Effect of Aging on Linear and Nonlinear MUX PUFs by Statistical Modeling

Anoop Koyily, Satya Venkata Sandeep Avvaru, Chen Zhou, Chris H. Kim, Keshab K. Parhi
University of Minnesota, Twin Cities

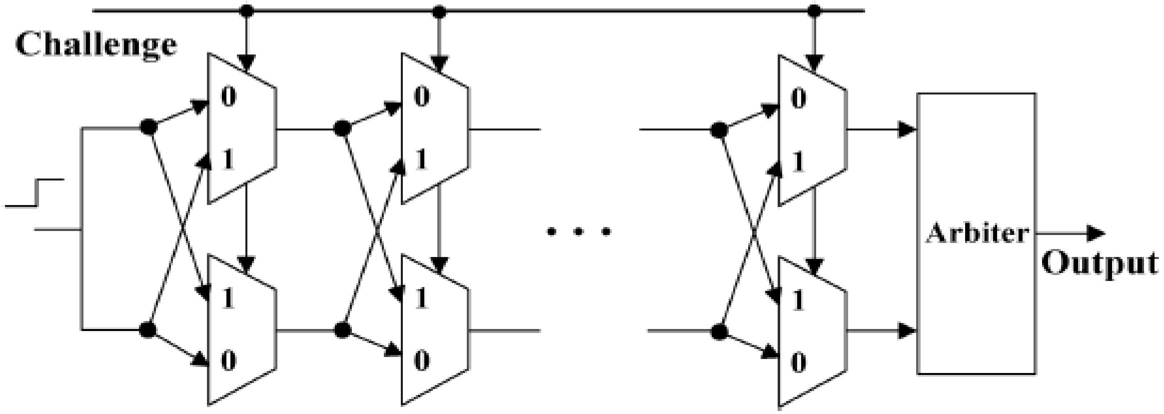
UNIVERSITY OF MINNESOTA
Driven to Discover®

OUTLINE

- MUX PUFs
- Authentication of PUFs from model parameters
- Total delay-difference distribution
- Aging model
  - Delay chain
  - Arbiter
  - Combined Model
- Monte-Carlo simulation
- Aging results
- Improving reliability

Physical unclonable function (PUFs)

- PUFs are hardware circuits that intrinsically store unique signatures without requiring non-volatile RAMs.
- The unique signature is a result of variations in the manufacturing process.
- For an N-stage MUX PUF: N-bit challenge is ***Input*** and 1-bit response is ***Output*****.** A rising clock edge at the input traverses through the delay chain.

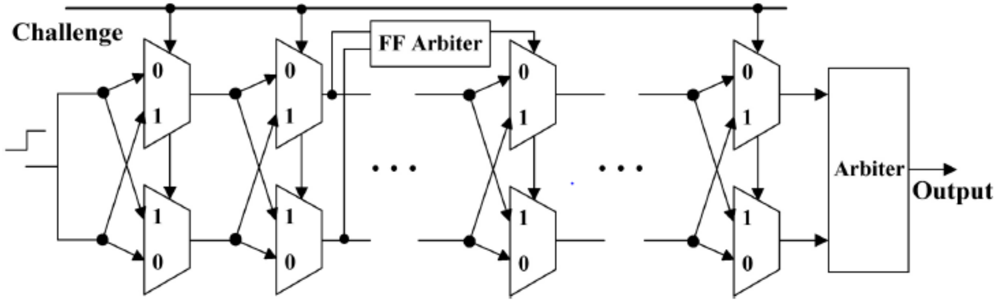


Example of Linear MUX PUF

[1] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, “Silicon physical random functions,” in *Proceedings of the 9th ACM conference on Computer and communications security,* ACM, 2002, pp. 148–160.

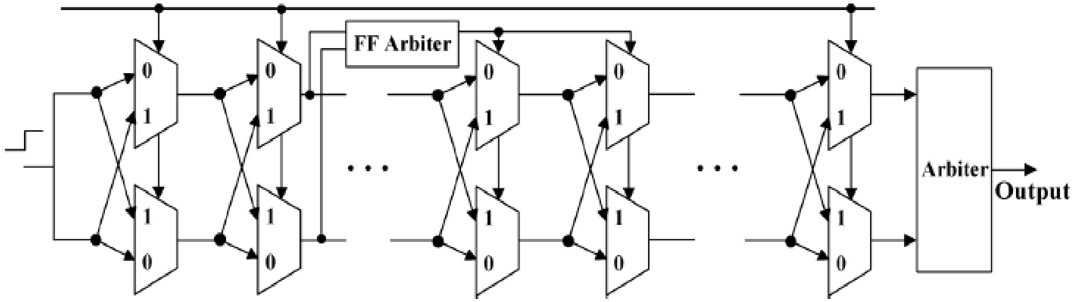
Linear and Nonlinear PUFs

- An N-stage multiplexer (MUX) based PUF has:
  - **Delay chain**: N stages of top and bottom multiplexers.
  - **Arbiter**: A Latch/flip-flop at the end.



Feed-forward PUF

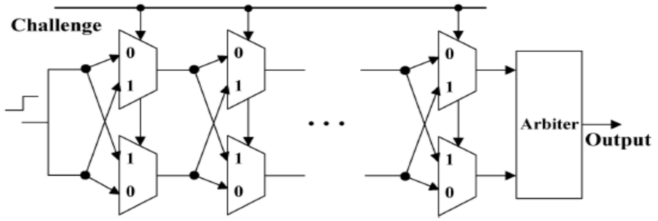
- Other configurations like modified feed-forward and feed-forward are formed by adding intermediate arbiters which generate internal challenge bits.
- This makes the PUF structure non-linear in nature.



Modified feed-forward PUF

[2] Y. Lao and K. K. Parhi, “Statistical analysis of MUX-based physical unclonable functions,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 5, pp. 649–662, 2014.

PUF model for linear MUX PUF



- The response or output can be modelled in terms of the delay-difference of MUX stages.
- Delay-difference of i[th] stage: $\Delta_i = D_i^t - D_i^b \sim N(0, 2\sigma^2)$

where D_i^t, D_i^b are the top and bottom multiplexer delays.

- Response bit, R, for a linear PUF can be decided based on total delay-difference, r_N as:

$$r_N = \sum_{i=1}^{N+1} (-1)^{C_i'} \Delta^i = \sum_{i=1}^{N} (-1)^{C_i'} \Delta^i + \Delta^{arb}$$

, $C_i' = \oplus_{j=i}^{N} C_j$ corresponds to delay chain

$(C_{N+1}' = 0)$ corresponds to arbiter

$$R = sign(r_N) = \begin{cases} 1, & r_N \geq 0 \\ 0, & r_N < 0 \end{cases}$$

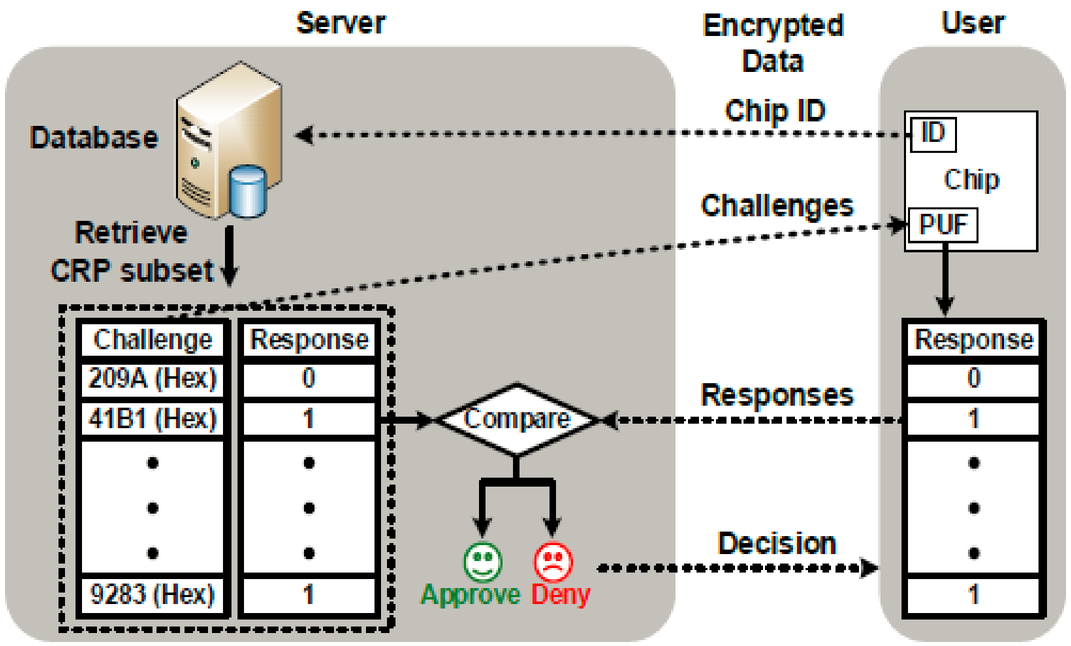
- Delay parameters, Δ^i and Δ^{arb} can be estimated using LMS method described in [3].

[3] S. S. Avvaru, C. Zhou, S. Satapathy, Y. Lao, C. H. Kim, and K. K. Parhi, "Estimating delay differences of arbiter PUFs using silicon data," in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2016, pp. 543–546.

Authentication of PUFs

<u>Chip enrollment phase</u>:
Reference challenge-response pairs (CRPs) are stored as LUT in server.
We **propose** to store **model parameters** instead (i.e., delay-differences). Needs much less area compared to storing a LUT.
<u>Authentication phase</u>:

- Server receives an AUTH request with chip ID from user.
- Selects "random" challenges from database. These are sent to the user and responses are sent back to the server.
- User is **granted access** if the responses from chip match the responses stored/obtained in the server.
- Certain amount of error can be tolerated.



Authentication of Soft-PUFs

Stability of challenges:

- Due to variations by noise, the response to a challenge can vary upon multiple attempts. In such case, we want to classify challenges as *stable* and *unstable* in terms of their soft-response, R_s.

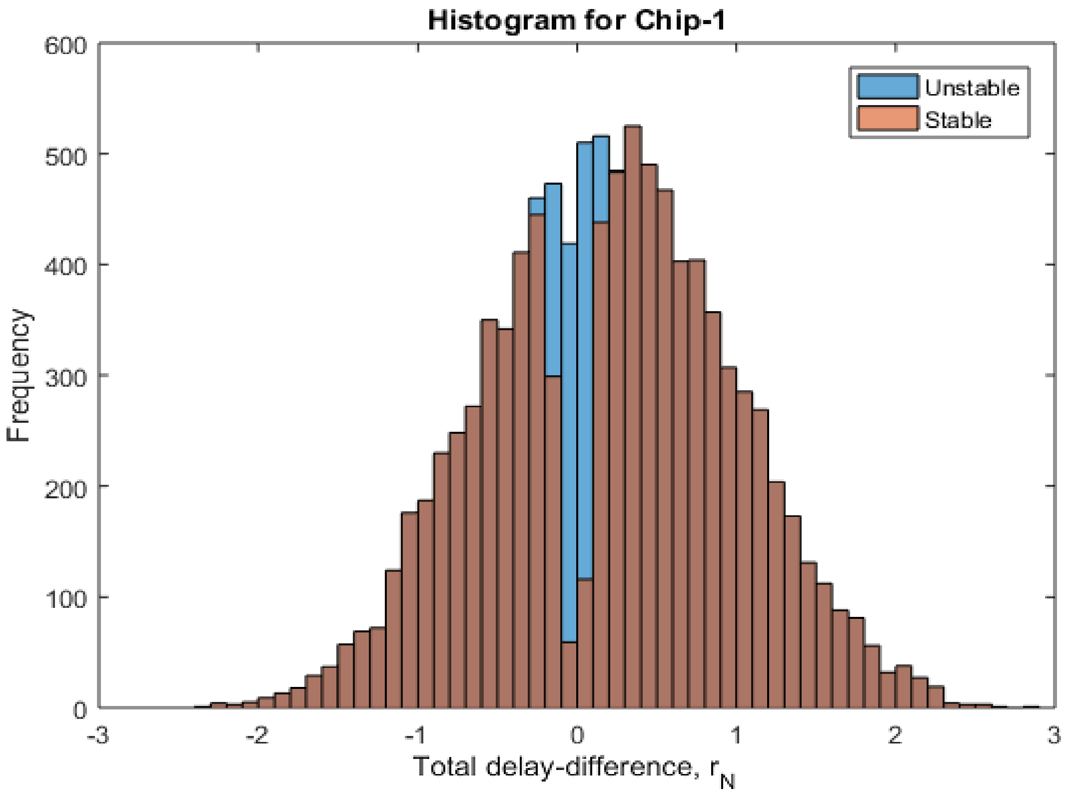
$$R_s = \frac{\#(\text{number of times response bit is 1})}{\text{total measurements}}$$

- Thresholds are defined to determine stability – If $R_s < 0.1$ or $R_s > 0.9$, challenge is termed *stable*, otherwise *unstable*.
- During authentication phase, it is desirable to select challenges that are stable.

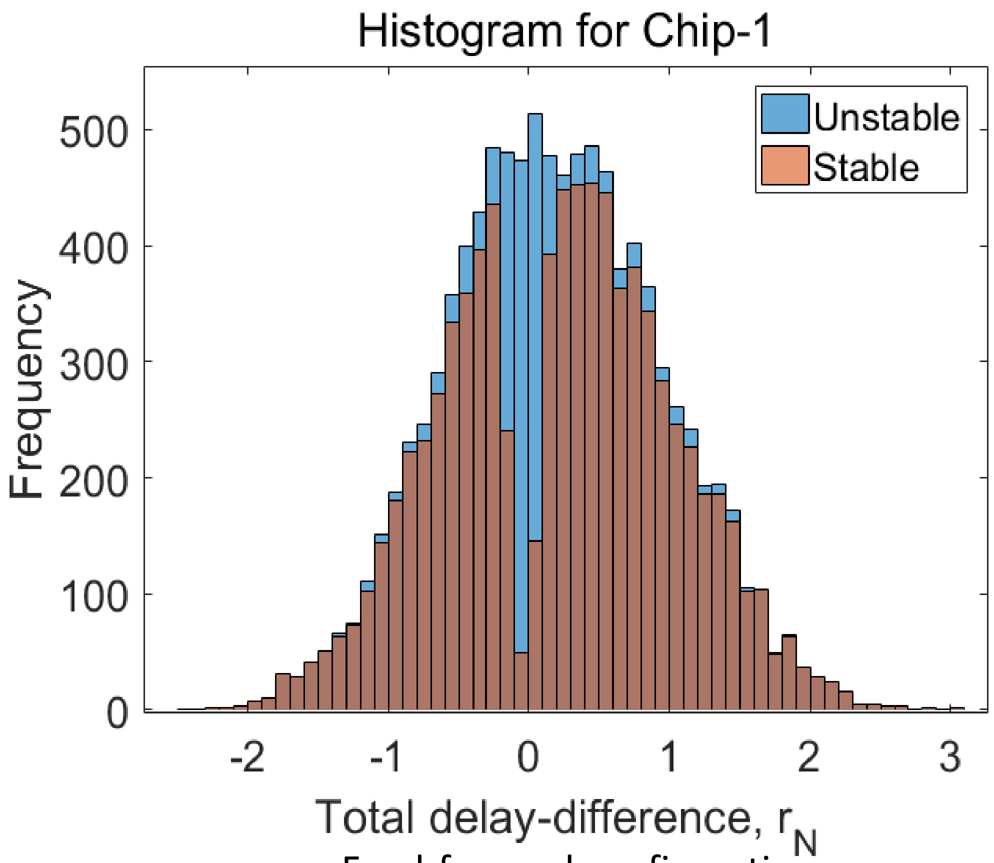
[4] C. Zhou, S. Satapathy, Y. Lao, K. K. Parhi, and C. H. Kim, "Soft response generation and thresholding strategies for linear and feed-forward MUX PUFs," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, ACM, 2016, pp. 124–129.

Total delay-difference distribution

$$r_N = \sum_{i=1}^{N+1} (-1)^{C_i'} \Delta^i = \sum_{i=1}^{N} (-1)^{C_i'} \Delta^i + \Delta^{arb}$$



Linear configuration

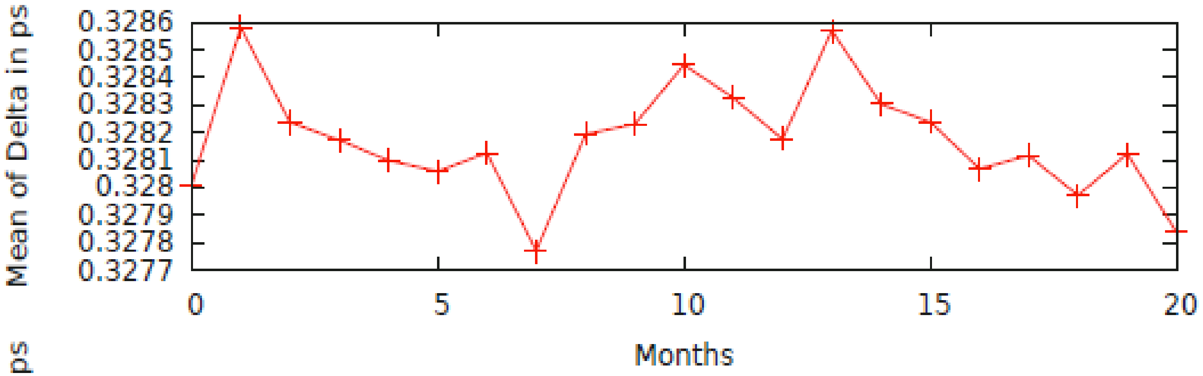


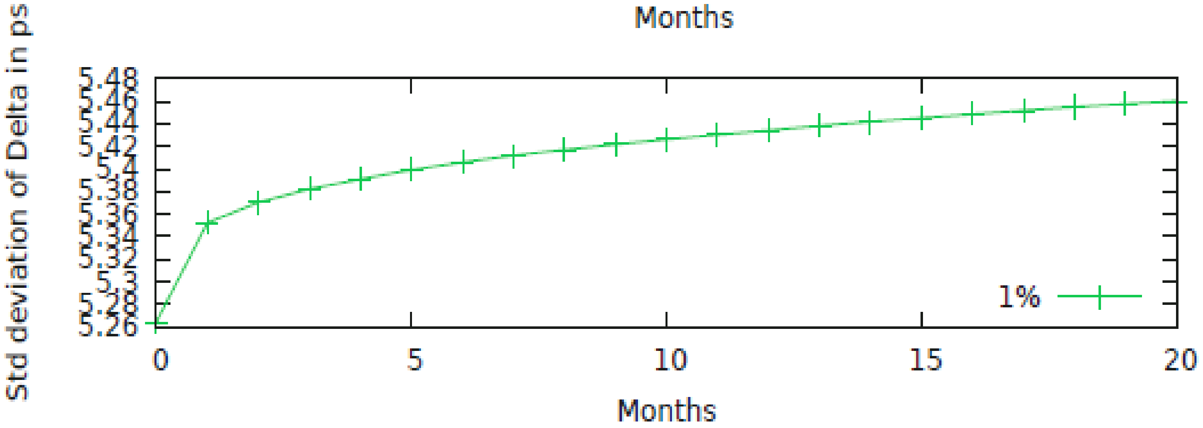
Feed-forward configuration

- % unstable challenges for feed-forward is much higher than linear – for example, 15% vs 11% (for chip-1)
- Standard deviation (σ) of total delay-difference, r_N = 0.77 (for chip-1)

AGING MODEL

- Aging is caused by undesirable changes in hardware structure such as **NBTI (Negative Bias Temperature instability)**, HCI (hot carrier injection) and TDDB (time dependent dielectric breakdown).
- NBTI happens continuously <u>when the circuit is powered on</u>, whereas HCI only when the circuit has some activity.
- NBTI and HCI cause progressive slowdown in hardware and therefore, increase delays of hardware like MUX.
- Work in [5] showed that variance of delay-differences of delay chain increases with aging, whereas mean of delay-difference can increase or decrease.
- However, variations in delay-difference of the delay chain and arbiter delay is modeled in a slightly different manner.

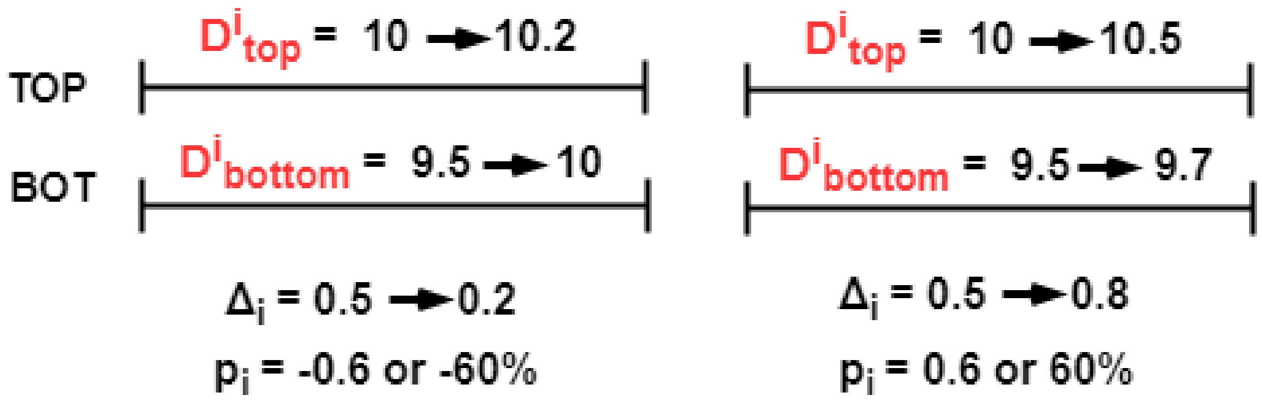


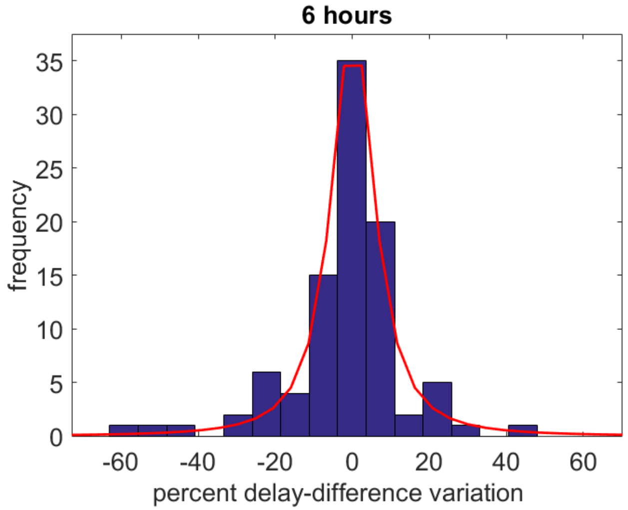


[5] N. Karimi, J.-L. Danger, F. Lozach, and S. Guilley, "Predictive aging of reliability of two delay PUFs," in International Conference on Security, Privacy, and Applied Cryptography Engineering. Springer, 2016, pp. 213–232.

Aging model for delay chain

$$r_N = \sum_{i=1}^{N+1} (-1)^{C_i'} \Delta^i = \sum_{i=1}^{N} (-1)^{C_i'} \Delta^i + \Delta^{arb}$$

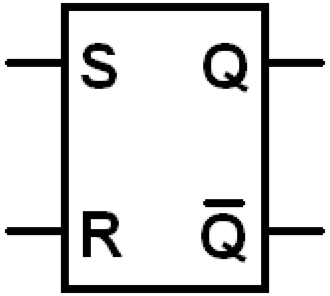




- The delays of multiplexers increase with aging. However, the delay-difference can increase or decrease depending on whether the top or bottom multiplexer increases more.
- The percent delay-difference variation, p_i, is modeled as a Gaussian with zero mean and variance increasing with aging [6].
- New delay-difference is expressed as: $\Delta^i_{aged} = \Delta^i \left(1 + \frac{\Delta^i_{aged} - \Delta^i}{\Delta^i}\right) = \Delta^i \left(1 + p_i\right)$

[6] G. Marsaglia, "Ratios of normal variables and ratios of sums of uniform variables," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 193–204, 1965

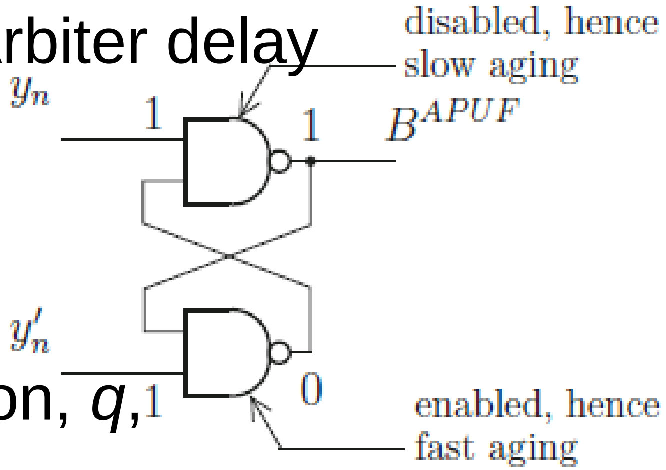
Aging model for arbiter



$$r_N = \sum_{i=1}^{N+1} (-1)^{C'_i} \Delta^i = \sum_{i=1}^{N} (-1)^{C'_i} \Delta^i + \Delta^{arb}$$

- Arbiter is modeled in terms of its propagation delay (or clock-to-output time).
- However, unlike delay-differences, the arbiter delay takes positive value and therefore, has a positive mean.
- The percent variation, q, of arbiter delay is modeled as a Gaussian with positive mean and variance increasing with aging. Arbiter delay with aging is expressed as:

$$\Delta^{arb}_{aged} = \Delta^{arb}(1 + q)$$



- Arbiter ages in an asymmetric fashion [3] – the % variation, q, will be much higher than for delay-difference, p_i.

Combined Aging Model

- Environmental noise is added to account for variations in delay parameters.
- Total delay-difference with noise accounted for:

$$r_N = \sum_{i=1}^{N+1} (-1)^{C'_i} \Delta^i + \sum_{i=1}^{N+1} n_i$$

Model assumptions:

- Under a fixed environmental condition, the effect of noise is static (i.e., noise variance remains fixed).
- Variance of percent variations, p_i and q, increases with aging.
- For a fixed amount of aging, we can assume that the variance of q > p_i.

Monte-Carlo simulation for aging

- The original delay parameters are estimated using the LMS method for un-aged PUF.

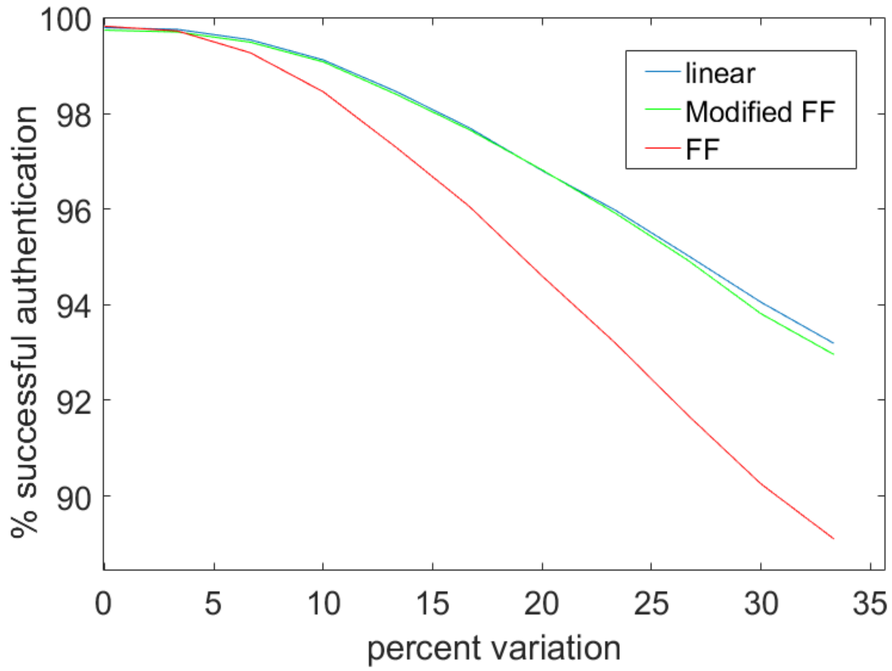
Delay Chain	Arbiter
$\Delta^{i}_{aged} = \Delta^{i}(1 + p_i)$	$\Delta^{arb}_{aged} = \Delta^{arb}(1 + q)$
Δ^{i} estimated from original PUF	Δ^{arb} estimated from original PUF
1000 p_i samples from Gaussian with zero mean and STD(p_i)	1000 q samples from Gaussian with positive mean, μ and STD(q); $\mu^2 = 3\,Var(q)$

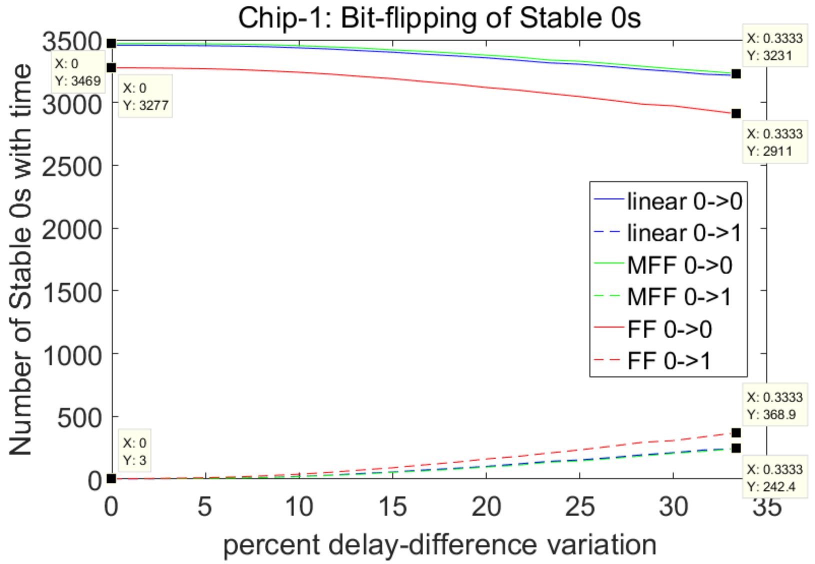
- **Equal aging scenario**: when variation in delay chain and arbiter is assumed same/similar, i.e., STD(p_i)=STD(q).
- **Unequal aging scenario**: when variation in arbiter is more than delay chain, i.e., STD(q)>STD(p_i). Due to asymmetric aging of arbiter, **<u>this scenario is more likely</u>**.

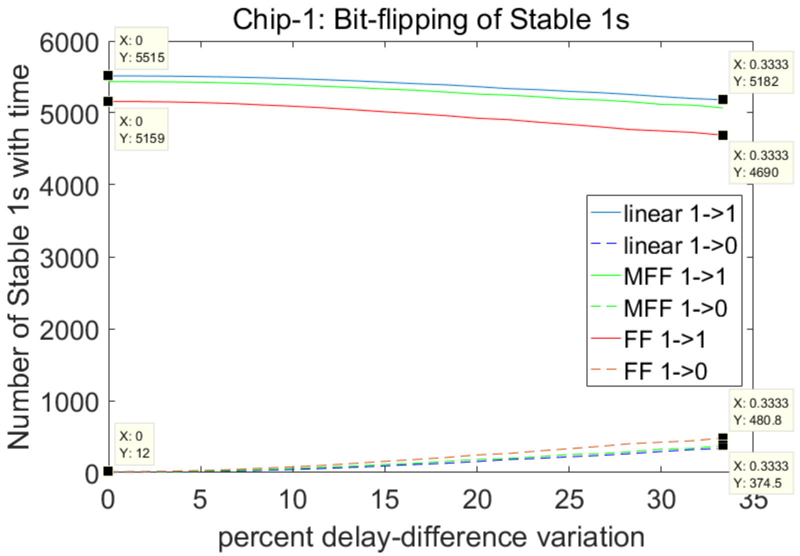
Performance Metrics

- **Reliability or intra-chip variation**
  - Authentication accuracy: % of responses which match with original responses.
  - Divergence between stable-0 and stable-1 distributions using divergence metrics like Jensen-Shannon (JSD) and Henze-Penrose (HPD).
- Uniqueness or inter-chip variation – how different are the responses of each PUF. Uniqueness improves due to random nature of aging.
- Randomness – ability to generate unbiased 0 or 1 as response bit. Randomness decreases due to increase in number of 1s with aging.
- Experimental results are presented for a 32-stage Soft-PUF.

Reliability: Authentication Accuracy (Equal aging)







- Percent variation considers equal variation in both p_i and q.
- **Reliability**: Linear > Modified FF > FF
- **Randomness** decreases as number of bit-flips *Stable-0-->1* are higher than *Stable-1-->0* for all 3 configs.
- Example: Feed-forward has 11.3% *Stable-0-->1* and 9.3% *Stable-1-->0*

Reliability: Authentication Accuracy (Equal aging)

AGING vs NOISE

TABLE I
PERCENTAGE SUCCESSFUL AUTHENTICATION UNDER EQUAL AGING SCENARIO; $STD(q)=STD(p_i)$

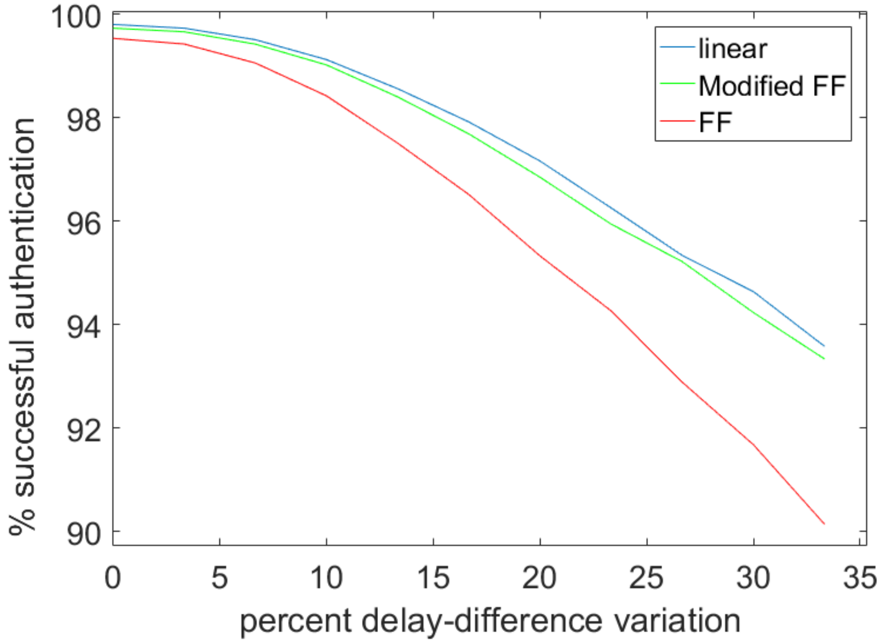
% STD		No Noise	Noise STD=5%	Noise STD=10%	Noise STD=20%
Linear	Original	0.9993	0.9980	0.9930	0.9729
	5%	0.9981	0.9967	0.9917	0.9714
	10%	0.9927	0.9911	0.9860	0.9674
	20%	0.9697	0.9683	0.9639	0.9487
MFF	Original	0.9985	0.9974	0.9921	0.9710
	5%	0.9977	0.9963	0.9911	0.9698
	10%	0.9923	0.9906	0.9854	0.9661
	20%	0.9690	0.9675	0.9629	0.9486
FF	Original	0.9982	0.9954	0.9863	0.9528
	5%	0.9955	0.9927	0.9837	0.9523
	10%	0.9842	0.9817	0.9728	0.9450
	20%	0.9463	0.9442	0.9387	0.9187

- %-authentication is more degraded in case of aging-alone than noise-alone.
- However, the degradation is not significant.

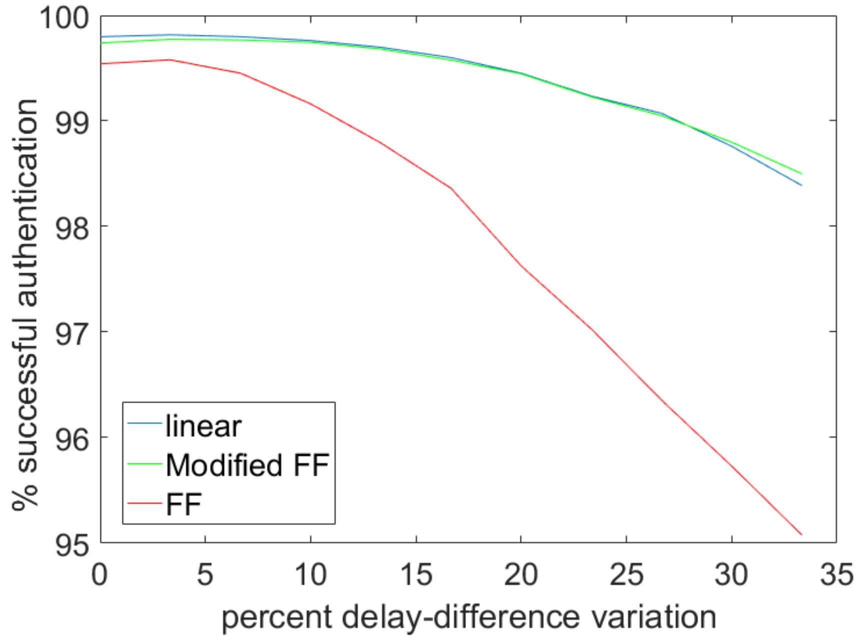
Example:

- %-authentication for FF with 20% STD(p,q) is **94.63%**, whereas with 20% STD(noise) is **95.28%**.
- The difference between their performance is only **0.65%**.

Reliability: Authentication Accuracy (Unequal aging)



Delay chain



Arbiter

- Authentication accuracy with aging considered for delay chain and arbiter separately is shown.
- We expect STD(q) > STD(p_i) ⇨ degradation due to arbiter becomes much more significant than due to delay chain.

Reliability: Authentication Accuracy (Unequal aging)

AGING vs NOISE

TABLE II

PERCENTAGE SUCCESSFUL AUTHENTICATION UNDER UNEQUAL AGING SCENARIO; $STD(q)$=$STD(p_i)$+20%

$\% STD(p, q)$		No Noise	Noise STD=5%	Noise STD=10%	Noise STD=20%
Linear	(0,20)	0.9961	0.9941	0.9892	0.9704
	(5,25)	0.9914	0.9898	0.9843	0.9657
	(10,30)	0.9825	0.9805	0.9761	0.9594
	(20,40)	0.9568	0.9556	0.9526	0.9409
MFF	(0,20)	0.9960	0.9944	0.9891	0.9694
	(5,25)	0.9912	0.9896	0.9848	0.9663
	(10,30)	0.9827	0.9815	0.9761	0.9592
	(20,40)	0.9566	0.9561	0.9528	0.9391
FF	(0,20)	0.9795	0.9773	0.9700	0.9441
	(5,25)	0.9672	0.9654	0.9591	0.9354
	(10,30)	0.9506	0.9494	0.9433	0.9248
	(20,40)	0.9136	0.9124	0.9096	0.8937

- %-authentication is more degraded in case of aging-alone than noise-alone.
- However, the degradation in this case is much more significant.

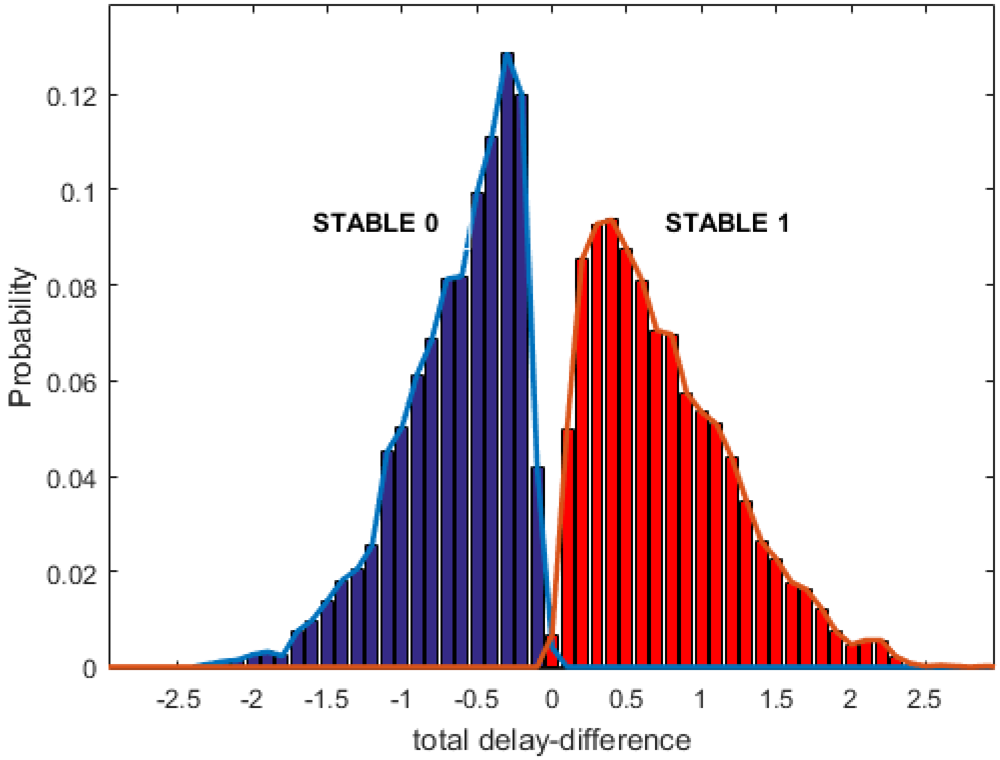
Example:

- %-authentication for FF with 20% STD(p_i), 40% STD(q) is **91.36%**, whereas with 20% STD(noise) is **95.28%**.
- The difference between their performance now is **3.92%**.

Reliability: Divergence Metrics

Total delay-difference distribution of stable 0 and stable 1

- Ideally, there should be no overlap between the stable 0 and stable 1 distributions – represents error/noise of the model.
- With aging, as delay parameters start to vary, the overlap between these distributions increase.
- This overlap reflects the bit-flips occurring in the responses of these challenges.
- Metrics like Jensen-Shannon, Henze-Penrose divergence are used to analyze these overlaps.



Probability distributions for unaged linear MUX PUF

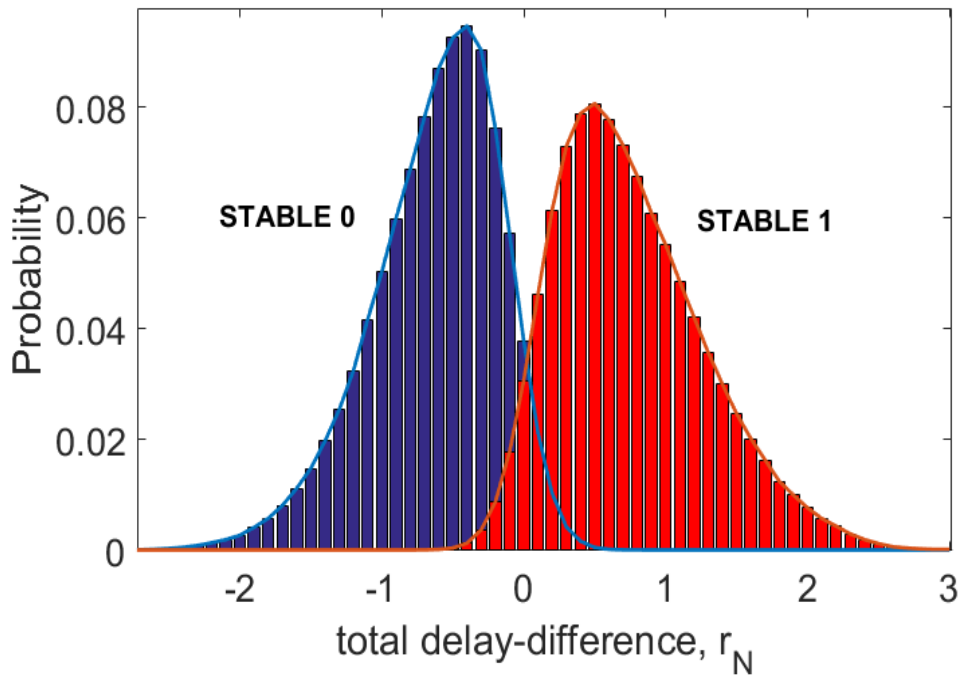
Reliability: Divergence Metrics

- **Jensen-Shannon divergence**: symmetric form of KL divergence. KL divergence was found to be sensitive to low values of probability.

$$JS(P||Q) = \frac{1}{2}(KL(P||R) + KL(Q||R)),$$

$$where\ R = \frac{1}{2}(P + Q)$$

- **Henze-Penrose divergence [7]**: Randomly sample r_N from a set of r_N values obtained for an equal number of Stable-0 and Stable-1 CRPs. Sort them in increasing/decreasing order and count the number of differing classification, *R* out of total N. HPD is computed as: $\boldsymbol{HPD = 1 - R / N}$

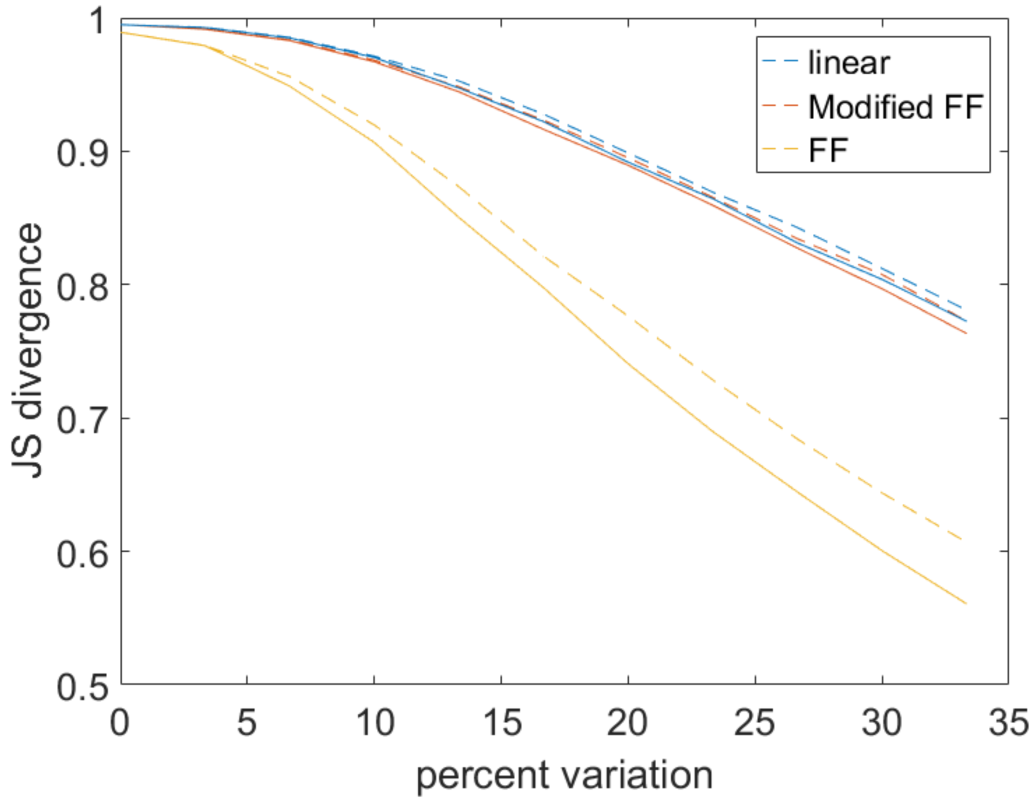


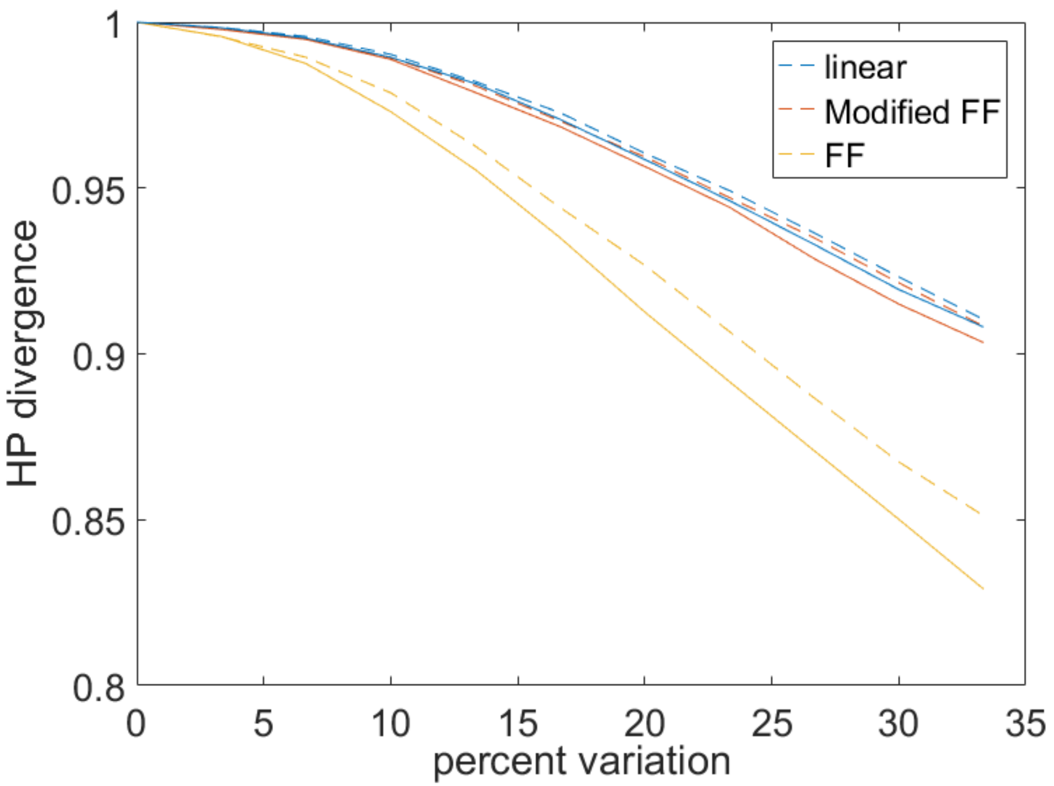
STD(q)=STD(p_i)=$\frac{STD(ni)}{STD(\Delta_i)}$= 30%

Probability distributions with 30% variation due to aging/noise

[7] M. Brown, T. Netoff, and K. Parhi, “A low complexity seizure prediction algorithm,” in *Proc. IEEE Eng. Med. Biol. Soc.*, 2011, pp. 1640–1643

Reliability: Divergence Metrics



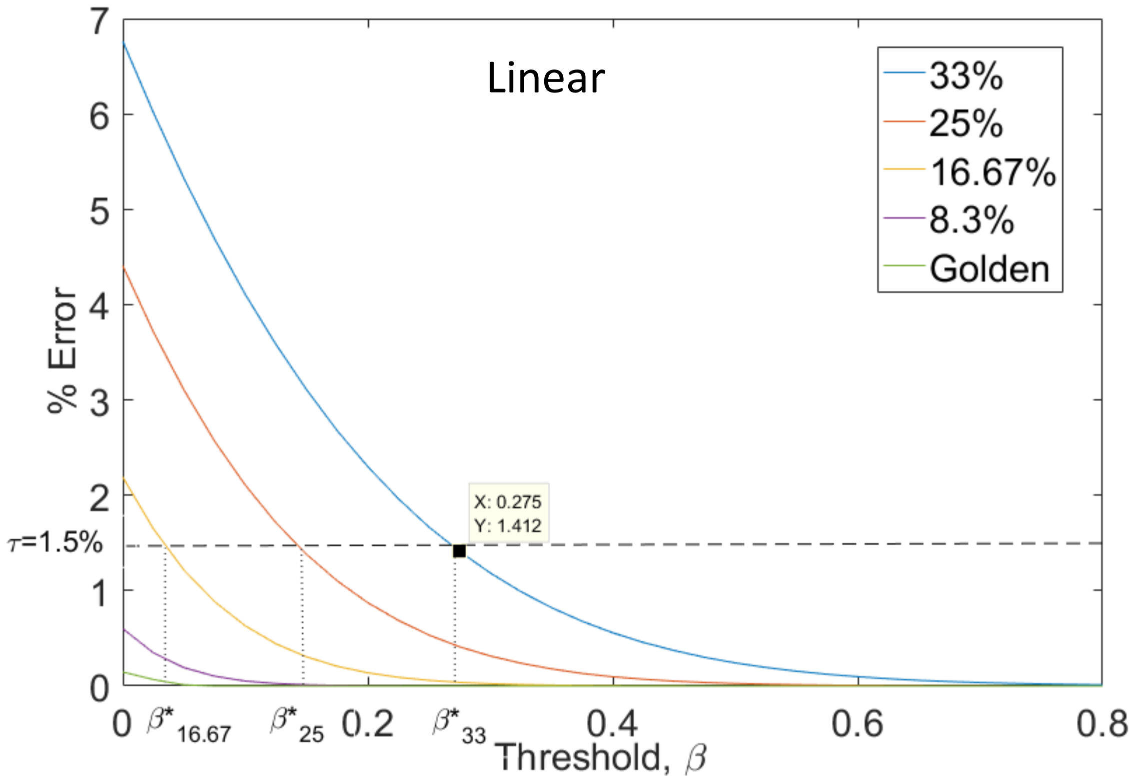


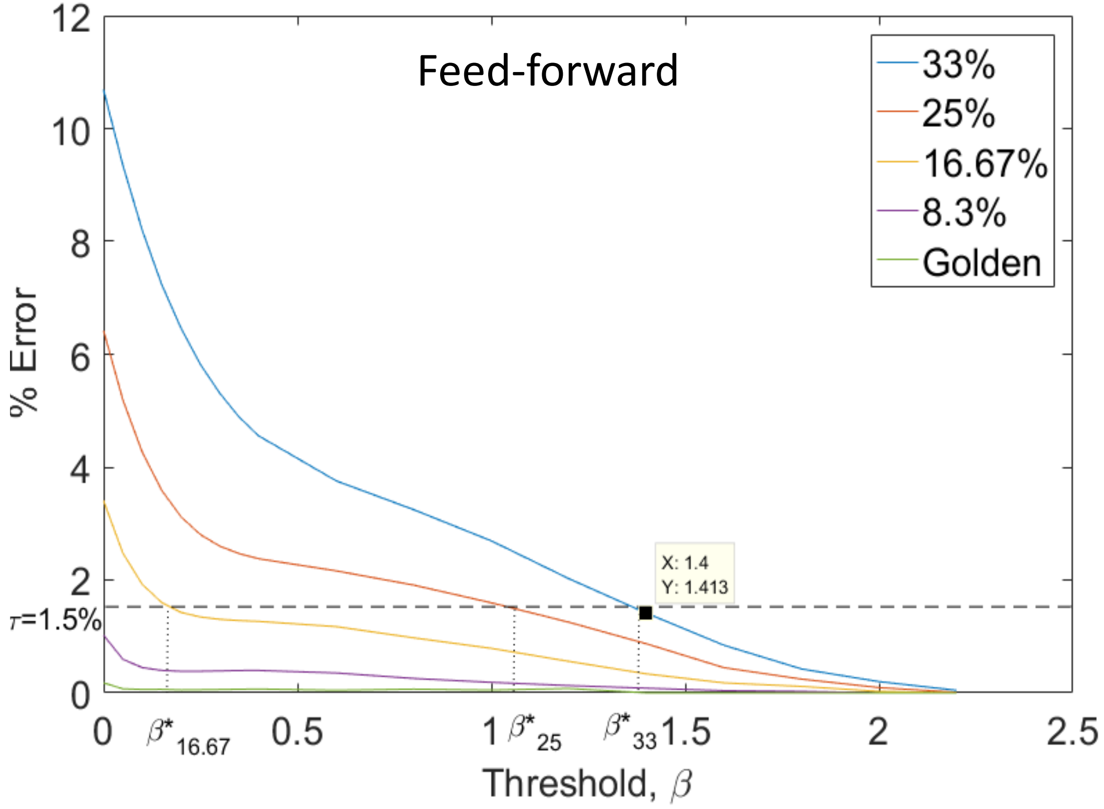
- Dashed lines show the performance in case of **noise-alone** and solid line for the case of **aging-alone** scenario.
- A lower divergence value corresponds to a higher overlap between the Stable-0 and Stable-1 distributions.
- Range of JSD is 0-1 and HPD is 0.5-1.

Improving reliability

- **Recalibration**: The delay parameters can be recalibrated using LMS method. But not feasible as thousands of devices will need to be need calibration.
- **Tuning a threshold** based on total delay-difference, r_N:
  - Challenges with r_N close to 0 are more prone to aging related bit-flips.
  - Therefore, choosing a threshold on r_N will improve the reliability albeit a lower number of available challenges.
  - Higher the threshold, better reliability is guaranteed.
  - However, we do not need 100% reliability as certain error in the responses to the set of challenges is tolerated. Threshold requirement is further lower in this case.

Improving reliability of 32-stage PUF





Plots show % Error in authentication by considering only CRPs with $|r_N| \geq \beta$.

Example:

- Error Tolerance = 1.5%, Thresholds for STD(p_i)=STD(q)=33% (equal) amount of aging - **0.275** for linear, **1.4** for feed-forward.
- # of challenges with with $|r_N| \geq \beta$,threshold – 2^{31}, 2^{26} respectively.

Conclusion

- Aging effects of delay chain and arbiter can be modeled in terms of Gaussian distributions.
- Aging degradation is similar (or slightly worse) to that of noise under equal aging scenario. This is because aging and noise are modeled in a similar manner.
- Under the assumption that arbiter “ages” much more significantly than delay chain (unequal aging), the performance degradation due to aging is much more prominent compared to noise.
- The performance degradation due to aging can be improved by tuning thresholds based on total delay-difference. This decreases the number of challenges available for authentication purposes.