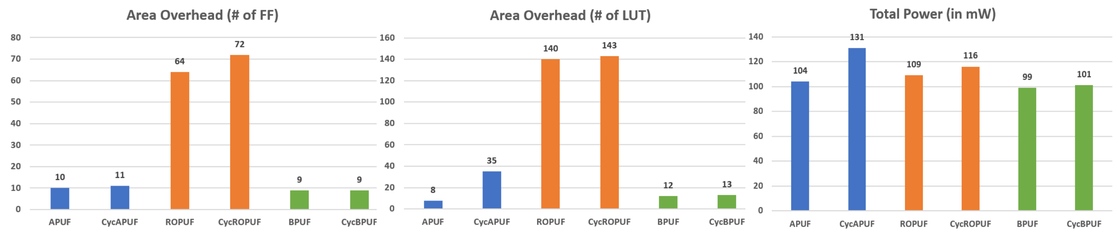
Experimental Results

PUF Design	Challenge Size	# of Training CRPs	Model Accuracy
APUF	64	500,000	99.38%
CycAPUF	64	703,340	59.49%
Faulty CycAPUF	64	500,000	76.27%
ROPUF	64	500,000	78.00%
CycROPUF	64	1,048,576	48.74%
Faulty CycROPUF	64	642,603	50.02%
BPUF	64	500,000	83.32%
CycBPUF	64	938,420	54.76%
Faulty CycBPUF	64	582,645	61.44%

Modeling attack results combining DNN-based modeling attack [6] with fault-injection [7]

PUF Design	Uniqueness	Uniformity	Reliability
APUF	7.55%	55.42%	99.77%
CycAPUF	47.10%	47.30%	98.34%
ROPUF	44.26%	53.81%	99.05%
CycROPUF	50.68%	52.12%	95.18%
BPUF	11.48%	55.04%	97.45%
CycBPUF	53.05%	46.67%	98.62%

Functional metrics results with 4-bit challenge and 4-bit response sizes



Overhead analysis with 4-bit challenge and 4-bit response sizes on FPGA

Takeaways:

- There is clear resistance against modeling attacks by CycPUFs.

- The selection of a PUF design should carefully consider hardware overhead and power consumption trade-offs.

- CycPUFs outperform their acyclic counterparts in uniqueness and inherit the reasonable uniformity and reliability of their acyclic versions.

Conclusion

We introduced CycPUF, a novel lightweight PUF generation framework featuring strong resistance against ML-based and hybrid modeling attacks. This work offers researchers new perspectives on approaching hardware security problems with cyclic combinational circuits and looking into developing synthesis-friendly toolkits for them.

References

[1] S. Hemavathy et al., "Arbiter PUF - A Review of Design, Composition, and Security Aspects," in IEEE Access, 2023.
[2] S. R. Sahoo et al., "A Novel ROPUF for Hardware Security," in VDAT, 2015.
[3] S. S. Kumar et al., "The Butterfly PUF Protecting IP on Every FPGA," in HOST, 2008.
[4] C. Gu et al., "A Modeling Attack Resistant Deception Tech. for Securing Lightweight-PUF-Based Auth.," in IEEE TCAD, 2021.
[5] L. Wu et al., "FLAM-PUF: A Response–Feedback-Based Lightweight Anti-Machine-Learning-Attack PUF," in IEEE TCAD, 2022.
[6] P. Santikellur et al., “Deep Learning based Model Building Attacks on Arbiter PUF Compositions,” in Cryptology Archive, 2019.
[7] S. Tajik et al., "Laser Fault Attack on Physically Unclonable Functions," in FDTC 2015.