SYNOPSYS®

Entropy from Silicon

A Constructive Approach to PUF-based Roots of Trust

Roel Maes, R&D Manager, Synopsys

CASCADE 2026, Regensburg

Setting the Scene

Roots of Trust

Information Security from the Ground up

security reductions:
- provable, or …
- heuristic

security assumptions!

CRYPTOGRPAHY

PROTOCOLS

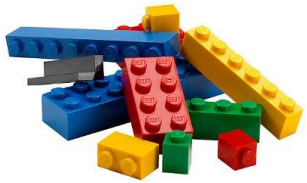
SCHEMES

PRIMITIVES

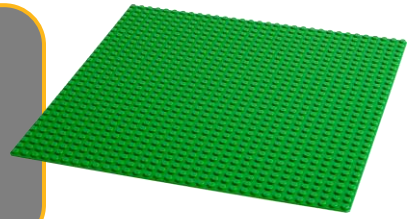
ROOTS OF TRUST
elements in security system for which trust is **assumed** and not **derived**

e.g., TLS 1.3
(network security)

e.g., AES-GCM mode
(authenticated encryption scheme)



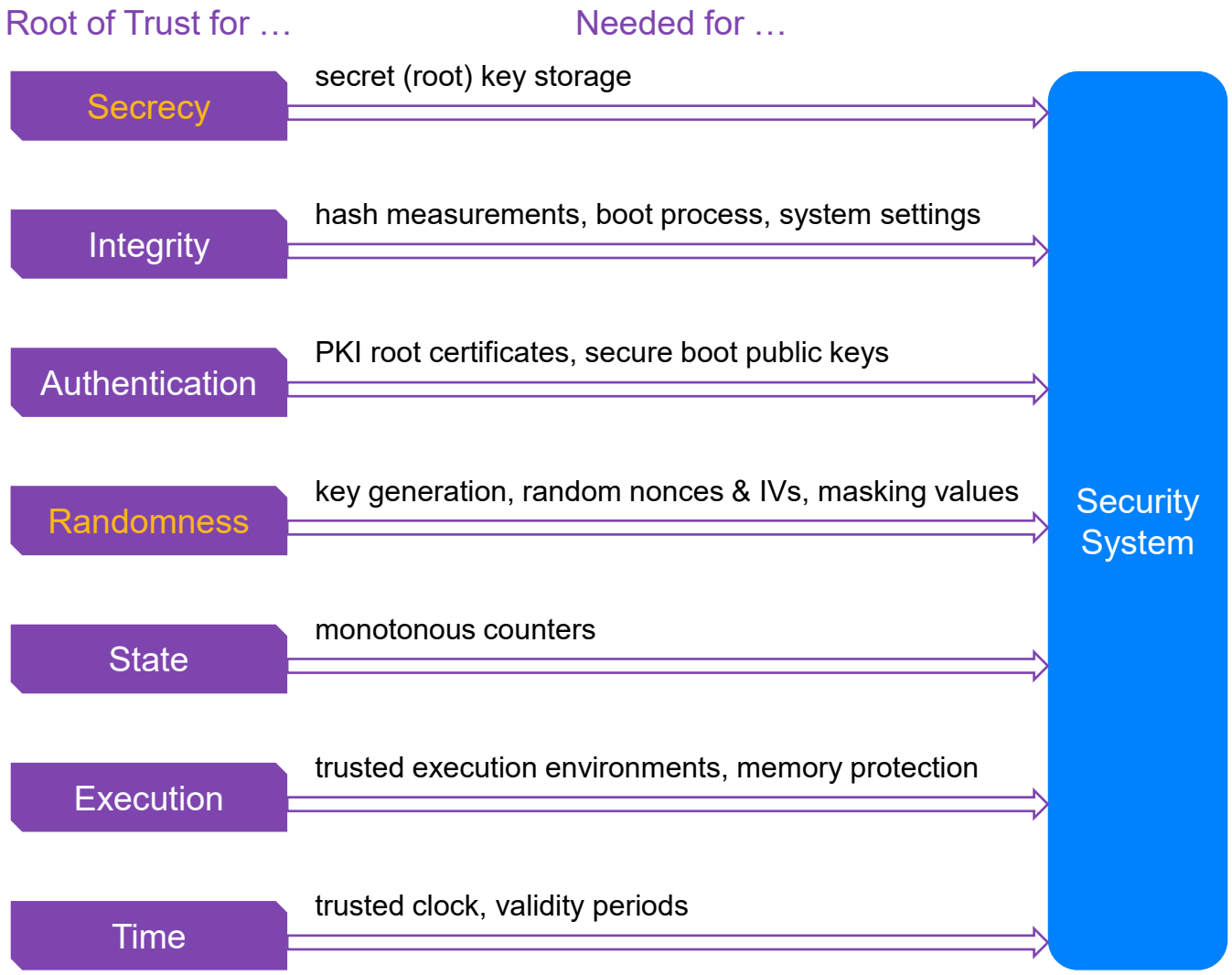
e.g., AES
(block cipher)



e.g., secrecy of keys,
source of randomness,
trusted execution, etc.

Roots of Trust

Categories



Root of Trust

=

element in security system for which trust is assumed and not derived

Roots of Trust

Why? & How?

- Trust in “RoT” is not reducible to another component
  - why do we trust it?
  - trust needs to come from the **implementation** itself (physics)!
  - fundamental RoT → based in **HW (silicon)**

- Security of RoT in silicon?
  - how do we obtain it?
  - based on physical difficulty to inspect a VLSI circuit
  - based on physical difficult to modify a VLSI circuit
  - …
  - based on unpredictable/uncontrollable processes in a VLSI circuit
    - process variations and noise

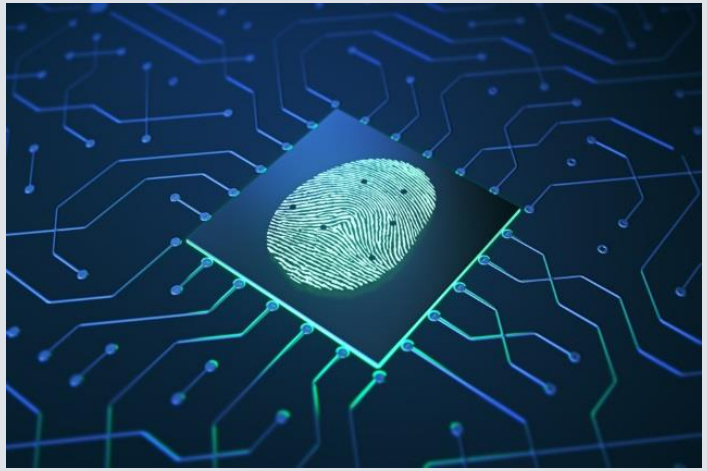
Caveat!

- Physically difficult ≠ impossible !
  - electron microscopes
  - focused ion beams
  - …
- **Perfect security does not exist!**
  - cost
  - countermeasures

Setting the Scene

Physically Unclonable Functions (PUFs)

PUF: Definition



A PUF is a method for evaluating an object's **instance-specific**, **inherent** and **physically unclonable** features

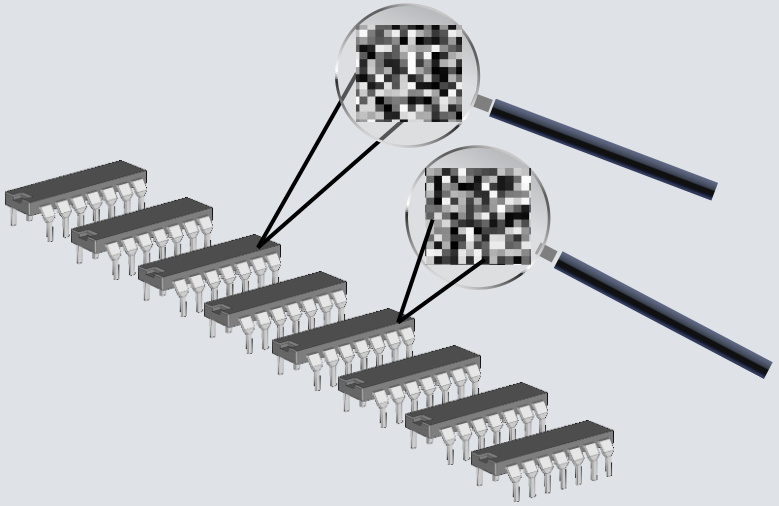
- **instance-specific:** reproducible yet unique to a single instance of the object
- **inherent:** acquired as part of its physical creation process
- **physically unclonable:** infeasible to (ab)use the creation process for creating a clone

PUF is to an object what a biometric is to a person:

- (silicon) PUF = **"silicon fingerprint"**

Silicon Process Variations

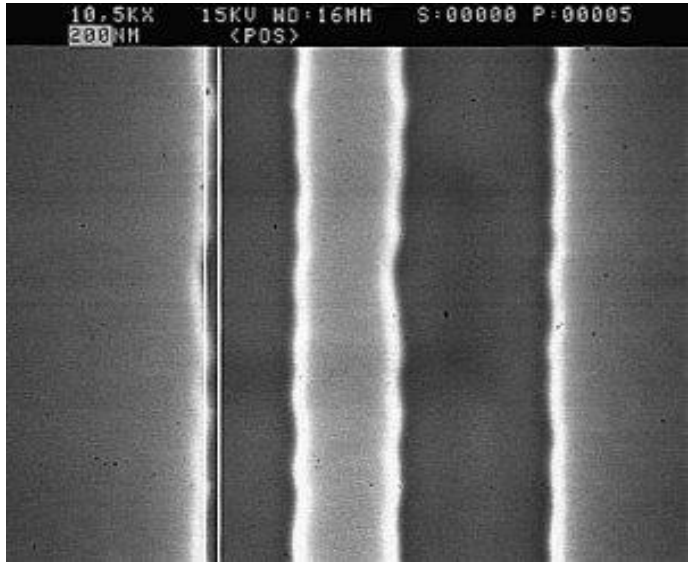
Entropy from Silicon Production



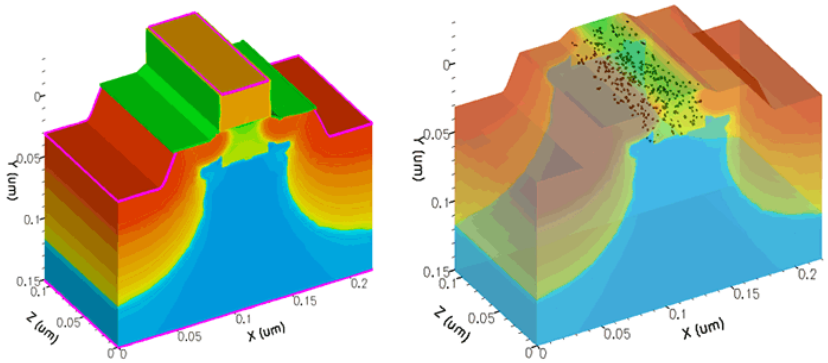
Pelgrom's Law (1989):

- $\sigma^2(\Delta V_{Th}) \sim \frac{1}{WL}$
- ΔV_{Th} = MOS transistor 'mismatch'
- WL = MOS transistor area

Smaller transistors
→ Larger variation on mismatch



Line Edge Roughness
[http://www.microtechweb.com/2d/lw_pict.htm]

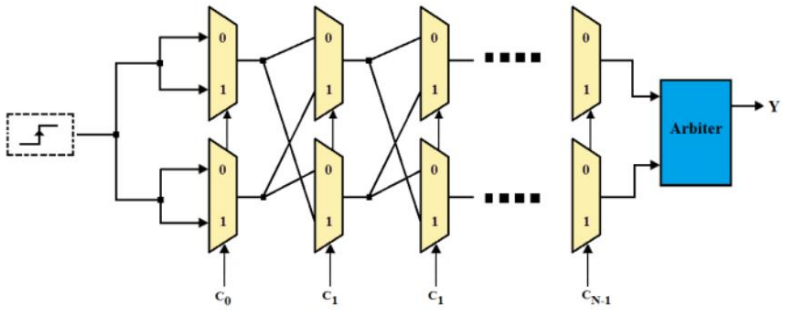


Random Dopant Fluctuation
number and location of dopant atoms
[http://www.intel.com/technology/itj/2008/v12i2/3-managing/3-sources.htm]

Silicon PUFs

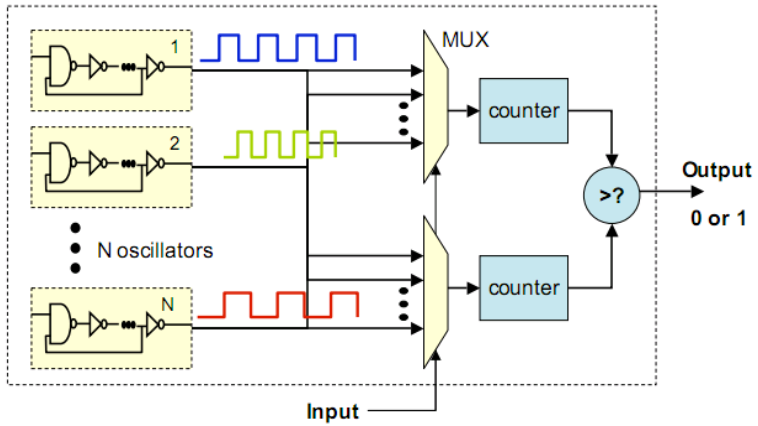
Typical Examples

Arbiter PUF



- Result of race condition between equally designed delay circuits
- Unpredictable delay mismatch due to process variations

Ring Oscillator PUF



- Measuring frequency difference between equally designed ring oscillators
- Unpredictable frequency mismatch due to process variations

SRAM PUF

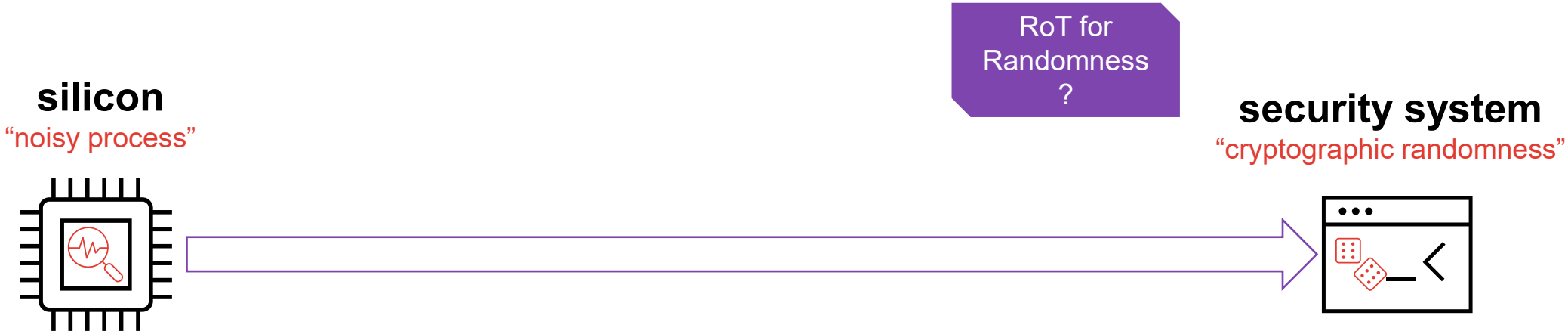
- Power-up state of volatile bi-stable memory cells
- Unpredictable power-up state due to process variations

RoT for Entropy

Framework

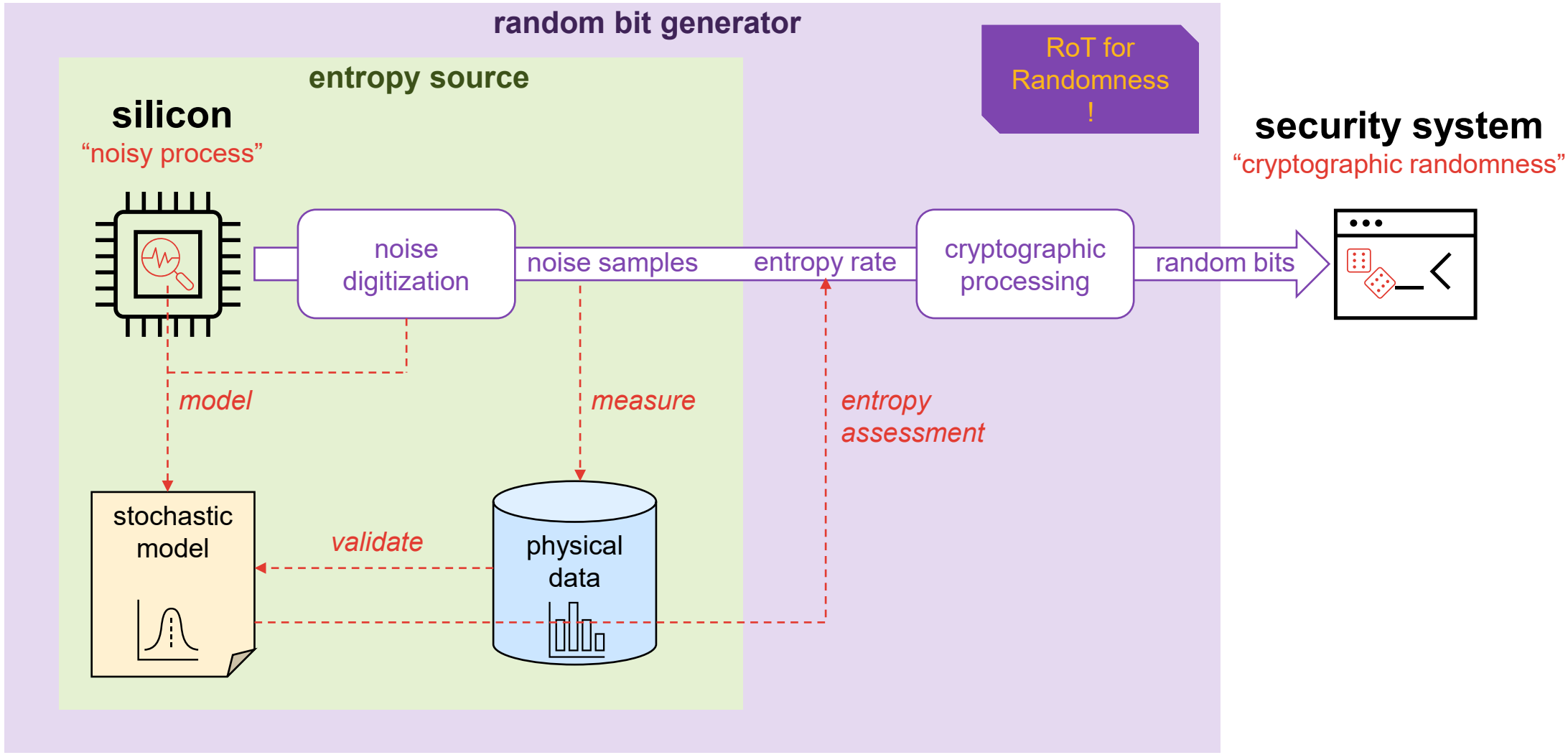
Entropy from Silicon

Approach for Random Bit Generators (e.g., NIST SP800-90 Framework)



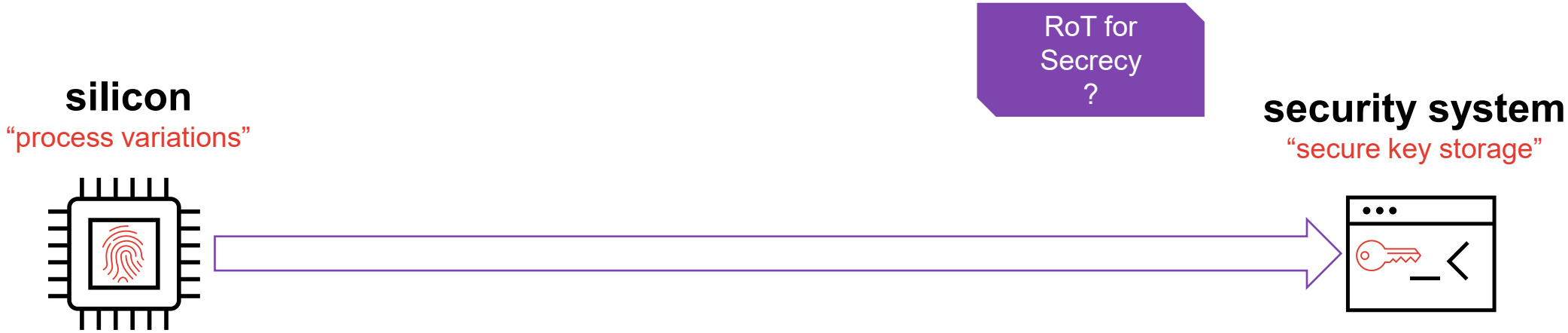
Entropy from Silicon

Approach for Random Bit Generators (e.g., NIST SP800-90 Framework)



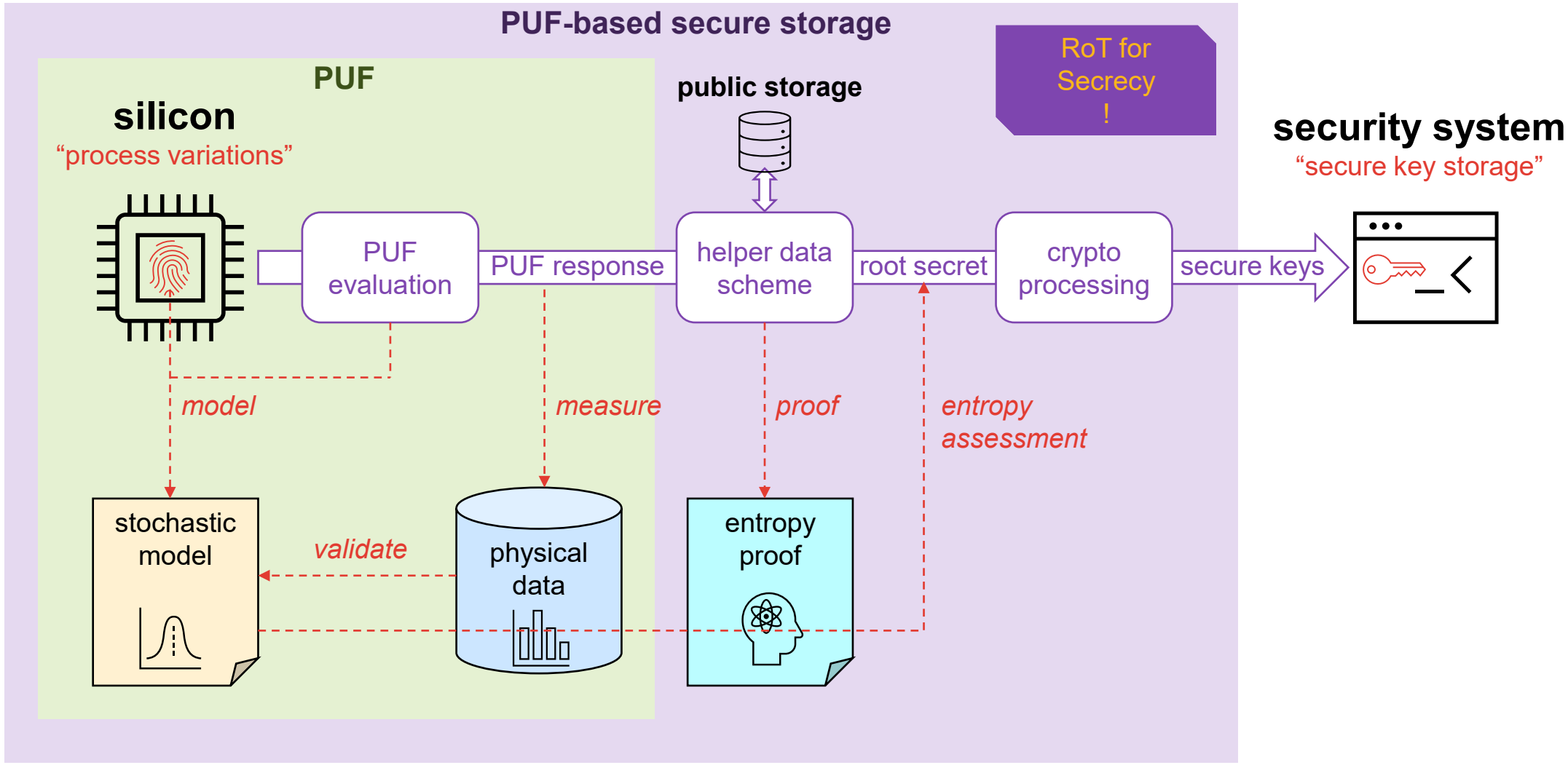
Entropy from Silicon

Approach for PUF-based Key Generation & Storage



Entropy from Silicon

Approach for PUF-based Key Generation & Storage



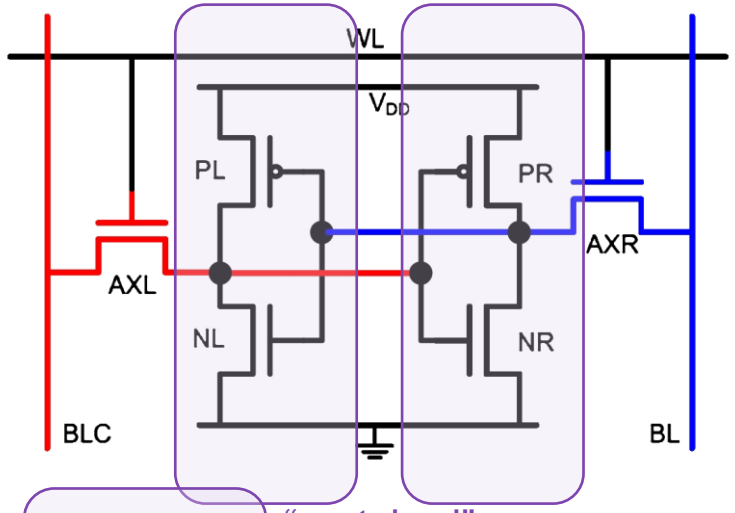
SRAM PUF

Operation & Stochastic Model

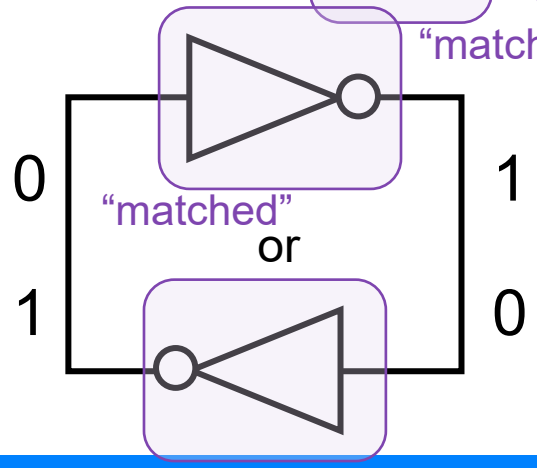
SRAM PUF

Operation

Designed to be matched

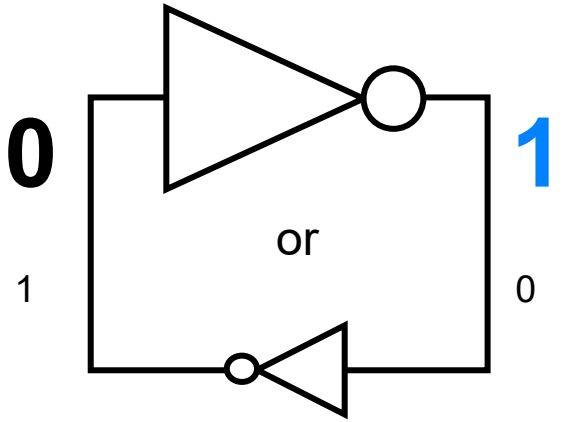


"matched"

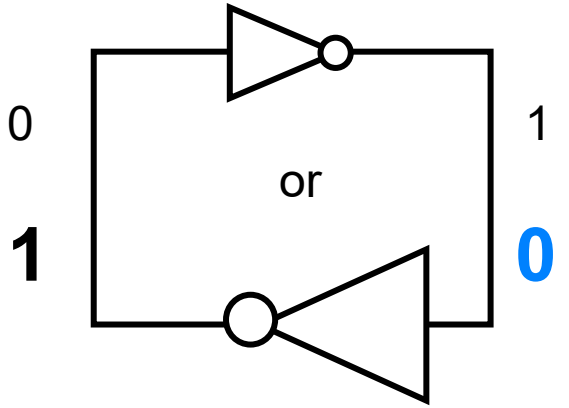


Preferred power-up state = ?
Pr(0) = 50%
Pr(1) = 50%

Process variations → Mismatch!



Preferred power-up state = 1
Pr(0) = 5%
Pr(1) = 95%



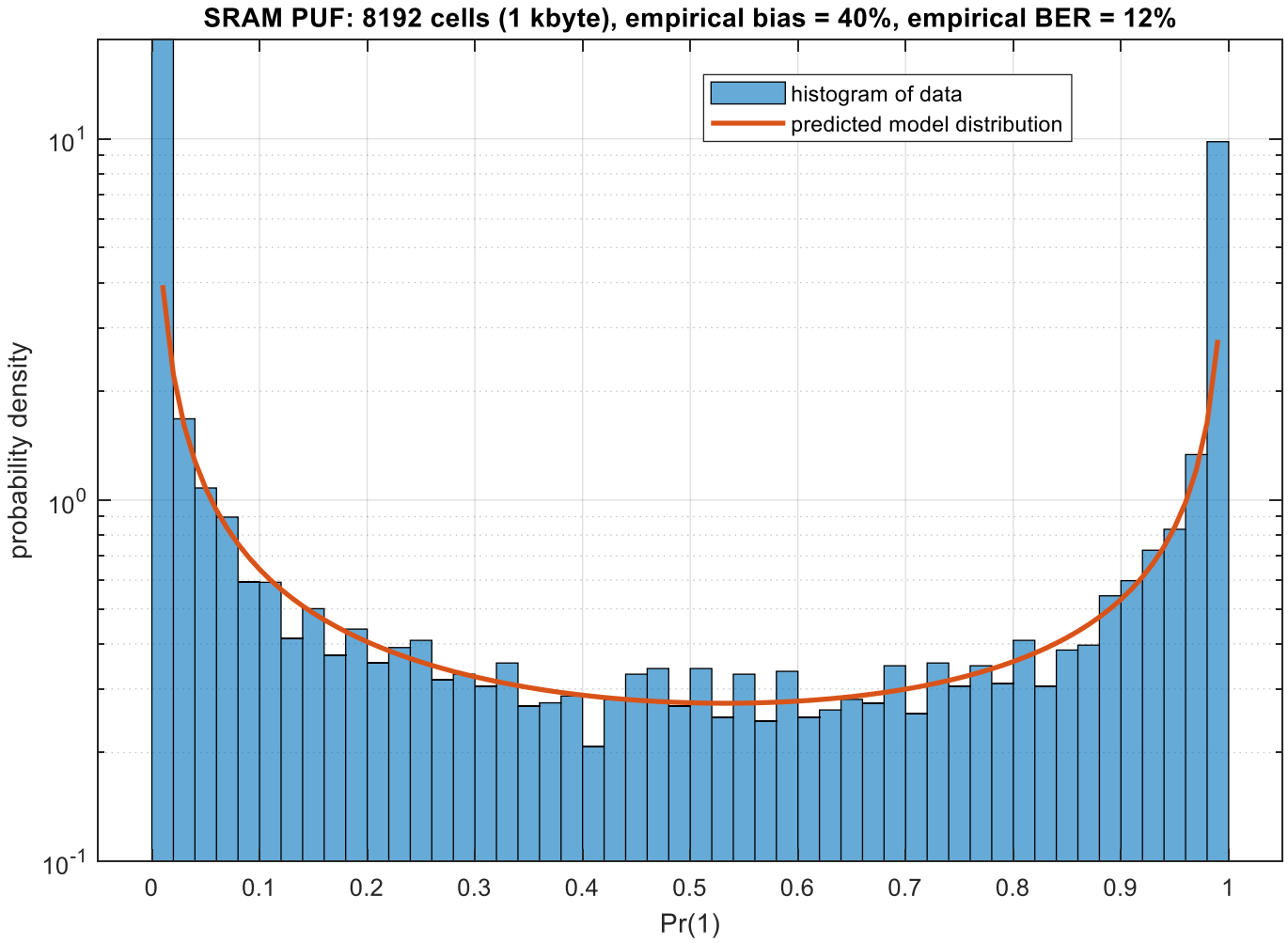
Preferred power-up state = 0
Pr(0) = 70%
Pr(1) = 30%

Every SRAM cell has its own specific power-up state probability determined by process variations

SRAM PUF

Stochastic Model (sketch)

- SRAM PUF cell distribution
  - strength of cell preference determined by strength of mismatch, and noise
  - distributions of mismatch and noise → distribution of preferences:
  - distribution can be skewed: bias !
- SRAM PUF cell independence
  - cell power-up preference = '*local differential measurement*' of mismatch between neighboring transistors
  - cells' preferences do not influence each other → preferences are independent



SRAM PUF

Validation

SRAM PUF Uniqueness

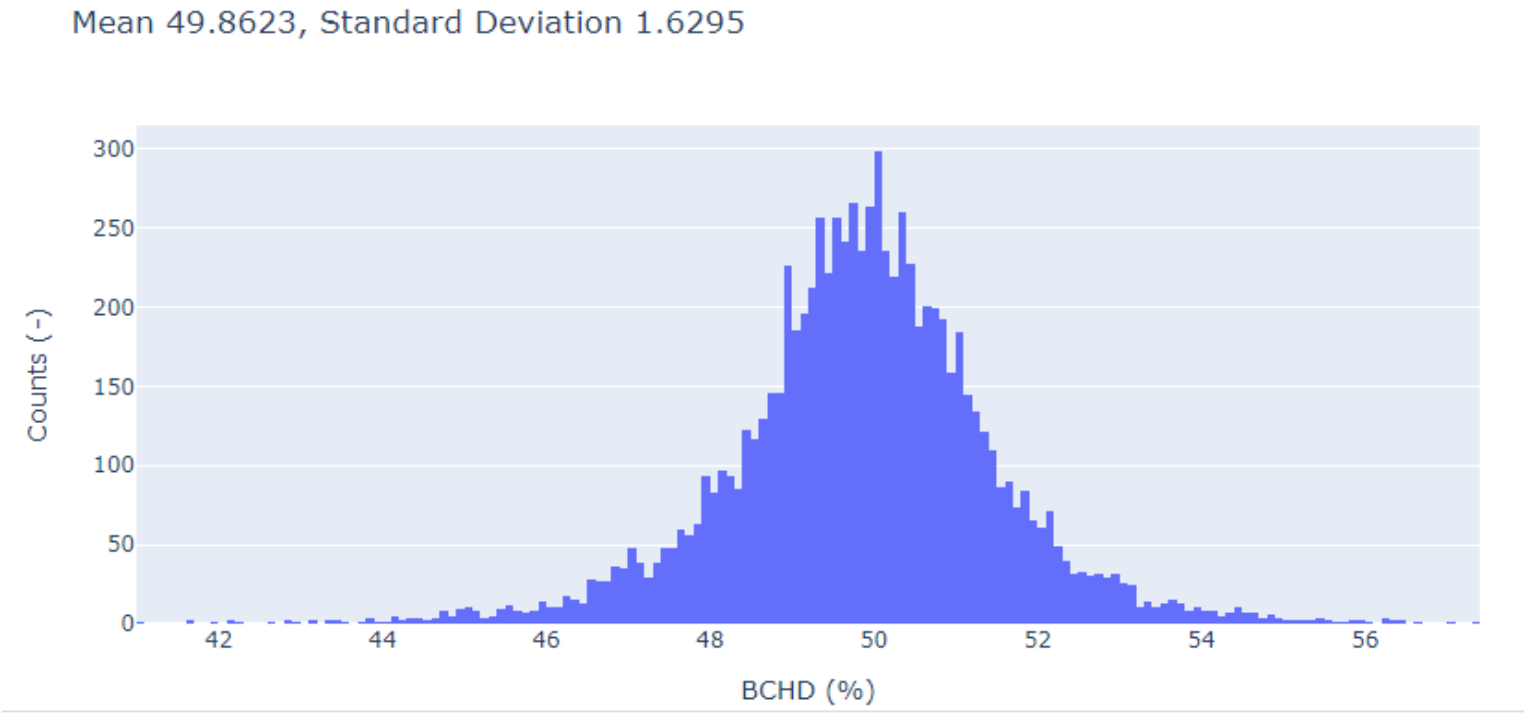
Random cell evaluations → Random differences ‘between’ devices (inter)

SRAM PUF Uniqueness

Empirical Data

- Set of 130 different devices:
  - 130x129/2 = 8,385 pairs of devices
    → 8,385 inter-distances
- Compute them all (relative) and derive histogram
- Closely centered around 50% (ideal uniqueness)
  - ~ binomially distributed

Relative Inter-Distance at 25°C (histogram)



SRAM PUF Reproducibility

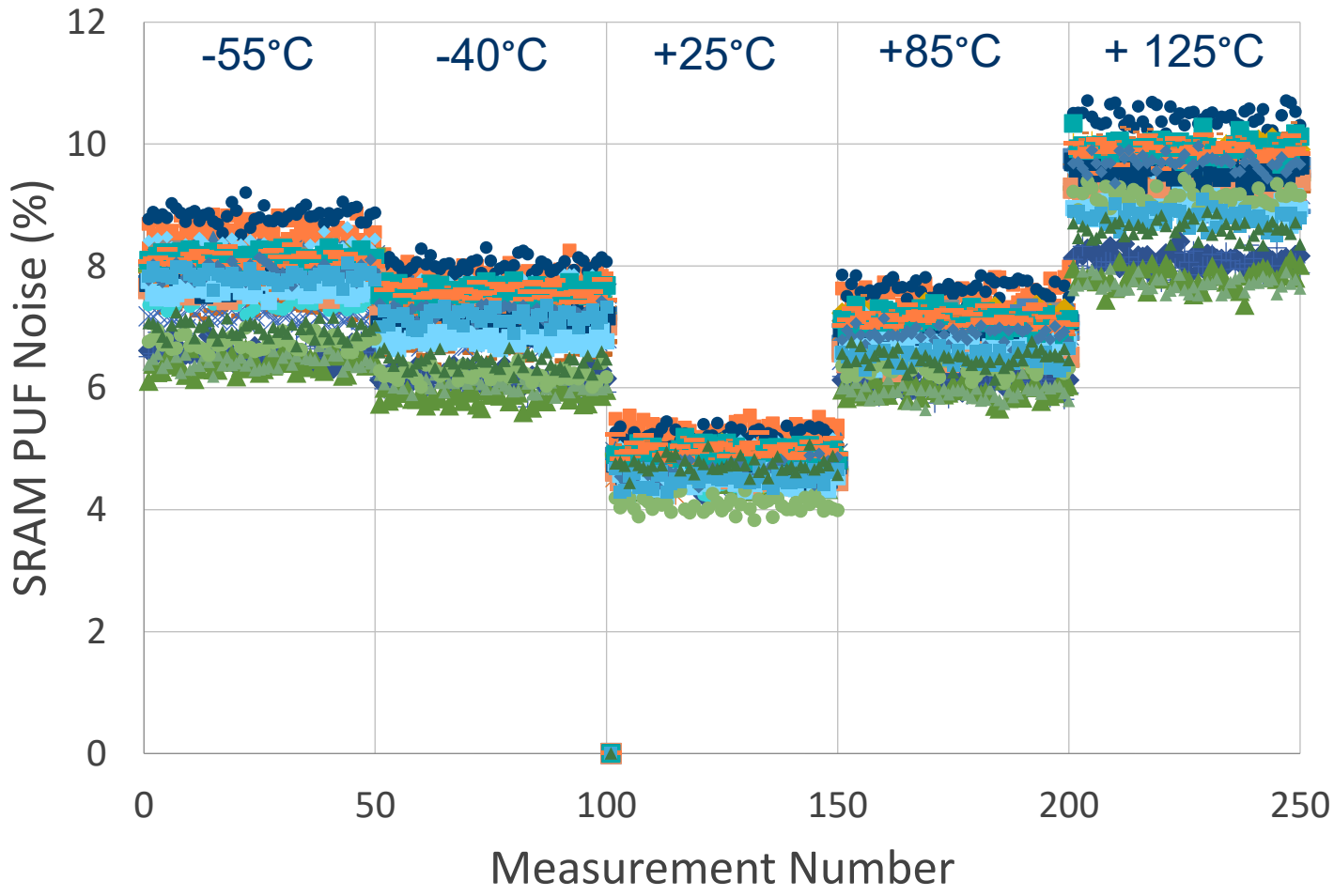
Noise → Random differences ‘inside’ a single device (intra)

~ 10% errors

SRAM PUF Reproducibility

Empirical Data – Temperature

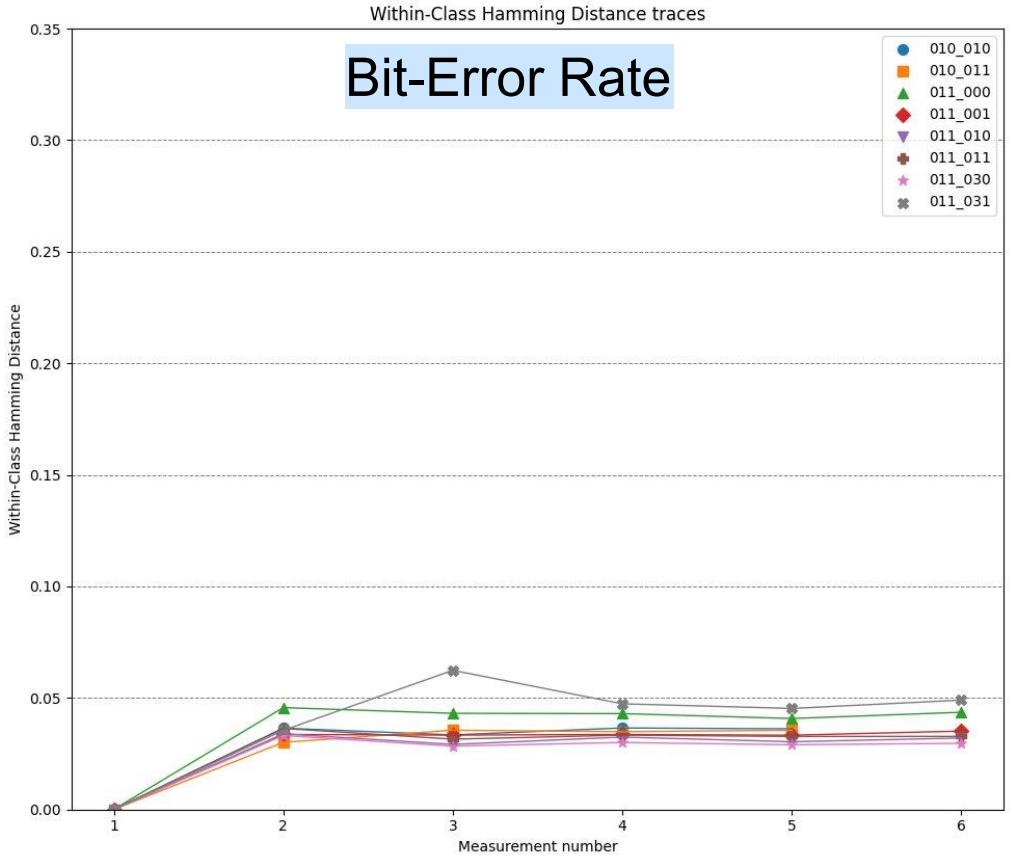
- Main influencing external condition is temperature
- Noise at room temperature ~ 4% to 6%
- Intra-distance (Hamming) with respect to enrollment condition (at 25°C) is highest at extreme temperatures

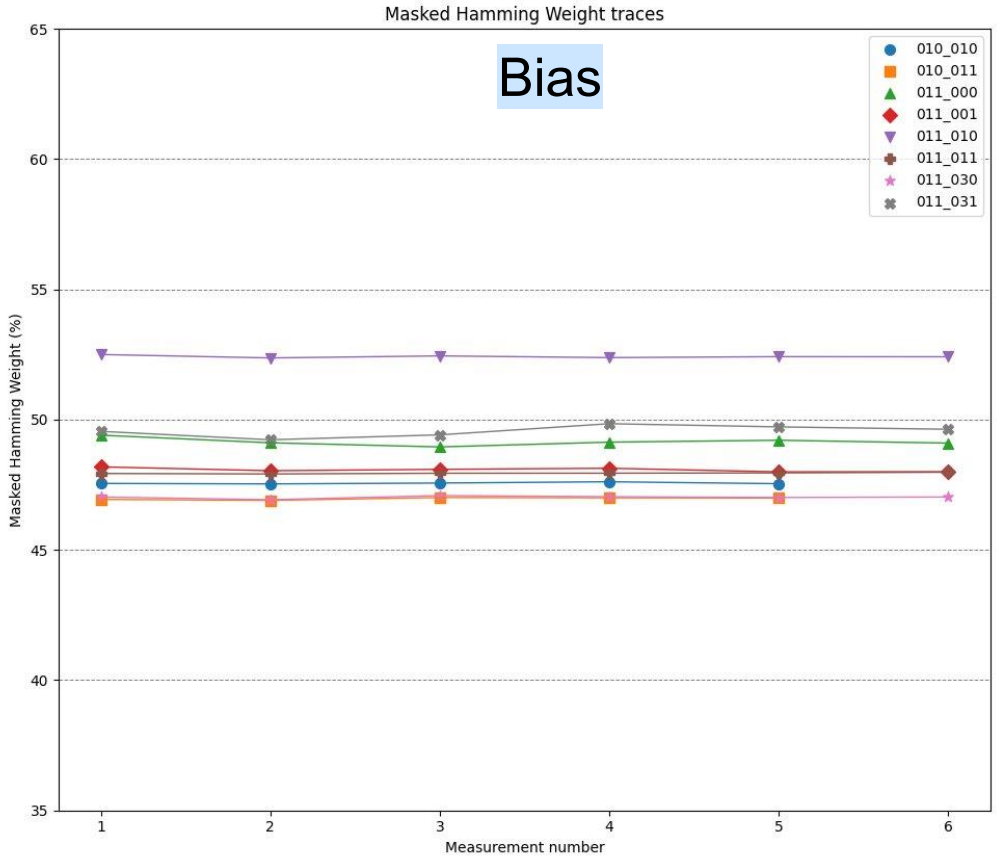


SRAM PUF Reproducibility

Empirical Data – Silicon Technology

Some very recent results from an N2 SRAM test chip



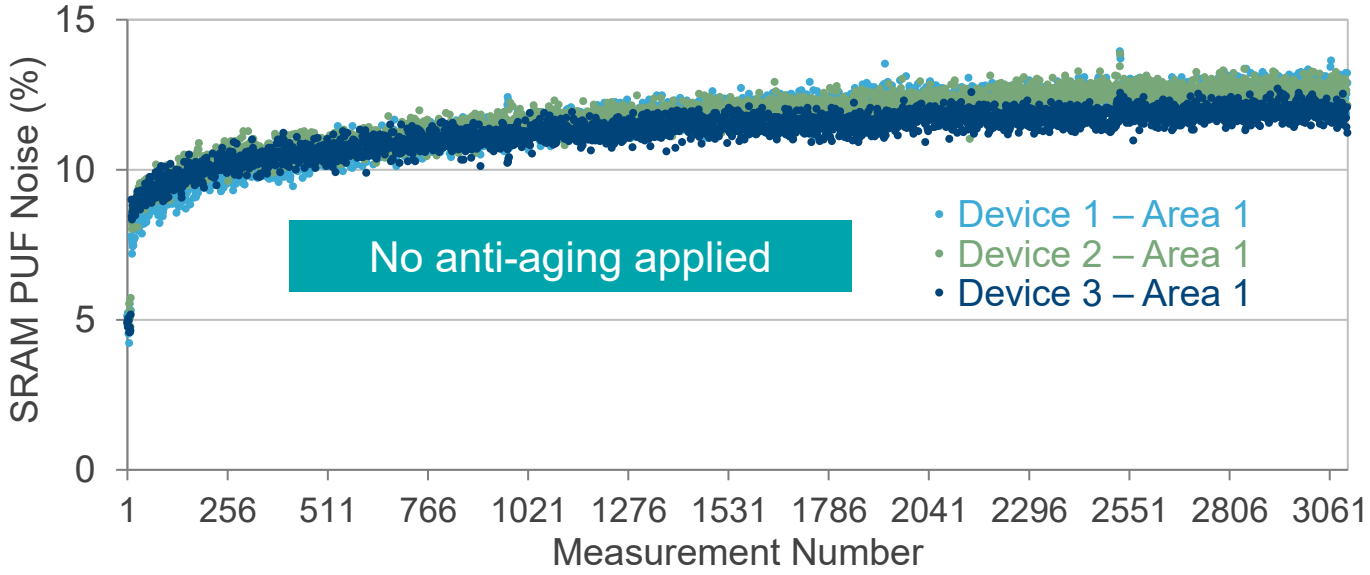


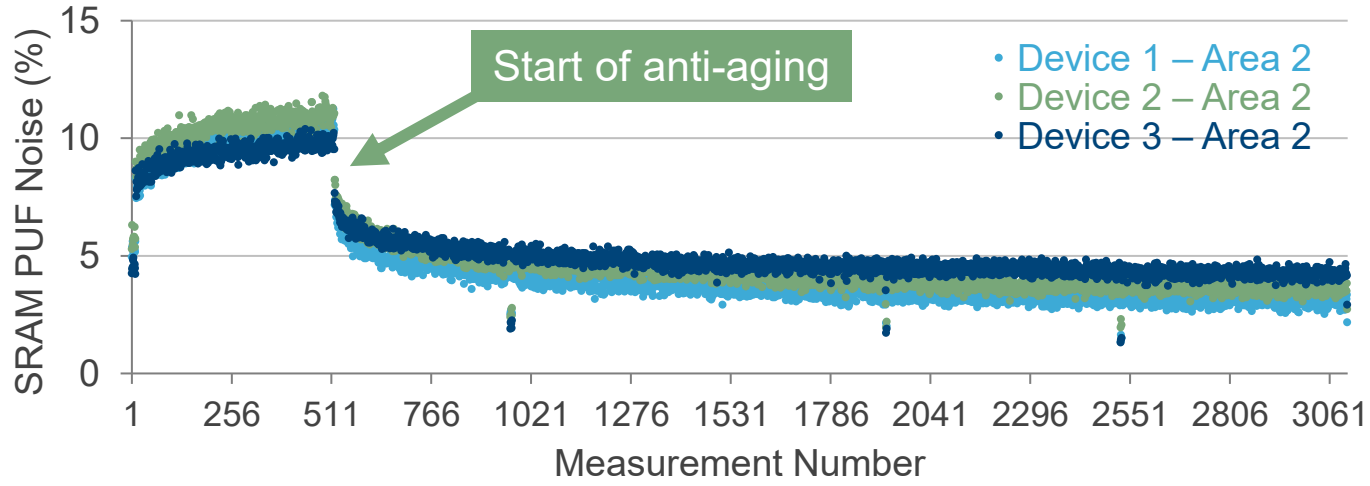
High-quality SRAM PUF behavior (low noise, limited bias) tracks very well over technology nodes!

SRAM PUF Reproducibility

Empirical Data – Silicon Aging

- NBTI (Negative Bias Temperature Instability) is the main aging mechanism for SRAM PUF
- SRAM PUF aging is data-dependent!
  - storing power-up value → noise increases
  - storing inverted power-up value → noise decreases! "anti-aging"
- Accelerated ageing experiment running for 130 days at 80°C , VDD+10% → simulates 5Y of ageing

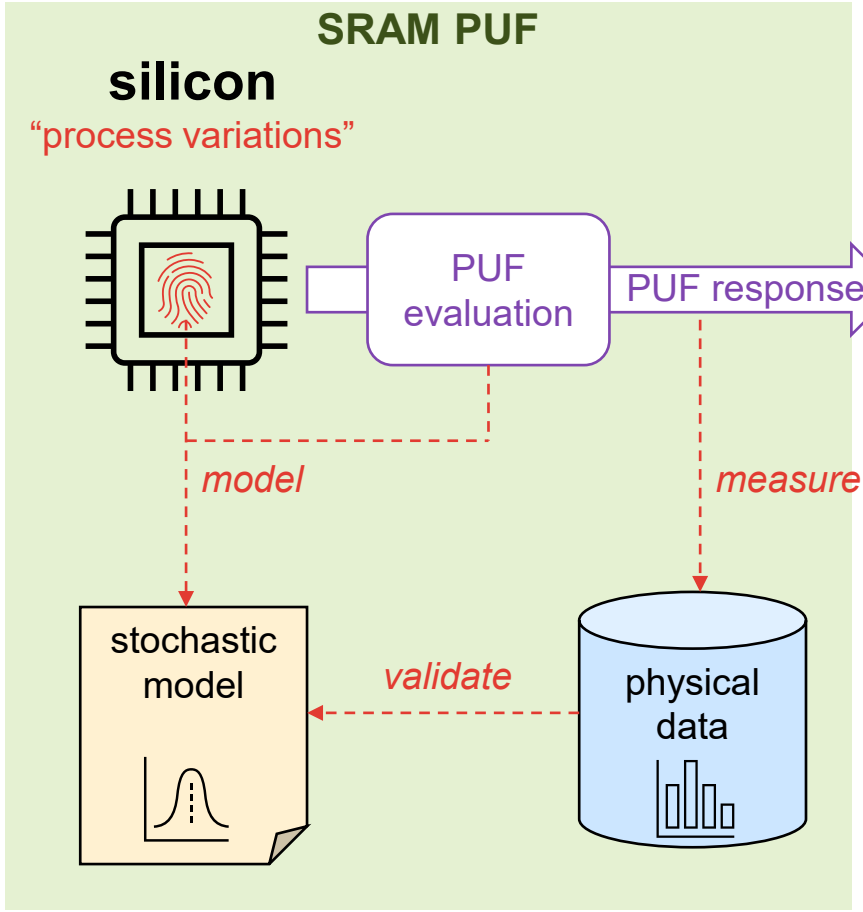




SRAM PUF

Model & Validation Overview

- SRAM PUF operation results in stochastic model with
  - predicted distribution of cell preferences
  - predicted independence between cell preferences
- Strong validation of model based on physical data
  - histograms closely match predicted distributions
  - data passes independence tests



- Continuous empirical data collection across all conditions
  - temperatures
  - CMOS technologies
  - silicon lifetimes
  - …
- SRAM PUF quality very consistent under all conditions
  - limited % of bit errors: range 10-15%
  - limited bias: range 30-70%

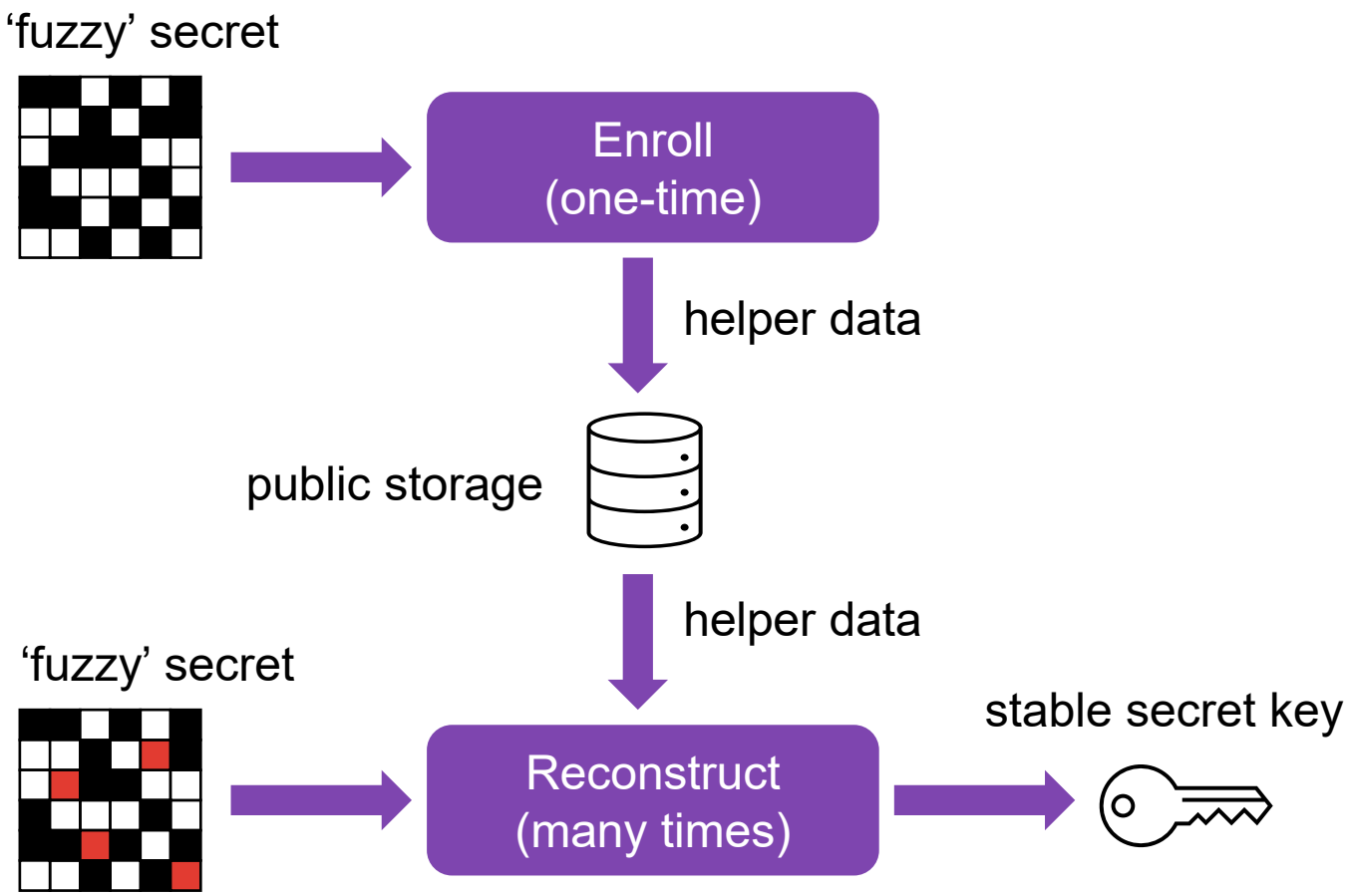
Helper Data Schemes

Concept & Construction

- Fuzzy secret
  - partially unpredictable: level of entropy
  - partially stable: level of noise
- Helper Data Scheme (HDS)
  - turns a fuzzy secret into a secret key
  - uses **helper data**
    - one-time enrolled
    - publicly stored
    - used for reconstruction
  - **Reliable:** key = fully stable Pr(wrong key) ≈ 0%
  - **Secure:** key = fully unpredictable H(key | helper data) = 100%

Helper Data Scheme (HDS)

Concept

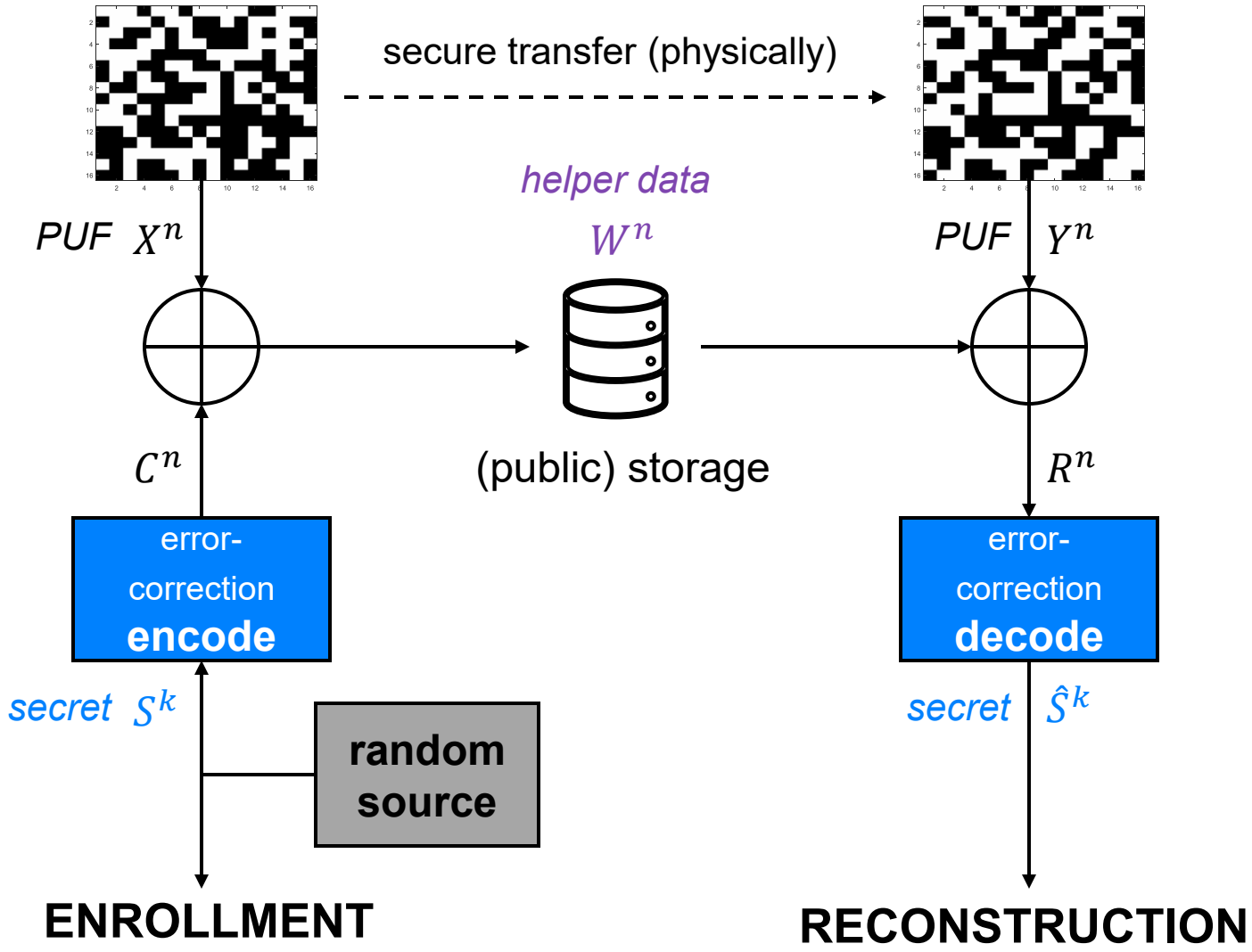


Both key reproducibility and unpredictability need to be proven, under the **stochastic model** of the fuzzy secret!

Fuzzy Commitment [Juels & Wattenberg 1999]

A Practical Helper Data Scheme

- Fuzzy Commitment is a HDS based on error-correcting codes
- Reliable?
  - yes, if a code is chosen which can correct max. # of expected bit errors
  - derive/simulate based on stochastic model!
- Secure?
  - secret key is randomly generated
  - proof: helper data does not 'leak' about secret key **IF** fuzzy secret is full entropy!
  - what does the stochastic model tell about the fuzzy secret entropy?



Helper Data Schemes

Entropy Proof

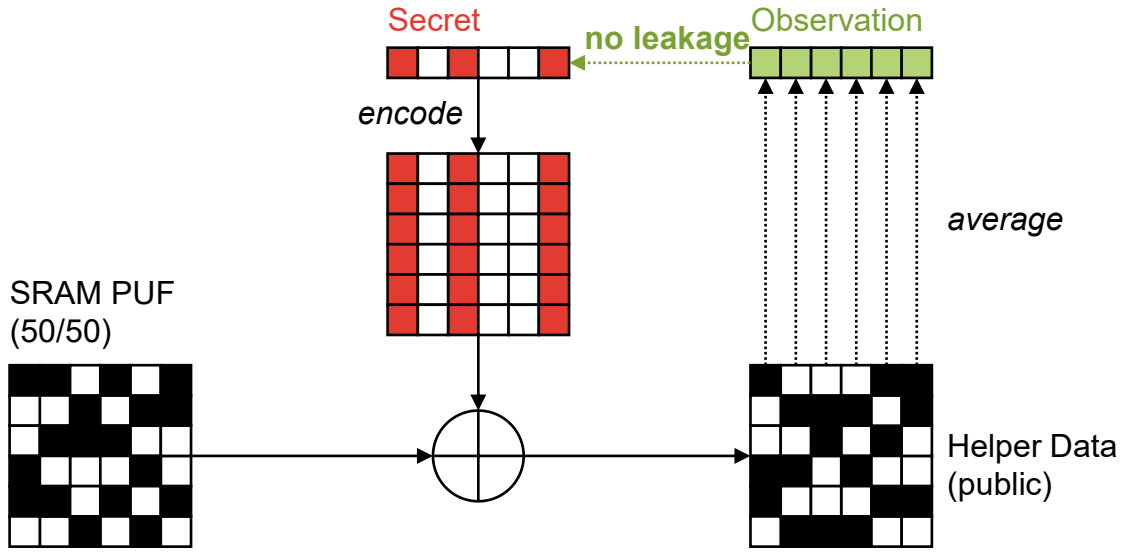
Fuzzy Commitment

Helper Data Leakage in Case of Bias

- **Unbiased** SRAM PUF has a **50/50** probability of 0 and 1 bits
  - No information about secret is contained in helper data:

$$I(Helper\ Data; Secret) = 0$$

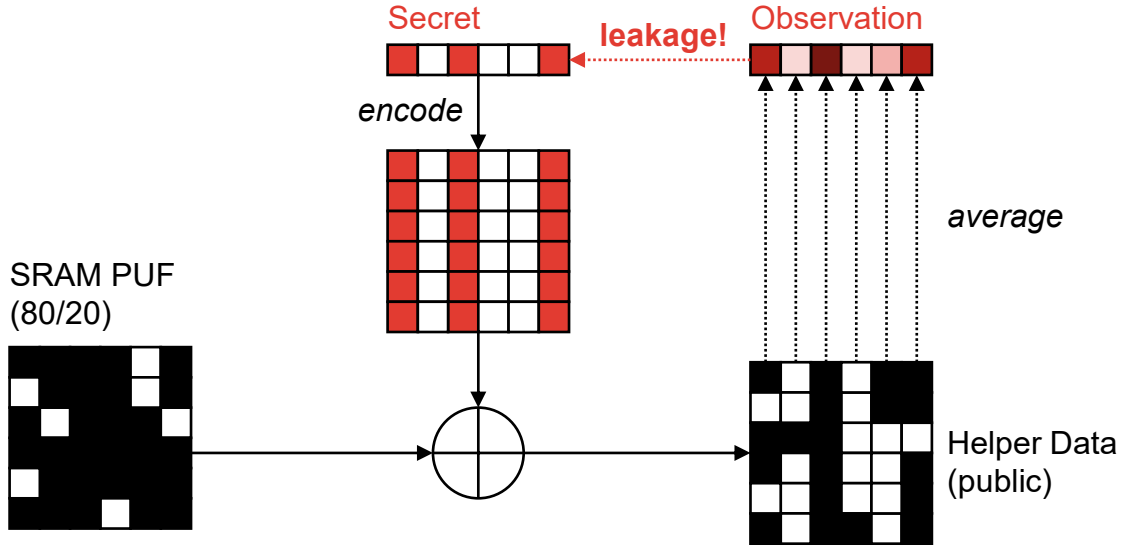
  - Even an unbounded adversary learns nothing: **perfect secrecy**

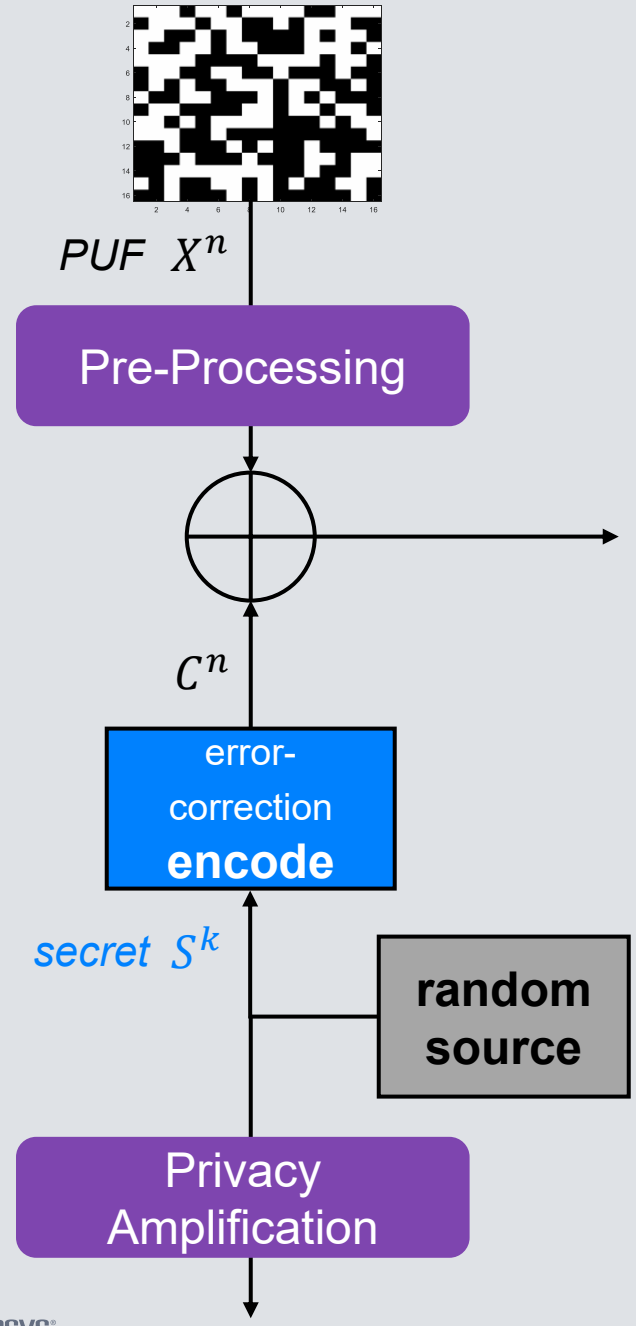


- **Biased** SRAM PUF has imbalanced 0 and 1 bits, e.g., **80/20**
  - Observations in helper data leak information about secret:

$$I(Helper\ Data; Secret) > 0$$

  - Worst-case: full secret exposed





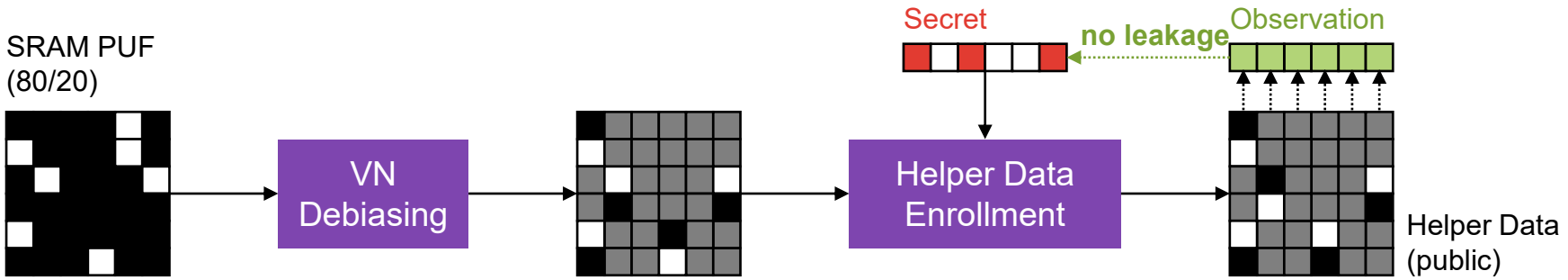
Fuzzy Commitment + SRAM PUF

Dealing with Bias

- SRAM PUF stochastic model predicts bias !
  - Fuzzy commitment not usable as-is
  - Modify it to account for bias in the entropy proof
- Option 1: Privacy Amplification
  - Extract the “remaining entropy” (if any) from the secret
  - Using a strong extractor → comes at a cost
  - Only works for weak bias (45-55%)
- Option 2: Pre-Processing
  - Remove the bias before fuzzy commitment → no leakage
  - Using a Von Neumann extractor
  - Impacts error-correcting → comes at a cost
  - Works for strong bias (10-90%)

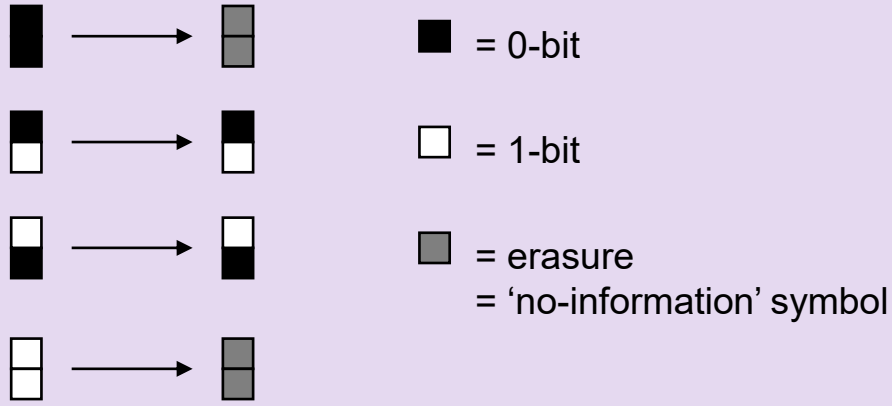
Fuzzy Commitment + Von Neumann Debiasing

How it Works?



- **Debiasing** = removing PUF bias prior to helper data generation
  - Results again in **perfect secrecy**
- Debiasing based on well-known **Von Neumann extractor**
  - Retain unequal bit pairs → these always occur with 50/50 probability and **do not leak**
  - Erase equal bit pairs → erased symbols carry no information and **cannot leak**

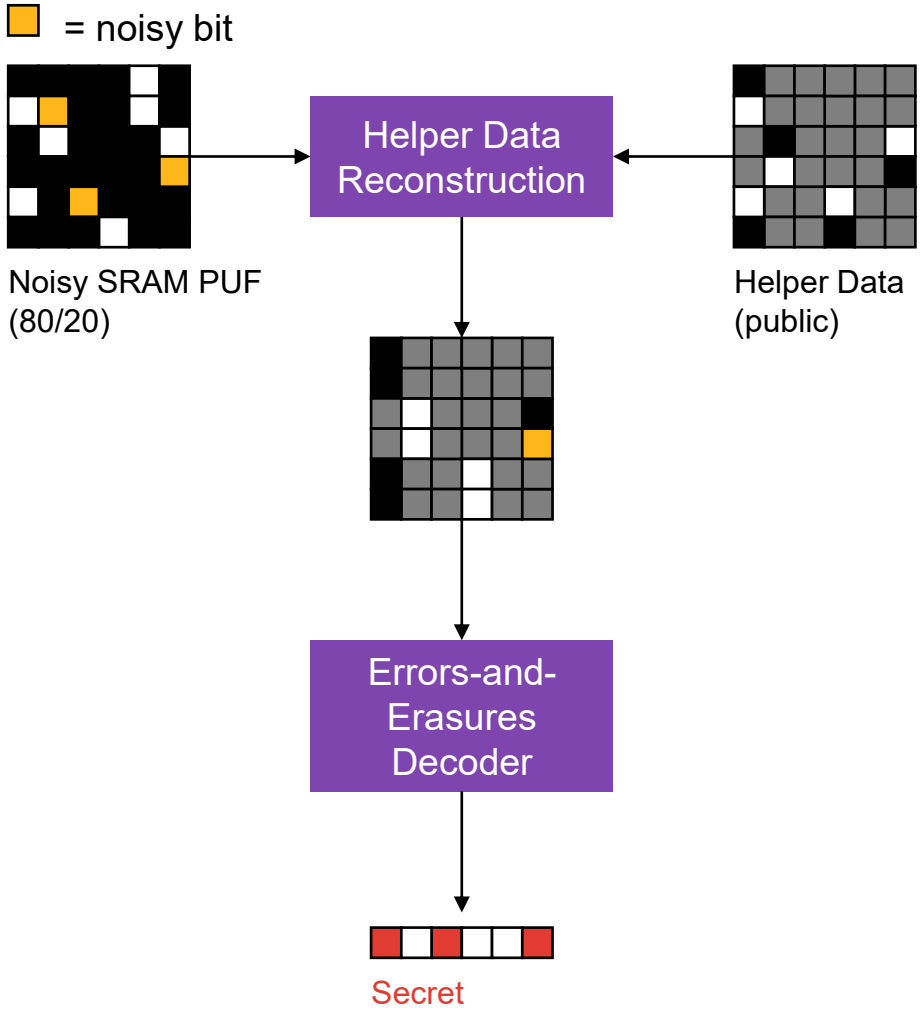
Von Neumann-based Debiasing



Fuzzy Commitment + Von Neumann Debiasing

Effect on Security & Reliability

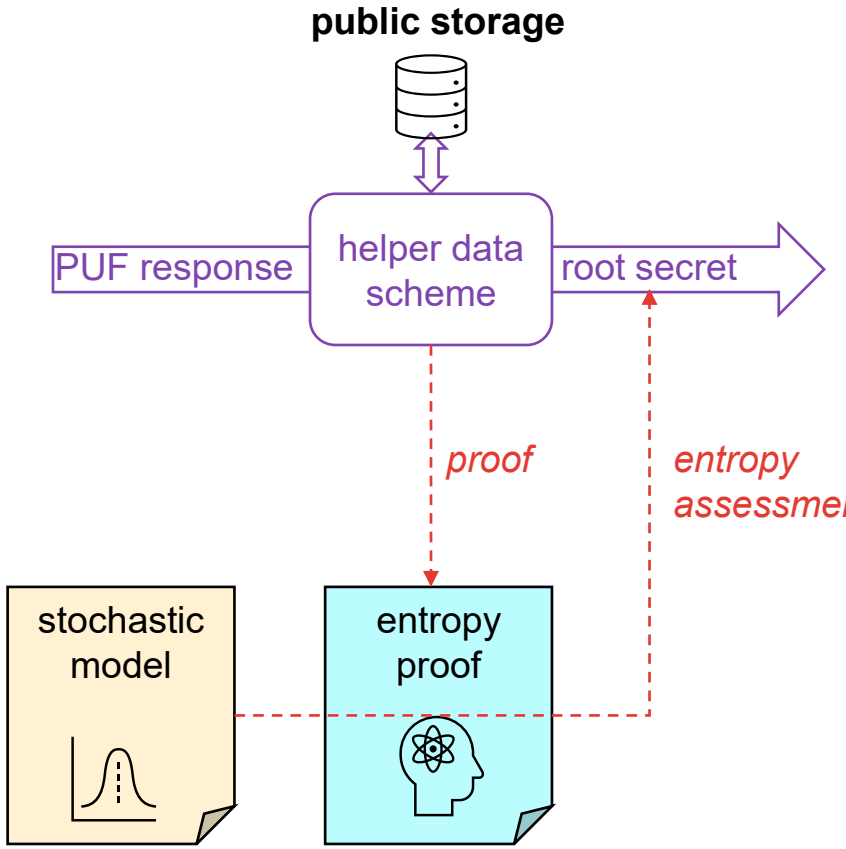
- Bias only affects reliability, never security
  - Von Neumann-based debiasing **always prevents leakage, regardless of bias level**
  - Stronger bias → more equal pairs
    → more erased symbols
    → less information in helper data
    → more difficult to decode secret
  - Requires errors-and-erasures decoder
  - Requires powerful error-correcting code
- Trade-off security / reliability
  - secure **at any bias level**
  - for very strong bias → reduced reliability



Fuzzy Commitment as Helper Data Scheme

Overview

- Fuzzy commitment algorithms designed based on (validated) stochastic model of fuzzy secret
  - Select error-correcting code to deal with max. # errors
  - Analyze entropy of fuzzy secret and assess impact of helper data leakage



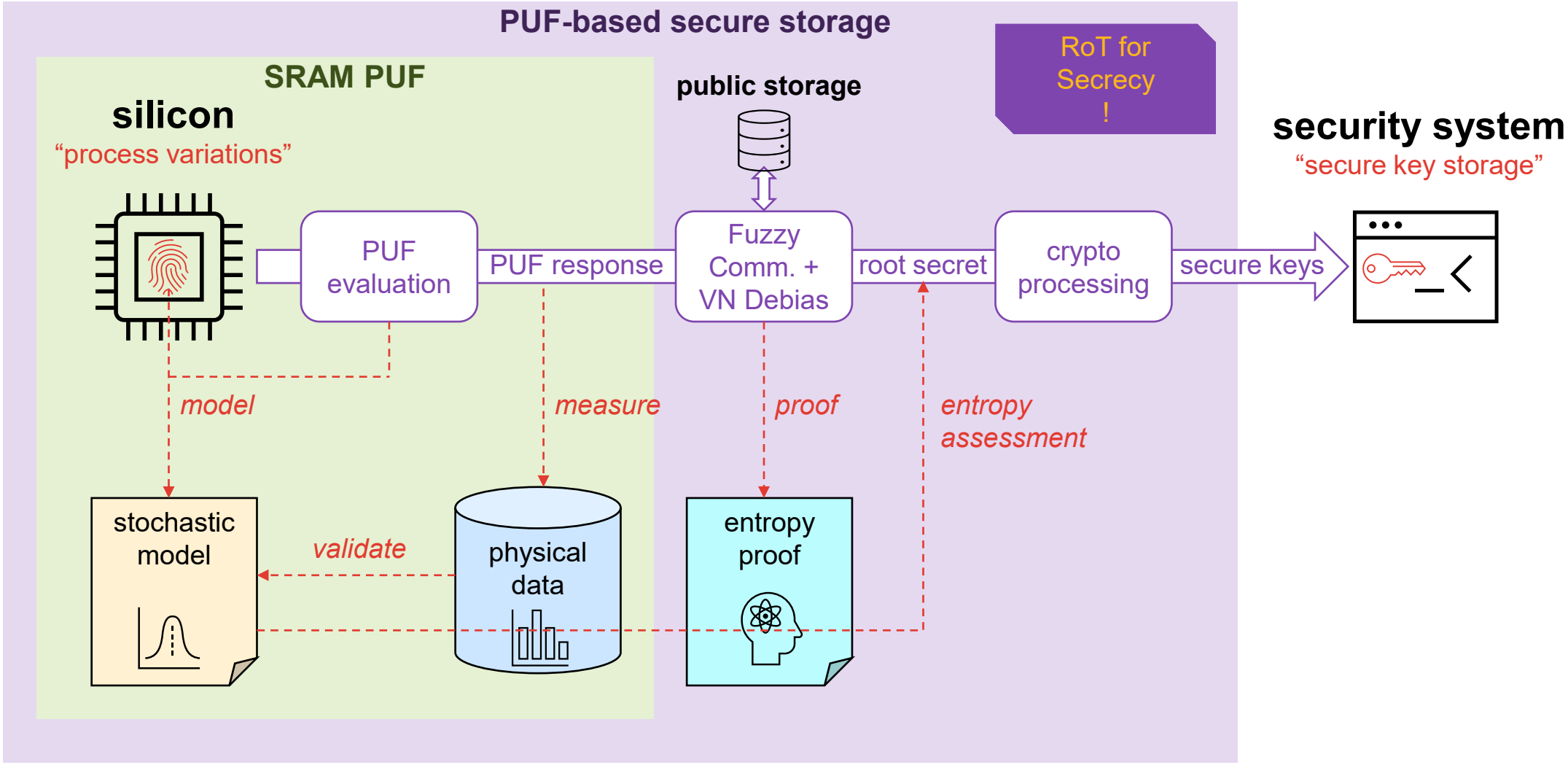
- Fuzzy commitment proven secure (perfect secrecy):
  - no bias: as is
  - for weak bias: with privacy amplification
  - for any bias: with Von Neumann debiasing
- Von Neumann debiasing:
  - secure for any bias level
  - worst-case: trade-off reliability for security
  - also proven secure when the same fuzzy secret is enrolled multiple times (many helper data sets available)

Synthesis

PUF-based Key Generation & Storage

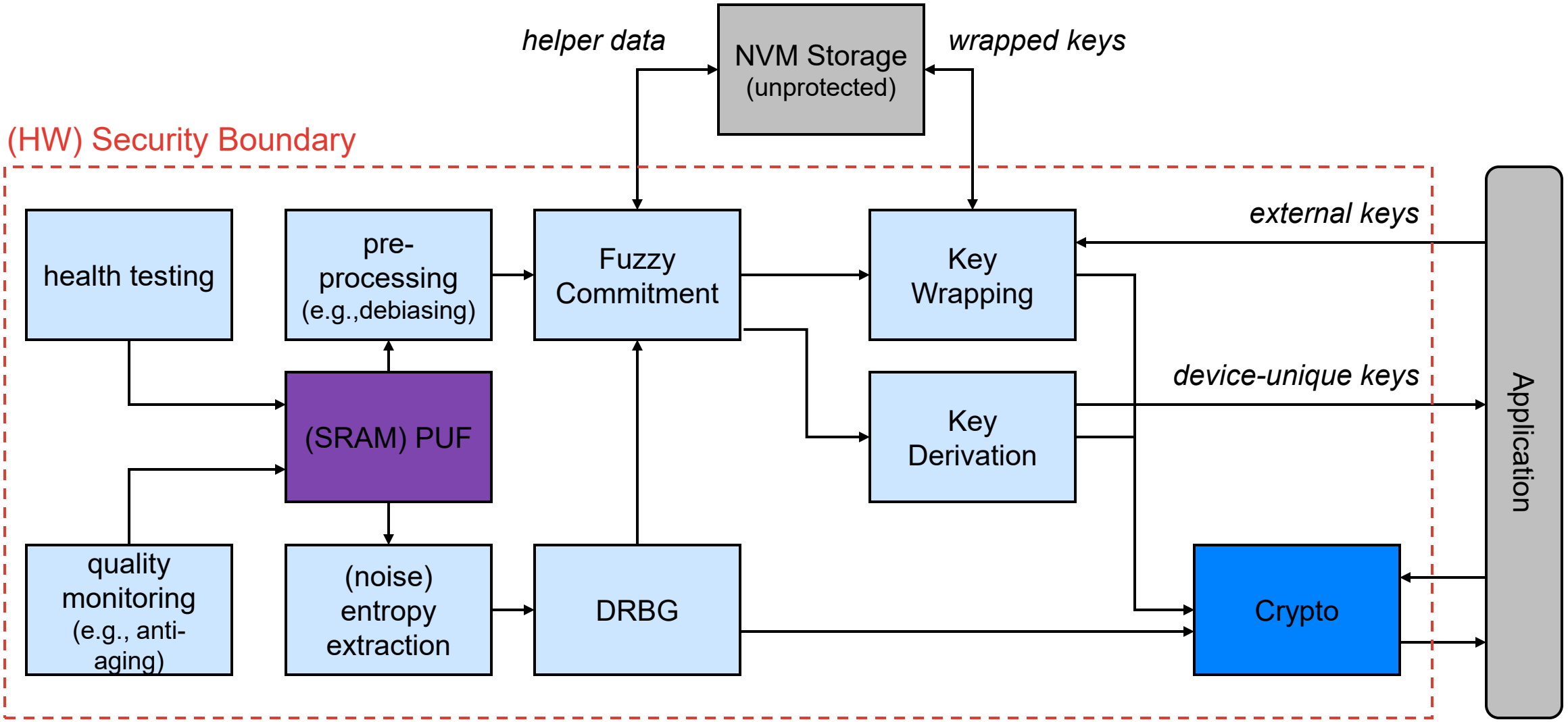
Revisiting: Entropy from Silicon

Approach for PUF-based Key Generation & Storage



PUF-based Root of Trust

Security Architecture



Take-Away Points

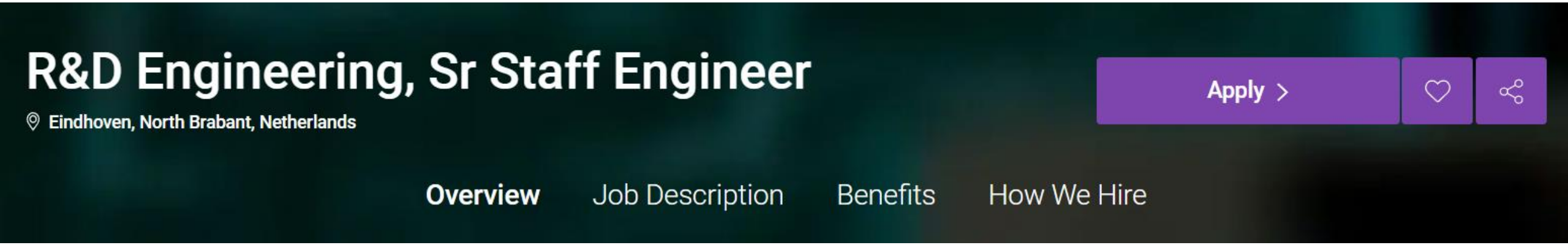
- PUFs are strong Roots-of-Trust for Entropy in a security system
  - entropy originating from the silicon
  - process variations for secrecy: PUF-based key generation & storage
  - noise for randomness: PUF-based RBG

- Framework for studying PUF-based key generation & storage is needed
  - equivalent to for RBGs (NIST SP800-90)
  - stochastic model validated by physical data
  - entropy proof for helper data scheme!

- SRAM PUF is basis for practical Root-of-Trust for Secrecy
  - SRAM PUF stochastic model well understood and validated
  - Fuzzy commitment with Von Neumann-based debiasing = resilient and provable secure helper data scheme

Thank You for Your Attention

Any Questions



https://careers.synopsys.com/