Lightweight and Secure PUF Key Storage Using Limits of Machine Learning

Meng-Day (Mandel) Yu[1], David M'Raïhi[1], Richard Sowell[1], Srinivas Devadas[2]

[1]Verayo, Inc., San Jose, CA, USA
[2]MIT, Cambridge, MA, USA

Agenda

Physical Unclonable Function (PUF) Overview

PUF Noise Profile

- Response size, temperature, voltage

Deriving Stable PUF Bits

- Traditional: Large block ECC, Two-stage ECC
- Lightweight: Stable bits w/o complex ECC

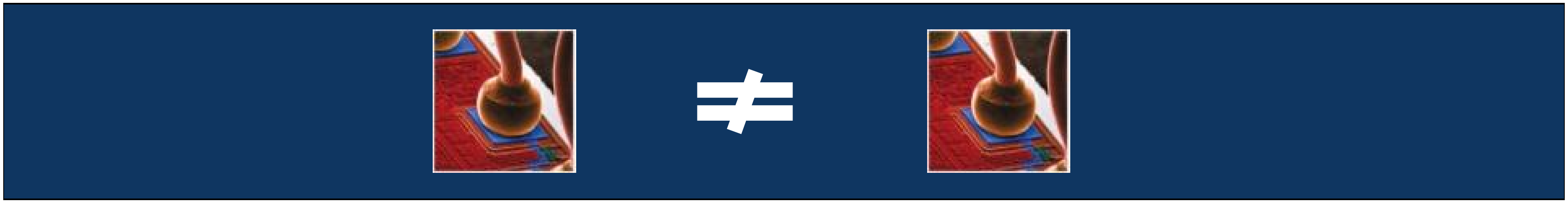
Security Framework

- *“Use what cannot be learned about the system”*

Conclusions

PUF Overview

Physical Unclonable Functions (PUF)

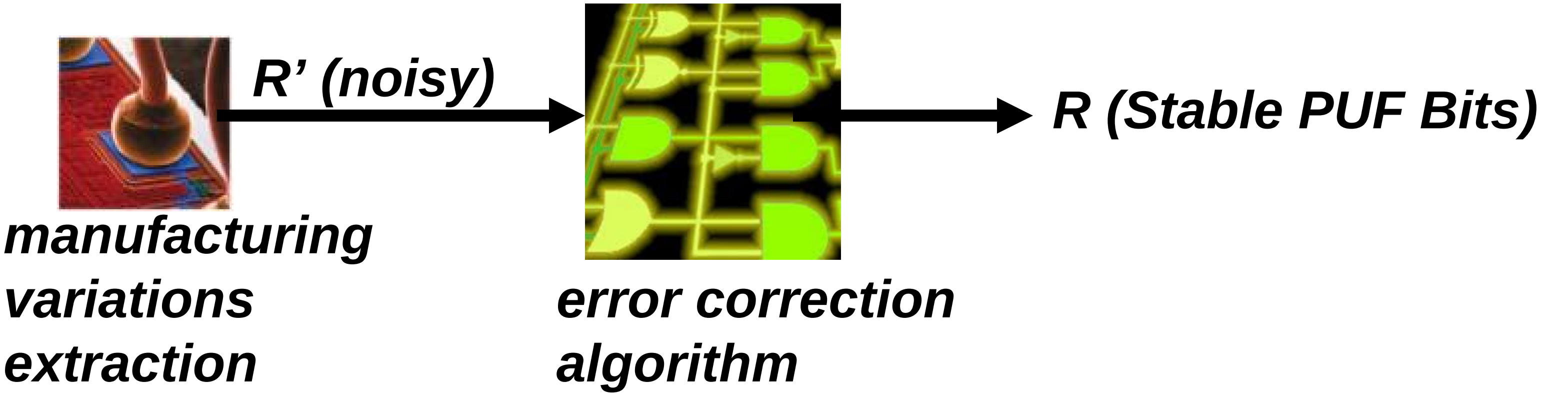


Tiny electronic circuits extract silicon *manufacturing variations*

Unique characteristics = “silicon biometrics”

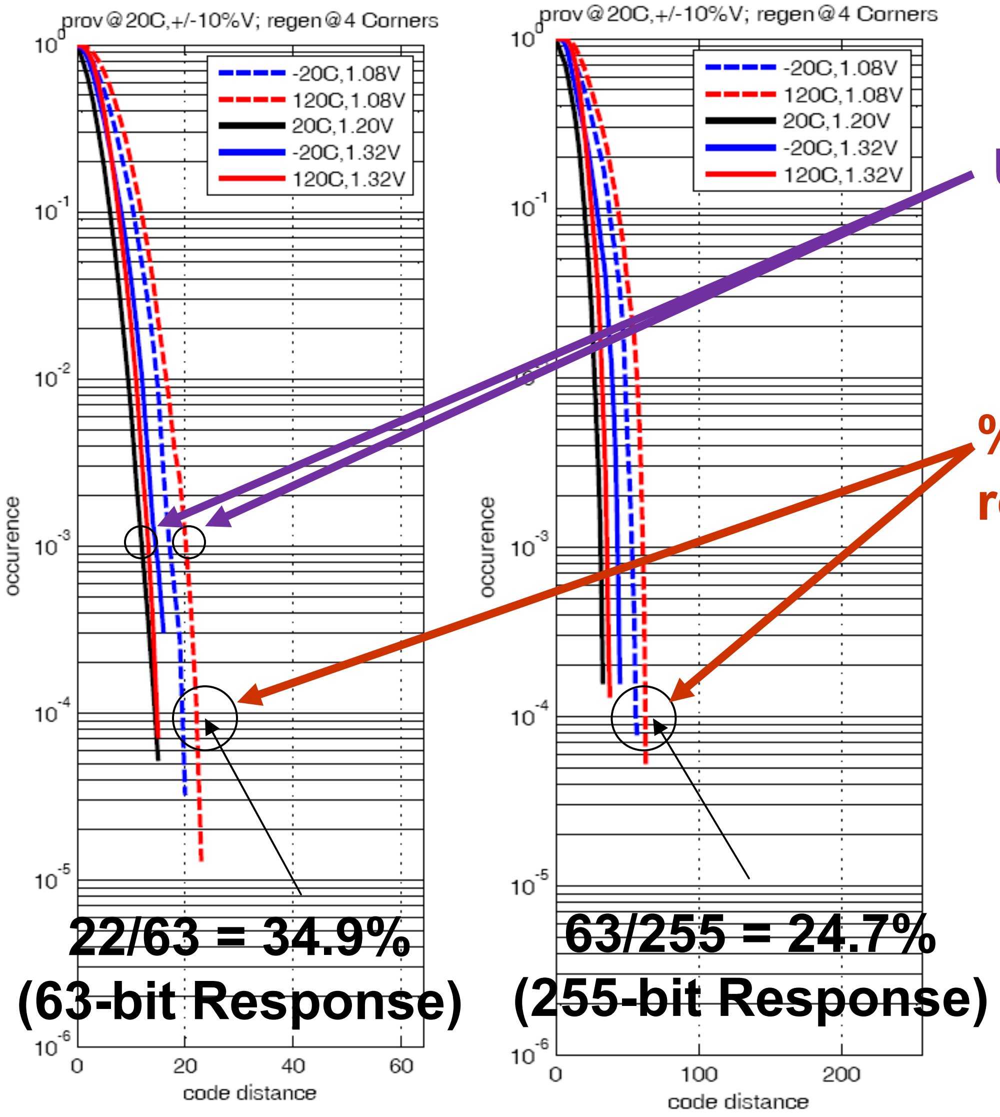
PUF responses are “noisy”

To generate Stable PUF Bits: add error correction algorithm


R’ (noisy)
R (Stable PUF Bits)
manufacturing variations extraction
error correction algorithm

PUF Noise Profile

PUF Noise Profile



22/63 = 34.9%
(63-bit Response)

63/255 = 24.7%
(255-bit Response)

Unstable bits ↑ with ΔV, ΔT

- **Noise ↑ ~1.5x to ~2x**

% of unstable bits ↓ with larger response size

- **BCH error correction limit = 25%**
- **Use larger response size to correct noisier PUF**

Deriving Stable PUF Bits

Methods to Derive Stable Bits

Large Block ECC

- Single stage error correction
  - BCH(255,63,t=30) [Suh-MIT2005]
  - BCH(255,13,t=59) [AMSSW-IEEE_S&P2011]

Two-stage ECC

- Quadratic reduction in complexity
  - Repetition(11,1,t=5) + Golay(24,13,t=3) [BGSST-CHES2008]
  - Repetition(11,1,t=5) + RM(16,5,t=3) [BGSST-CHES2008]
  - Repetition$_{\text{SoftDecision}}$(3,1,t=1) + RM$_{\text{SoftDecision}}$(64,22,t=7) [MTV-CHES2009]
  - IBS + BCH(63,30,t=6) [YD-IEEE_D&T2010]

Lightweight (*no complex ECC*)

- Use “Index Based Syndrome” (IBS) w/o BCH
- Additional complexity reduction (75%)
- Add retry, simple coding to improve reliability

Index-Based Syndrome (IBS) Coding

From [YD-IEEE_D&T2010]

Use a group of PUF output values to store a bit sequence

Simple case: a sequence of 1 bit

- Encoder:
  - If $\underline{B} = 1$, $\underline{S}$ = index of $f_1(\underline{R}_0 = r_0, \ldots \underline{R}_{q-1} = r_{q-1})$
  - If $\underline{B} = 0$, $\underline{S}$ = index of $f_0(\underline{R}_0 = r_0, \ldots \underline{R}_{q-1} = r_{q-1})$

Let f_1 = max function, f_0 = min function
$\underline{B}$ = bit to store, $\underline{S}$ = Syndrome Word

- Decoder:
  - $\underline{B'}$ = sign_of $(\underline{R}_S)$

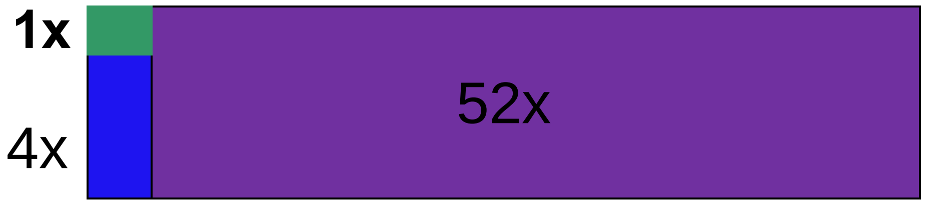
Advantages:

- Trivially simple encoder and decoder
- High coding gain -> reduction in ECC complexity
- Provably secure (more later)

Size Comparisons (Xilinx Virtex-5 LX50)

Lightweight (IBS)	2-stage ECC (IBS + BCH63)	Large Block (BCH255)
69 registers	471 registers	6400 registers (est. using 16x)
~1.2% SLICE count (99/7200)	~5% SLICE count (393/7200)	~65% SLICE count

- **Includes decoder + encoder**
- **Does not include APB interface, I/O buffering**
- **Even smaller if test logic, configurability removed**



Decoder Core Comparisons (Xilinx Spartan 3E-500)

Retargeted implementation for Spartan 3E (w/o optimizations) for comparison

Use best results from [MTV-CHES2009], [BGSST-CHES2008]

Goal: 128-bit key

	Current Work		[MTV-CHES2009]	[BGSST-CHES2008] PUF-Optimized	[BGSST-CHES2008] Decoder-Optimized
Area	116 SLICES (no μcode required)		164 SLICES (μcode ROM required)	580 SLICES (μcode ROM required)	110 SLICES (μcode ROM required)
Dec Cycles	~16640 cycles (@ 100Mhz+)		~10298 cycles (@ 50.2Mhz)	>= 24024 cycles (@ 151.5Mhz)	>= 29925 cycles (@ 175.4Mhz)
Helper Data	780 bit		13952 bit	3824 bit	6288 bit
PUF Size	1280 OSC* (Security-Optimized)	256 OSC* (PUF-Optimized)	1536 bit SRAM**	3696 bit SRAM**	6160 bit SRAM**
Stability	-55°C to 125°C, V_{nom} +/- 10%, WC VT Corners, Aging		PUF noise model does not account for V,T.	PUF noise model accounts for -20°C to 80°C [GSST-CHES2007]. No voltage.	
Security	Insufficient equations to learn system (no i.i.d assumption)	Use unlearnable part of system (no i.i.d. assumption)	Soft decision information is information-theoretically secure (i.i.d. assumption)	No explicit security argument to account for leaks associated w/ heavy repetition coding	

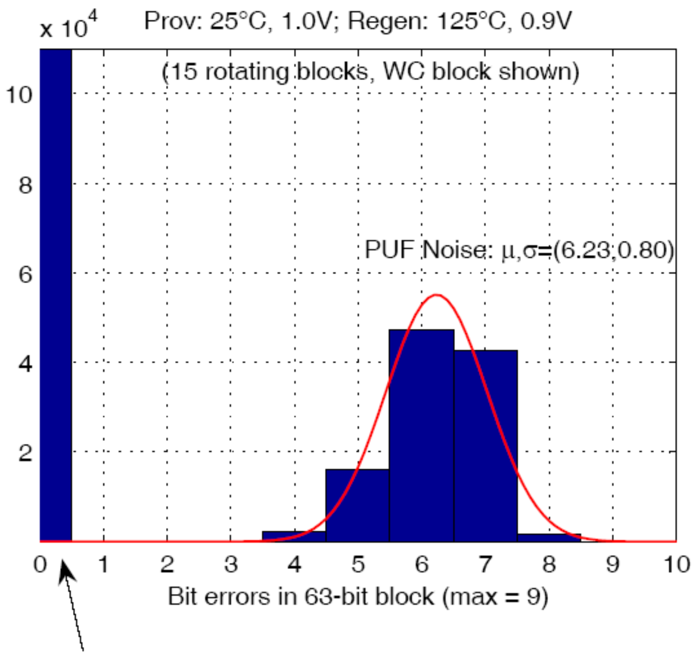
* 5 inversions per OSC (~3 NAND2 equivalent gate, 1st order est.) ** 6T cell per bit (~3 NAND2 equivalent gate)

WC Voltage / Temperature Corners

Empirical PUF data from Xilinx Virtex-5 FPGAs

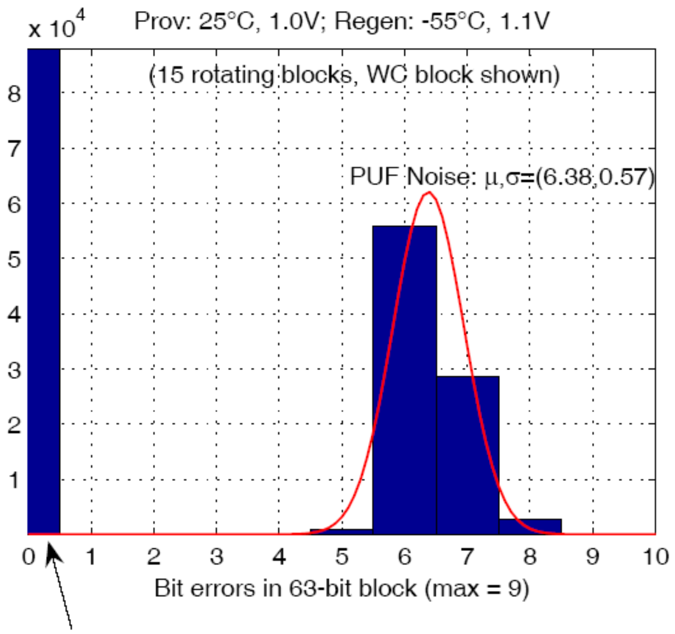
Error Free Performance using 4-bit Index

- **1M+ blocks, implied failure rate < 1 ppm**
- **SS Corner 125oC, 0.9V**
- **FF Corner -55oC, 1.1V**



1.65M+ blocks, no errors (4-bit Indexing)

Implied block failure rate <= 0.61ppm



1.32M+ blocks, no errors (4-bit Indexing)

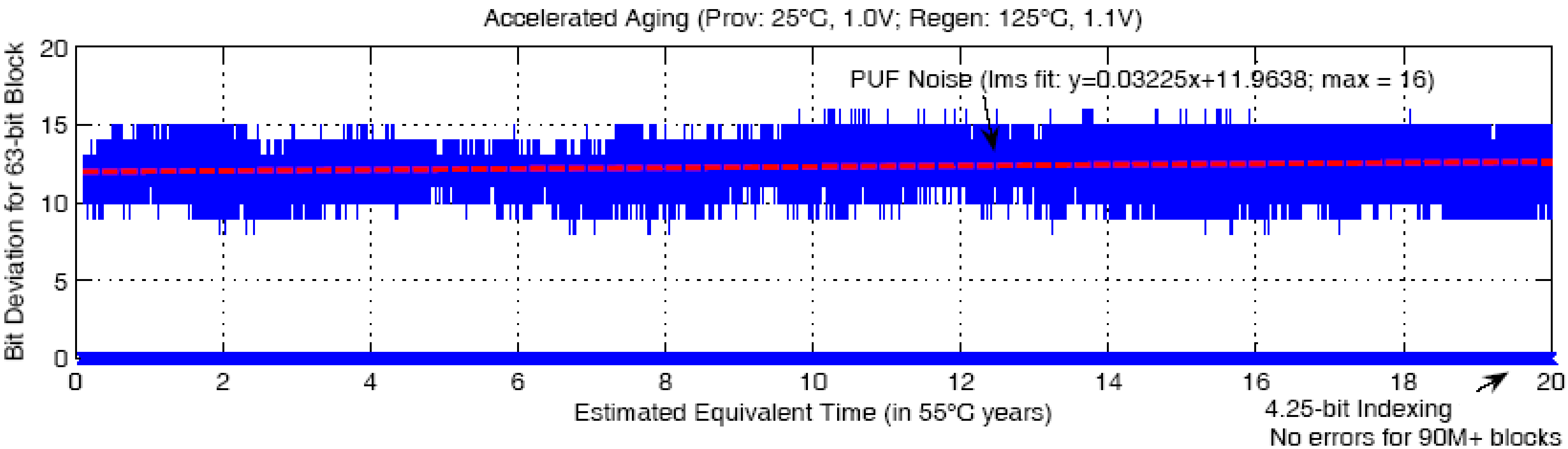
Implied block failure rate <= 0.76ppm

Accelerated Aging

~90M+ blocks, error free performance, 4.25 bit Index

- Implied error rate <= 12 parts per ***billion*** (ppb)
- Accelerated age: 20+ yrs @ 55oC
- Provisioning: 25oC, 1.0V; Regeneration: 125oC, 1.10V

Aging deteriorates silicon, increasing Indexing requirement by ¼ bit



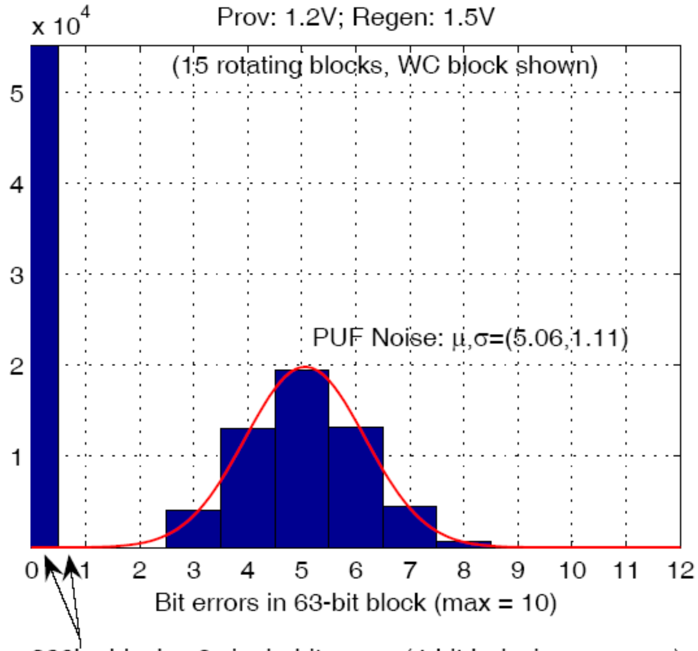
4.25-bit Indexing
No errors for 90M+ blocks
Implied block failure rate <= 12ppb

Voltage Testing, ASIC

PUF + Indexing Algorithm in .13µm silicon

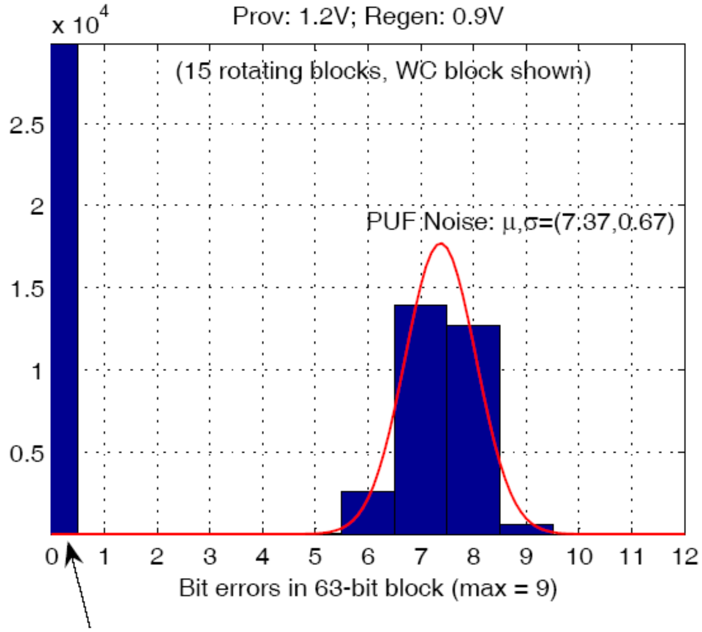
- 4 to 5 bit Index for reliable (ppm level or better) operation

Results consistent with FPGA


Prov: 1.2V; Regen: 1.5V

826k+ blocks, 2 single-bit errors (4-bit Indexing)

Block failure rate ~2.5ppm


Prov: 1.2V; Regen: 0.9V

447k+ blocks, no errors (5-bit Indexing)

Implied block failure rate <= 2.24ppm

Security Framework

Security Dependencies of Prior Work (1)

Recall: [BGSST-CHES2008]

- **No explicit security argument for use of Repetition[11,1,t=5] code**
- **Heavy repetition coding highly sensitive to PUF bias:**

Bits leaked per repetition-coded bit =

$$\frac{\left|PUFbias - 0.5\right|}{\left|\frac{\lceil repetition/2 \rceil}{repetition} - 0.5\right|}$$

[YD-IEEE_D&T2010]

1. if PUF bias = .55, all bits leaked!

2. if PUF bias = .505, 1 bit leaked out of every 9 bits repetition-coded

… this is true even if PUF output bits are assumed to be i.i.d.

<u>Current work avoids heavy repetition coding</u>

Security Dependencies of Prior Work (2)

[MTV-CHES2009] and [YD-IEEE_D&T2010] both use proofs that require i.i.d. PUF output assumption (implicitly or explicitly)

Questions:

- Memory PUF: Are there correlations based on memory word columns?
- Arbiter PUF / OSC PUF: Are there correlations with reuse of delay elements?

Can we remove i.i.d. assumption?

Unconditional Security

Recall:

- Shannon Entropy: $H(\underline{X}) = -\, p(x) \log_2 p(x)$
- Mutual Information: $I(\underline{Y}\,;\underline{X}) = H(\underline{Y}) - H(\underline{Y} \mid \underline{X})$

Unconditional security (perfect secrecy) [Shannon, 1949]

- Ciphertext share no information with the Key
- Secure against a *computationally-<u>un</u>bounded* adversary
- ***<u>Strongest form of security</u>***
- Information shared between Ciphertext and Key:
  - $\mathbf{I}(\underline{CT}^{alg}; \underline{Key}) = \mathbf{H}(\underline{Key}) - \mathbf{H}(\underline{Key} \mid \underline{CT}^{alg})$

We adapt this unconditional security measure to develop a syndrome leakage metric…

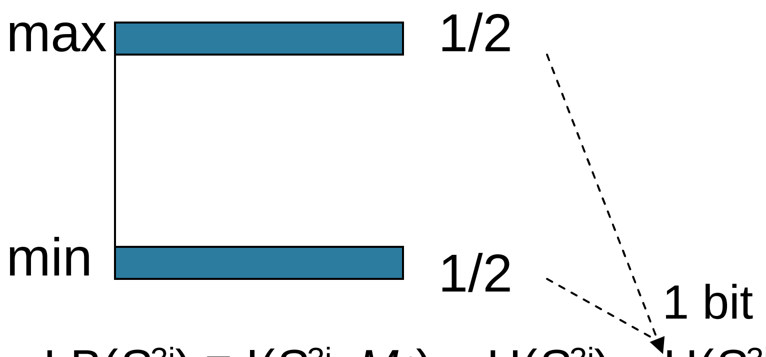
Leaked Bits (LB)

What is the information shared between a Syndrome Word and a perfect model of the PUF?

- Code offset [Dodis, 2004], 3x repetition coding
  - **$\mathrm{LB}(\underline{S}^{3x}) \equiv \mathrm{I}(\underline{S}^{3x}; \underline{M}^{\infty}) = \mathrm{H}(\underline{S}^{3x}) - \mathrm{H}(\underline{S}^{3x} \mid \underline{M}^{\infty}) = 3 - 1 = 2$ bits**
- Index-Based Syndrome (IBS) Coding [Yu, 2010], 3-bit index
  - **$\mathrm{LB}(\underline{S}^{3i}) \equiv \mathrm{I}(\underline{S}^{3i}; \underline{M}^{\infty}) = \mathrm{H}(\underline{S}^{3i}) - \mathrm{H}(\underline{S}^{3i} \mid \underline{M}^{\infty}) = 3 - 1 = 2$ bits**
- ***Can we leak less information?***

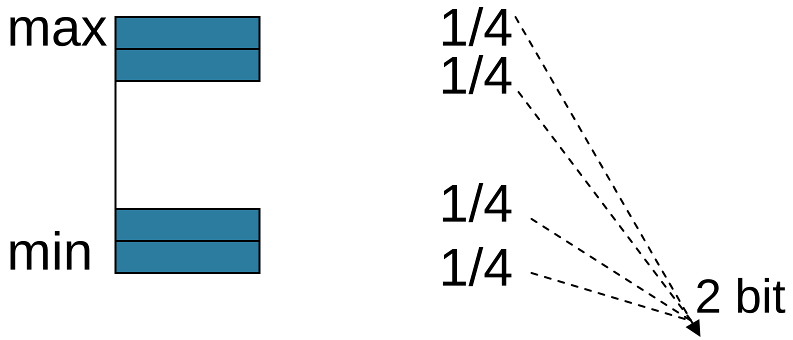
Syndrome Distribution Shaping (SDS): Intuition

IBS: pick most + or most - value, to encode a “1” bit or a “0” bit



$LB(\underline{S}^{3i}) \equiv I(\underline{S}^{3i}; \underline{M}^{\infty}) = H(\underline{S}^{3i}) - H(\underline{S}^{3i} \mid \underline{M}^{\infty}) = 2$ bits

SDS: randomly select two max or two min



Reduced Leaked Bits (per Syndrome Word) by “flattening” distribution

$LB(\underline{S}^{3i}) \equiv I(\underline{S}^{3i}; \underline{M}^{\infty}) = H(\underline{S}^{3i}) - H(\underline{S}^{3i} \mid \underline{M}^{\infty}) = 1$ bits

Syndrome Distribution Shaping (SDS)

Let *p* = probability a PUF output choice is ignored or skipped

- i.e., the max or min selection ignores that PUF output choice

Reducing Leaked Bits while preserving error correction power:

- $I(\underline{S}^{3i}, \underline{M}^{\infty}) = 2$ bits
- $I(\underline{S}^{W=4,\ p\,=\,1/2}, \underline{M}^{\infty}) = 1.02$ bits
- $I(\underline{S}^{W=5,\ p\,=\,3/4}, \underline{M}^{\infty}) = 0.80$ bits
- $I(\underline{S}^{W=6,\ p\,=\,7/8}, \underline{M}^{\infty}) = 0.71$ bits
- $I(\underline{S}^{W=7,\ p\,=\,15/16}, \underline{M}^{\infty}) = 0.67$ bits

"Choosing best out of 8"
"Choosing best out of 16, w/ ~half of the choices eliminated"

Leaked Bits ↓2x!

Machine Learning Results

Ruhrmair, et. al., “Modeling Attacks on PUFs”, ACM CCS 2010.

$$N_{CRP} \approx 0.5 \frac{k+1}{\varepsilon}$$

N_{CRP} : number of challenge / response pairs

k: # of delay parameters in an arbiter PUF

ε: classification error

Observation: Adversary with k C/R pairs cannot do much better than guessing, i.e., $\varepsilon \approx 0.5$.

What *cannot* be learned? (1)

Now rearrange the equation, rename terms, etc.

$$N_{CRP} \approx 0.5\frac{k+1}{\varepsilon} \Longrightarrow \varepsilon \approx \min(0.5\frac{k+1}{\Sigma LB},0.5)$$

Conservative: stay *safely within* boundary where ε = 0.5 such that virtually nothing is learned from Syndrome Bits.

When 0 <= ε < 0.5, something is learned from the Syndrome Bits.

But how much information *cannot* be learned?

What *cannot* be learned? (2)

We know "bias" reduces min-entropy H_∞

Now, instead of "bias", we have ε *conditioned upon* Syndrome Words known to the adversary at some point in time

-> Use *conditional* version of min-entropy

$$\tilde{H}_\infty(\underline{X} \mid \underline{Y}) \equiv -\log_2 \left(E_{y \leftarrow \underline{Y}}\left[2^{-H_\infty(\underline{X} \mid \underline{Y}=y)}\right]\right)$$

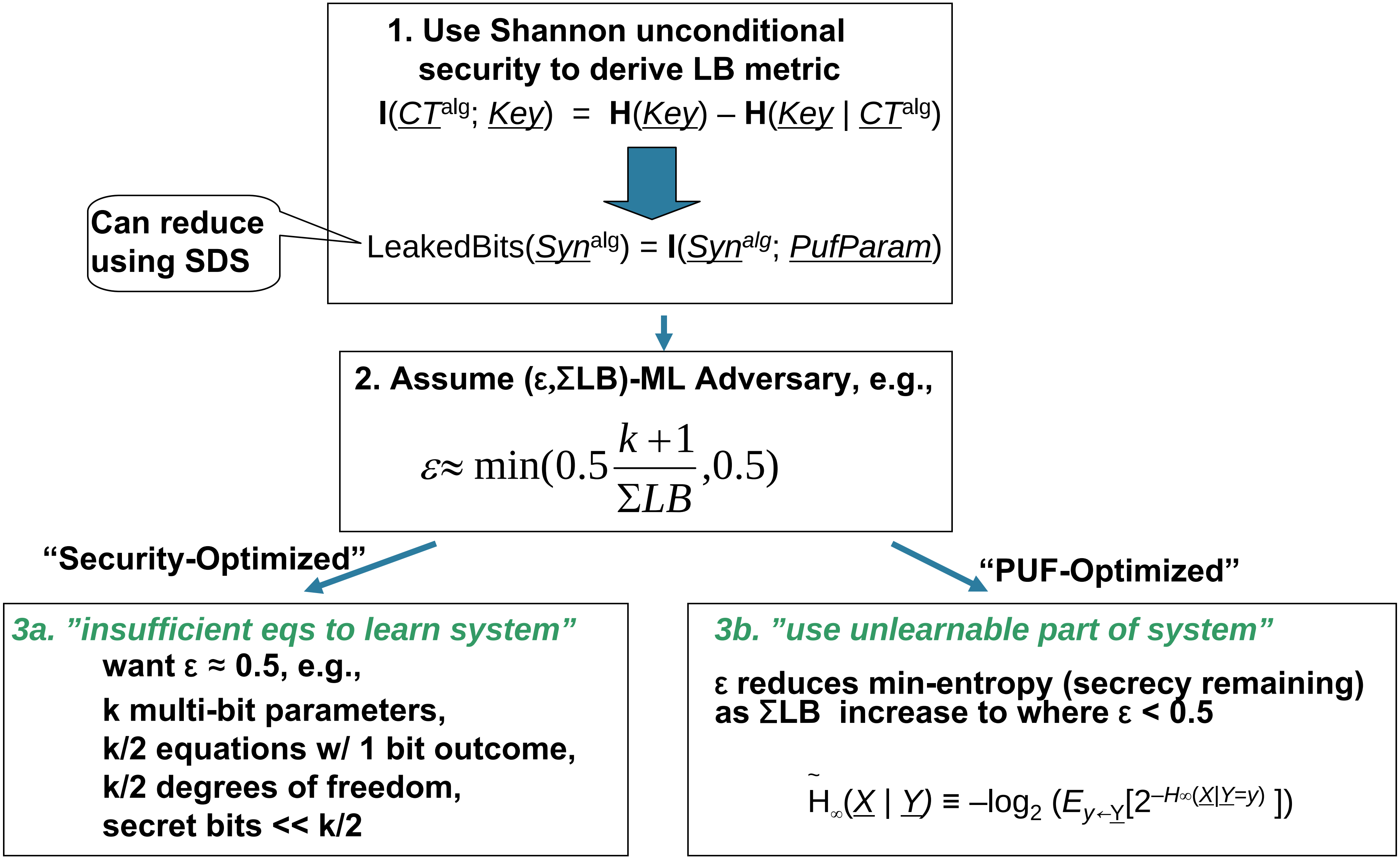
[Dodis, "Fuzzy Extractor", 2004]

where: $H_\infty \equiv -\log_2 (\mathrm{pr}_{max}(.))$

we note this applies to a bit-oriented learner as well as a block-oriented learner

- What the adversary can learn is reflected in classification error ε
- $\tilde{H}_\infty$ computes the amount of secrecy left in the system using ε
- $\tilde{H}_\infty$ reflects the min-entropy that the ML-adversary cannot "touch" or learn

Security Sketch



1. Use Shannon unconditional security to derive LB metric

$$\mathbf{I}(\underline{CT}^{alg}; \underline{Key}) = \mathbf{H}(\underline{Key}) - \mathbf{H}(\underline{Key} \mid \underline{CT}^{alg})$$

$$\text{LeakedBits}(\underline{Syn}^{alg}) = \mathbf{I}(\underline{Syn}^{alg}; \underline{PufParam})$$

Can reduce using SDS

2. Assume (ε,ΣLB)-ML Adversary, e.g.,

$$\varepsilon \approx \min(0.5\frac{k+1}{\Sigma LB}, 0.5)$$

"Security-Optimized"

"PUF-Optimized"

3a. "insufficient eqs to learn system"

want ε ≈ 0.5, e.g.,

k multi-bit parameters,
k/2 equations w/ 1 bit outcome,
k/2 degrees of freedom,
secret bits << k/2

3b. "use unlearnable part of system"

ε reduces min-entropy (secrecy remaining) as ΣLB increase to where ε < 0.5

$$\tilde{H}_\infty(\underline{X} \mid \underline{Y}) \equiv -\log_2 (E_{y \leftarrow \underline{Y}}[2^{-H_\infty(\underline{X} \mid \underline{Y}=y)}])$$

Now, without i.i.d. PUF output assumption…

Secure Construction #1:

- **640 OSC pairs, forming k = 640 “delay parameters”**
- **Only k / 2 equations (each with a 1 bit outcome) leaked via Syndrome**
- ***320 degrees of freedom to keep secret 128-bits of information***

”insufficient equations to learn system”

Secure Construction #4:

- **128 OSC pairs, forming k = 128 “delay parameters”**
- **Use ε curve to compute amount of secrecy left in the system**
- **Can extract a 128-bit key using results from** [Ruhrmair, 2010]
  - **Secure against a (ε,ΣLB) Machine-Learning adversary**

“use unlearnable part of system”

Conclusions

Conclusions

Lightweight PUF Key Generation

- 75% reduction in complexity from 2-stage ECC
  - Two stage ECC better characterized and also available
- Environmentally Stable
  - Temperature: -55°C to 125°C
  - Voltage: V_{nom} +/- 10%
  - WC VT Corners
  - Aging: 20+ yrs @ 55°C
  - Error free, 90M+ tests, FPGA, ASIC

Implied error rate <= 12 parts per *billion*

Security Framework

- Security-Optimized: "Insufficient eqs to learn sys"
- PUF-Optimized: "Use unlearnable part of system"

No i.i.d. PUF output assumption

Future Work

- De-rate results to account for side channel information

THANK YOU!

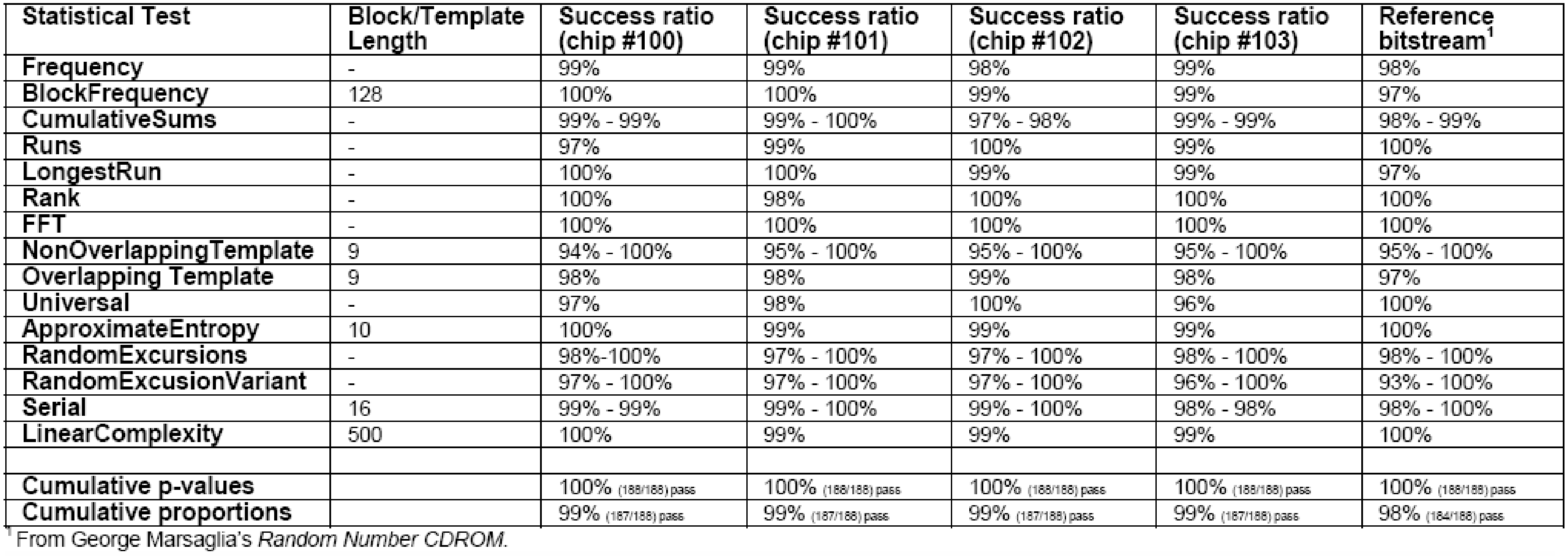
Extras:

PUF Randomness / Uniqueness

NIST Randomness Test Results

Verayo PUF Output bits are “random” based on NIST testing

- **Low bias (< +/- 1%)**
- **Results affirmed using on other statistical testing methods**

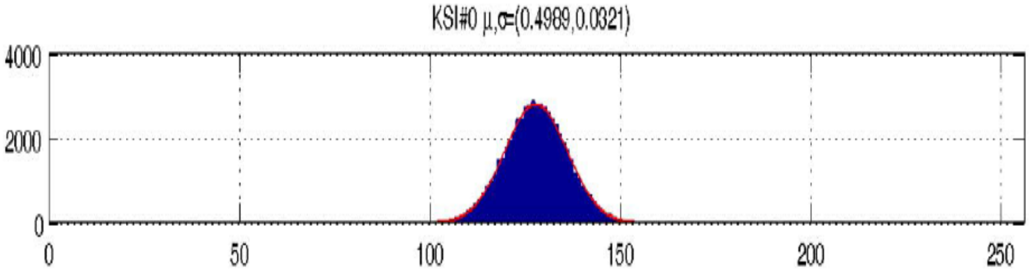


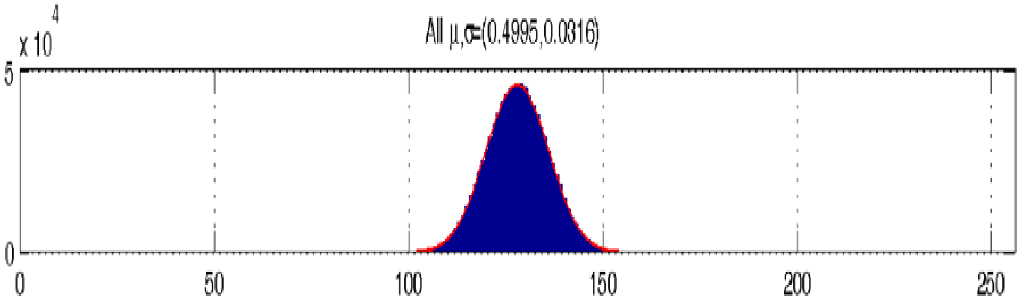
Statistical Test	Block/Template Length	Success ratio (chip #100)	Success ratio (chip #101)	Success ratio (chip #102)	Success ratio (chip #103)	Reference bitstream[1]
Frequency	-	99%	99%	98%	99%	98%
BlockFrequency	128	100%	100%	99%	99%	97%
CumulativeSums	-	99% - 99%	99% - 100%	97% - 98%	99% - 99%	98% - 99%
Runs	-	97%	99%	100%	99%	100%
LongestRun	-	100%	100%	99%	99%	97%
Rank	-	100%	98%	100%	100%	100%
FFT	-	100%	100%	100%	100%	100%
NonOverlappingTemplate	9	94% - 100%	95% - 100%	95% - 100%	95% - 100%	95% - 100%
Overlapping Template	9	98%	98%	99%	98%	97%
Universal	-	97%	98%	100%	96%	100%
ApproximateEntropy	10	100%	99%	99%	99%	100%
RandomExcursions	-	98%-100%	97% - 100%	97% - 100%	98% - 100%	98% - 100%
RandomExcusionVariant	-	97% - 100%	97% - 100%	97% - 100%	96% - 100%	93% - 100%
Serial	16	99% - 99%	99% - 100%	99% - 100%	98% - 98%	98% - 100%
LinearComplexity	500	100%	99%	99%	99%	100%
Cumulative p-values		100% (188/188) pass	100% (188/188) pass	100% (188/188) pass	100% (188/188) pass	100% (188/188) pass
Cumulative proportions		99% (187/188) pass	99% (187/188) pass	99% (187/188) pass	99% (187/188) pass	98% (184/188) pass

[1] From George Marsaglia’s *Random Number CDROM.*

Uniqueness Analysis

240 PUF devices (12 FPGAs, 20 PUFs each)





Comparisons	μ	δ
57k	.1% from ideal	.0321
920k	.05% from ideal	.0316

“sample mean converges to true mean for iid process and unbiased estimator”

“Student-t distribution converges to Gaussian as sample size → ∞”

Conclusions: μ and σ *should not get worse* with increase in number of comparisons