INTRINSIC ID

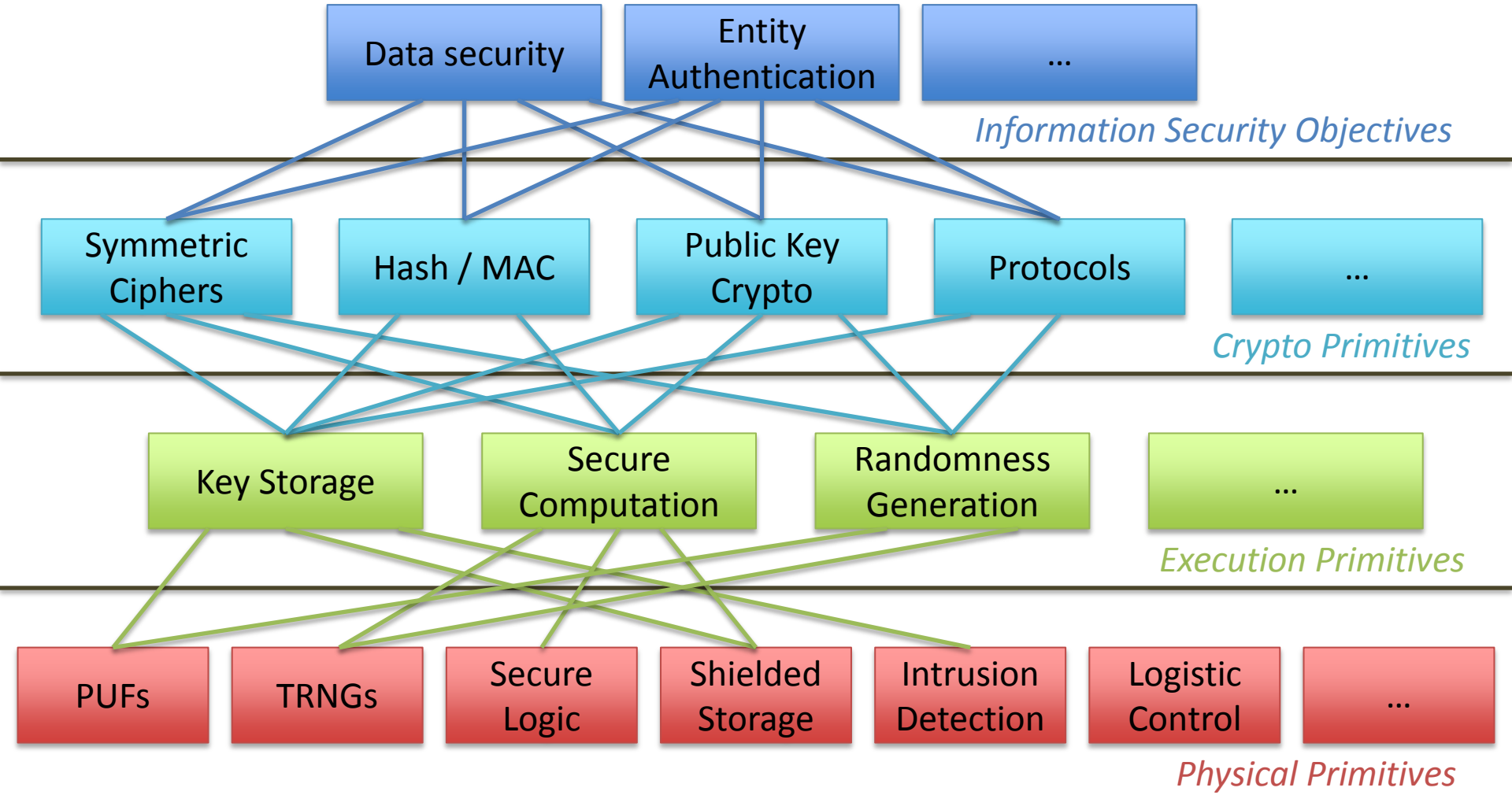
Secure your digital life™

Silicon PUFs and PUF-based Key Storage

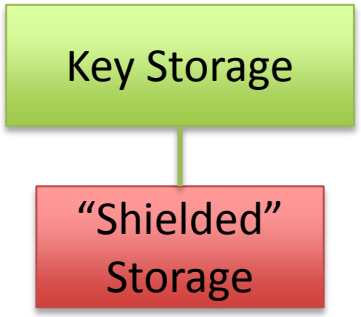
Roel Maes
Intrinsic-ID, Eindhoven (NL)

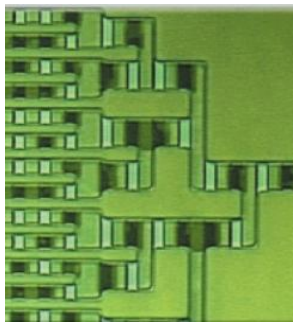
June 6, 2014
Summerschool:
Design and security of cryptographic algorithms and devices for real-world applications
Šibenik, Croatia

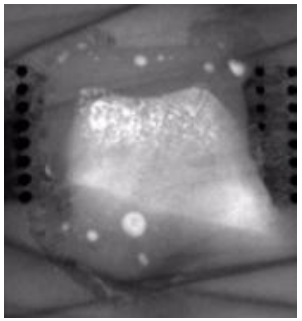
Roots of Trust

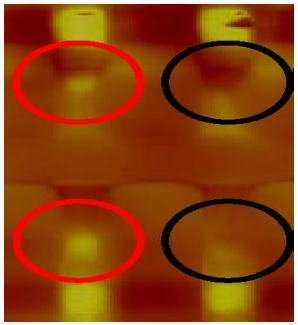


Physical Key Storage

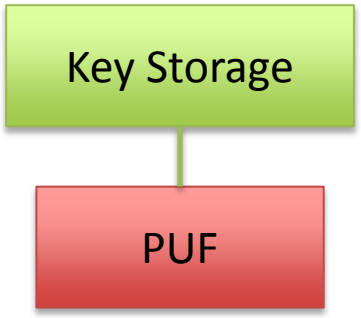



ROM


Fuses


Flash


Anti-fuses



- Alternative to NVM-based key storage: PUF-based key storage
- Main advantages:
  - Key not present when device is powered down
  - Key depends on device intrinsic randomness

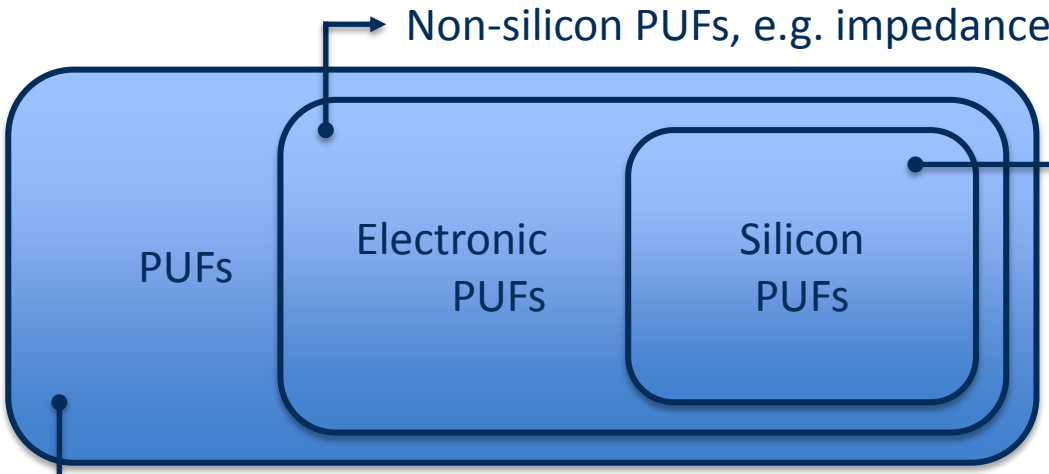
PUFs: Physically Unclonable Functions

- On many levels, PUFs are more like fingerprints than like programmed keys:

Human Fingerprint	PUF	Programmed Key
Unique per person	**Unique** per device	*No guarantee* of uniqueness
Inherent from birth	**Inherent** from production	Programmed *after* production
Impossible to "clone" humans with the same fingerprints	**Infeasible to "clone"** devices with the same PUF	*Easy* to program many devices with the same key

Silicon PUFs: classification & advantages

- Many PUF(-like) proposals in myriad of materials, techniques, …



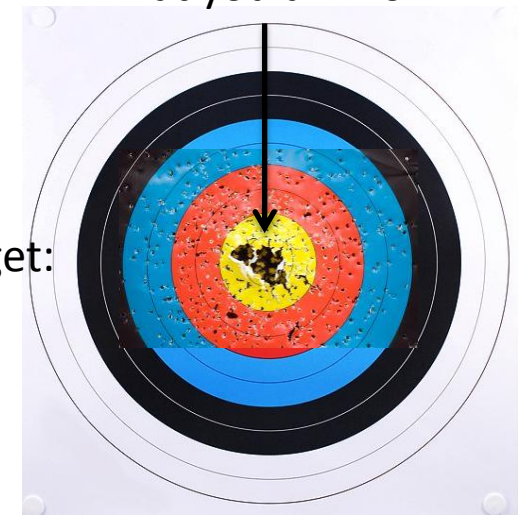
- Advantages of <u>silicon</u> PUFs:
  - Standard manufacturing with implicitly present randomness
  - Completely embedded in evaluating device
  - Easy integration with digital circuits → crypto implementations

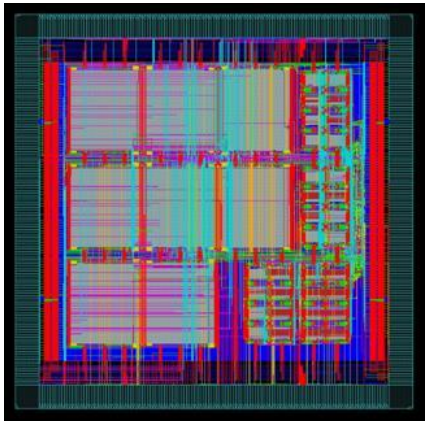
} *"Intrinsic PUFs"*

Silicon PUFs: process variations

What you aim for…

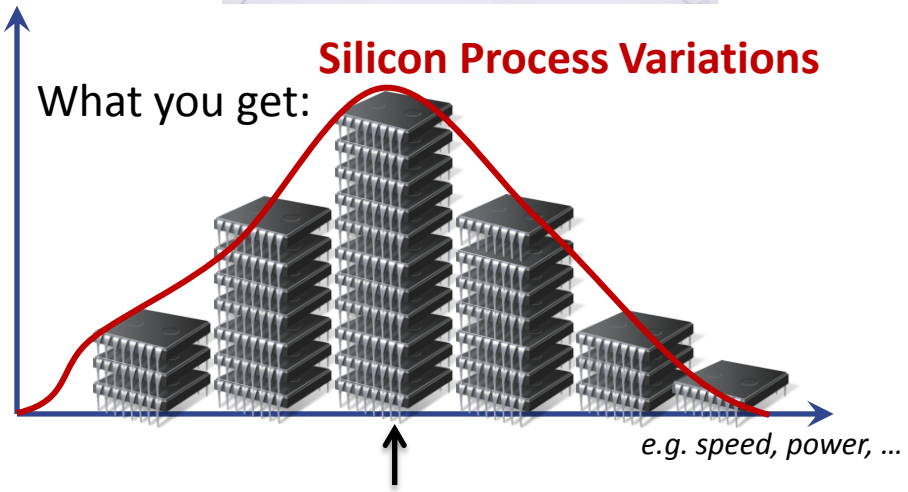
What you get:





Silicon Process Variations

What you get:



What you design for…

Silicon PUF Constructions: general idea

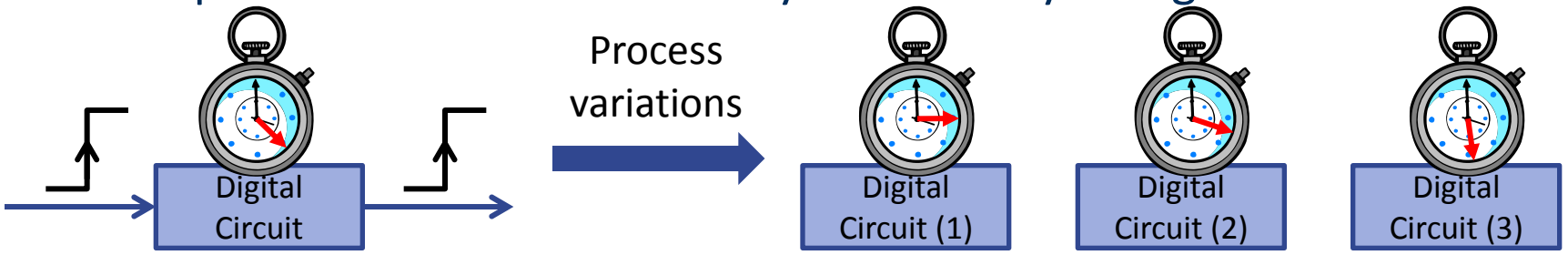
- **Silicon PUF construction =**
  a silicon circuit whose response (*y*) is mainly determined by process variations (*PV*) and the applied challenge (*x*)

- Ideal silicon PUF:
  $y = \boldsymbol{f}(PV, x)$

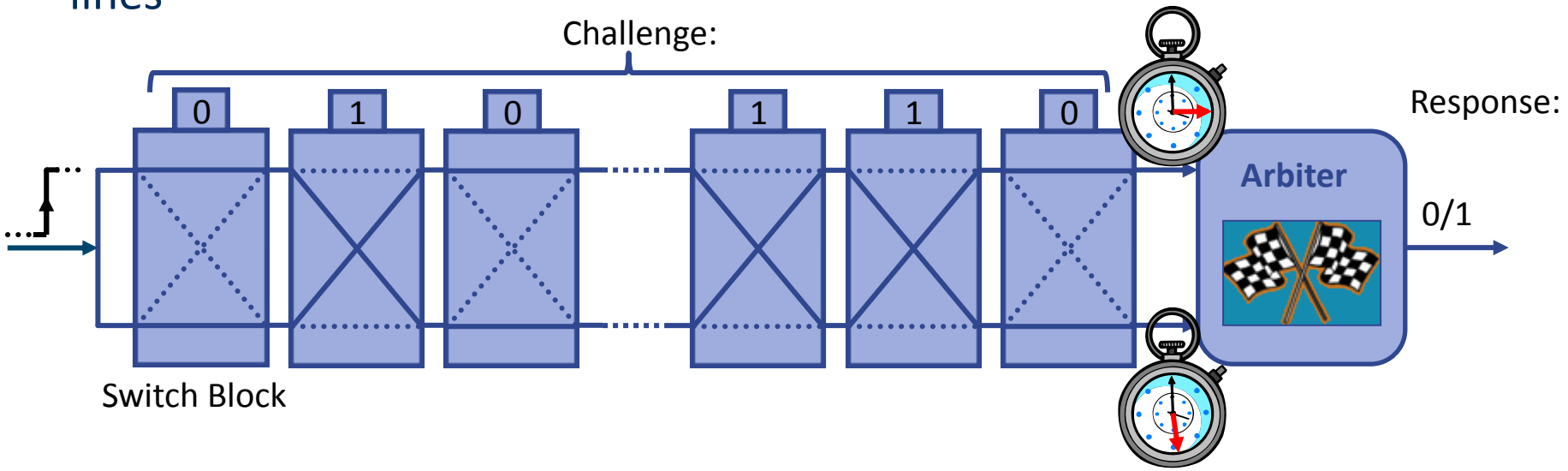
- Silicon PUFs in practice:
  $y = \boldsymbol{f}(PV, x; \ldots$ → PUF behavior
  Temp, V_{dd}, Noise, Device age, … → Unreliable
  Deterministic offset, Structural bias …) → Biased

Delay-based silicon PUFs

- Silicon process variations randomly affect delay of digital circuits

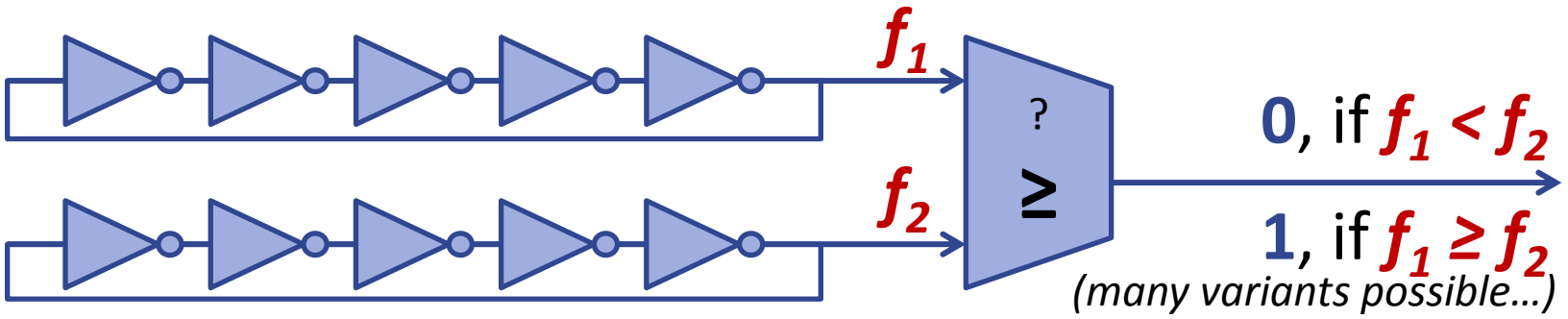


- **Arbiter PUF** exploits *race conditions* between identically designed delay lines

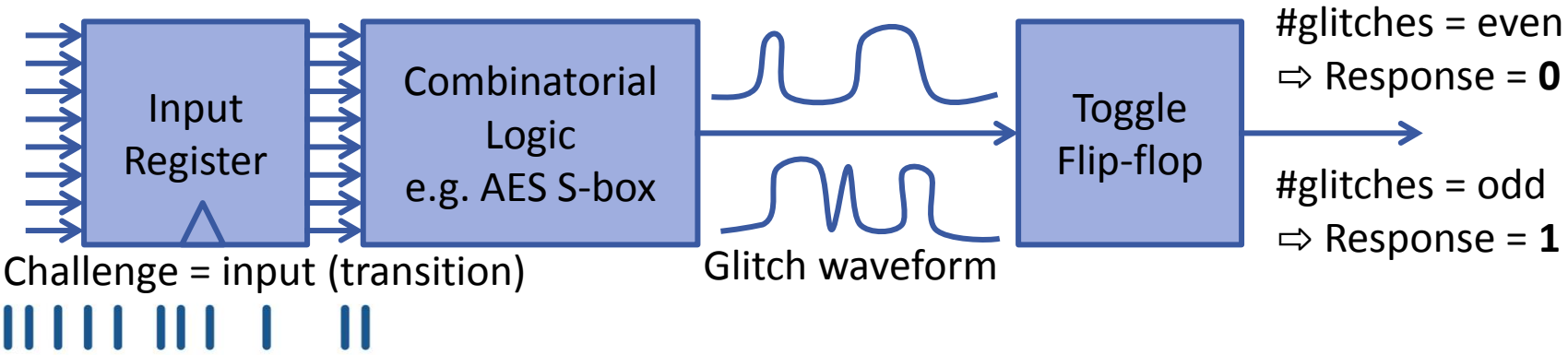


Delay-based silicon PUFs

- **Ring Oscillator PUFs** exploit *frequency variability* amongst identically designed ring oscillator circuits

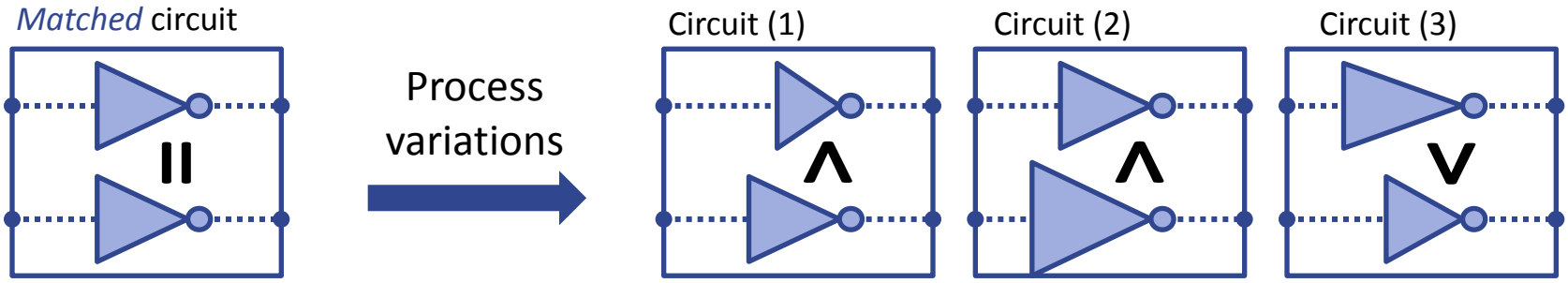


- **Glitch PUF** exploits variability in *glitch behavior* of identically designed combinatorial circuits

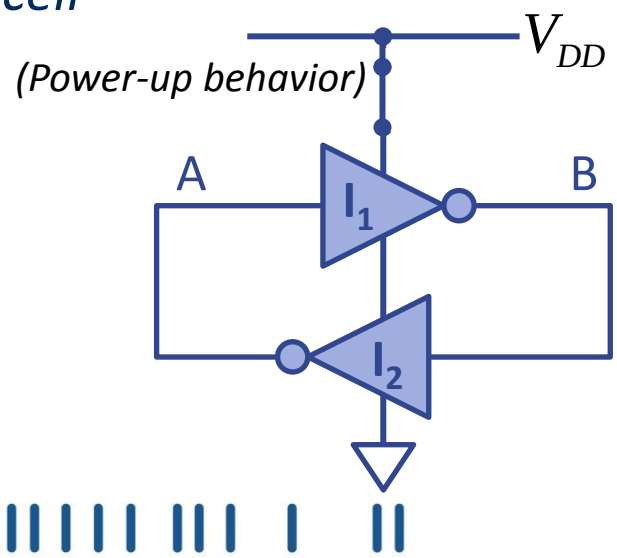


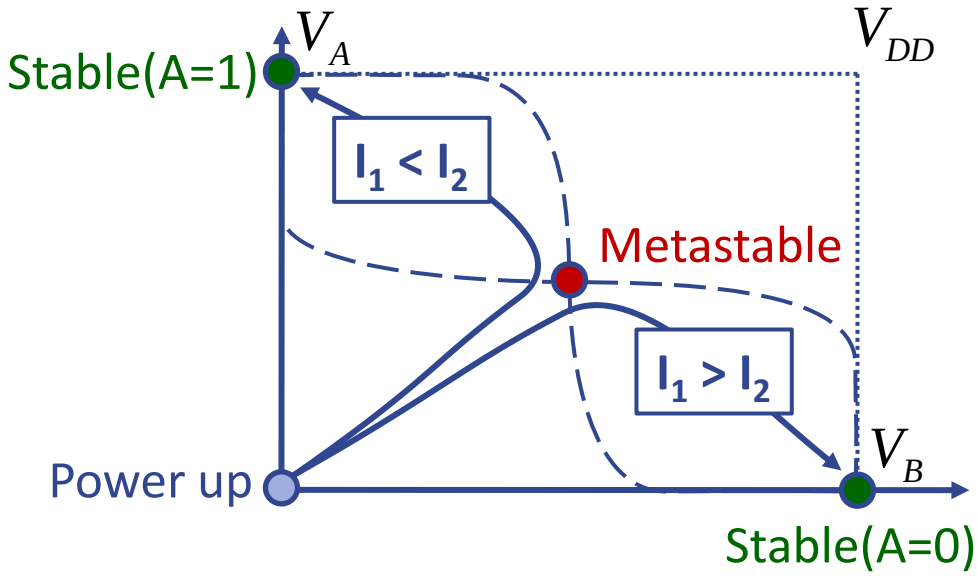
Bi-stable memory based PUFs: SRAM PUF

- Silicon process variations cause device "mismatch"



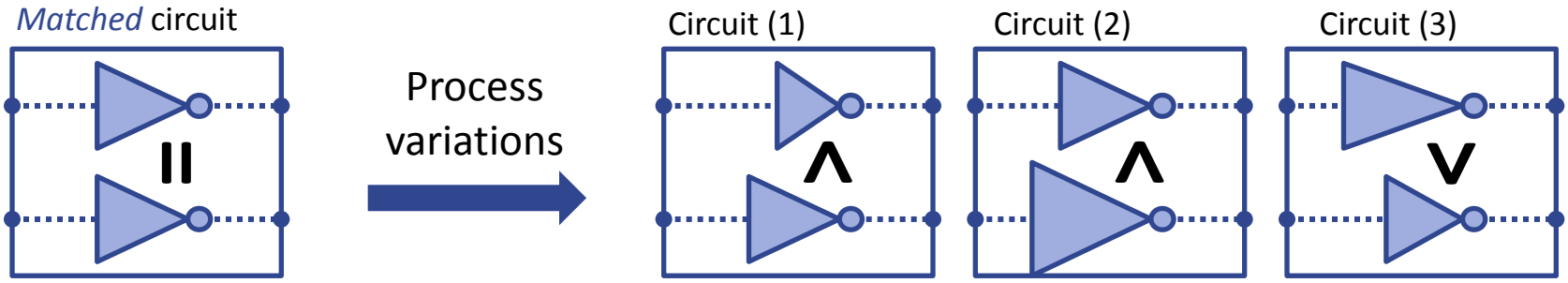
- **SRAM PUF** based on mismatch between "matched" invertors in *SRAM cell*





Bi-stable memory based PUFs: SRAM PUF

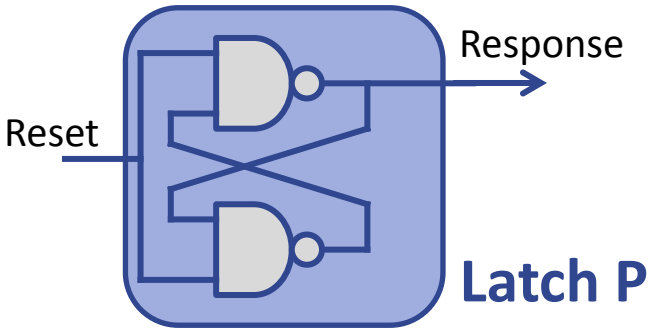
- Silicon process variations cause device “mismatch”



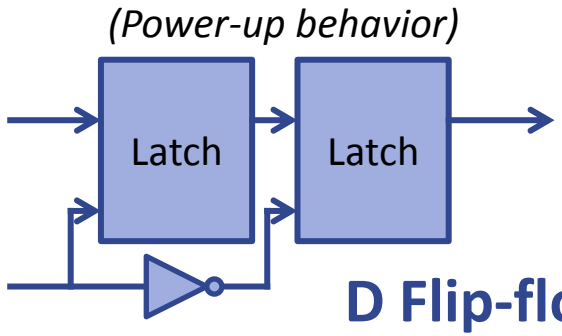
- **SRAM PUF** based on mismatch between “matched” invertors in *SRAM cell*

Bi-stable memory based PUFs: other elements

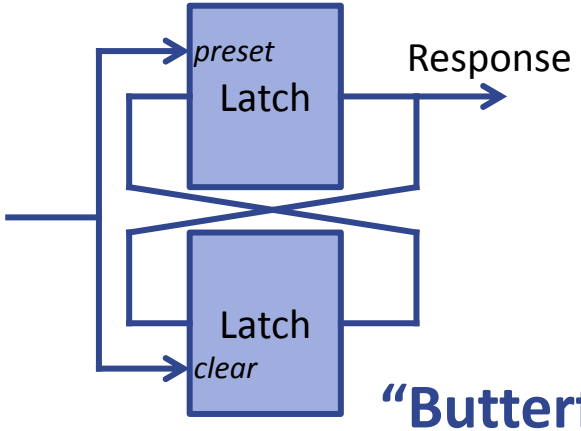
- Similar PUF behavior in other memory cells



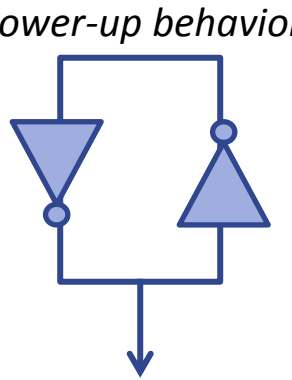
Latch PUF



D Flip-flop PUF



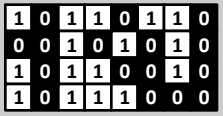
“Butterfly” PUF

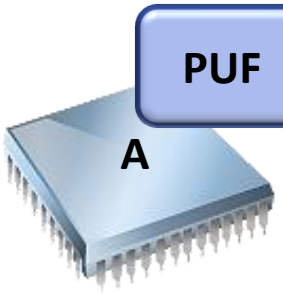


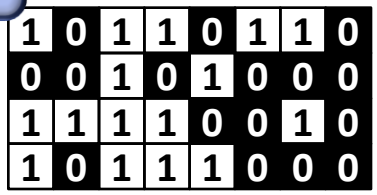
Buskeeper PUF

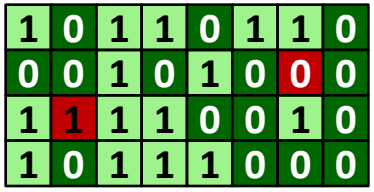
Basic PUF properties: reproducibility

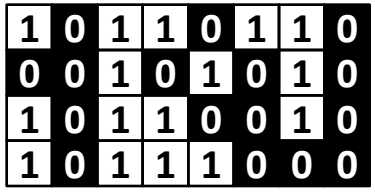
Name	Fingerprint
Alice	

Chip	PUF response
A	



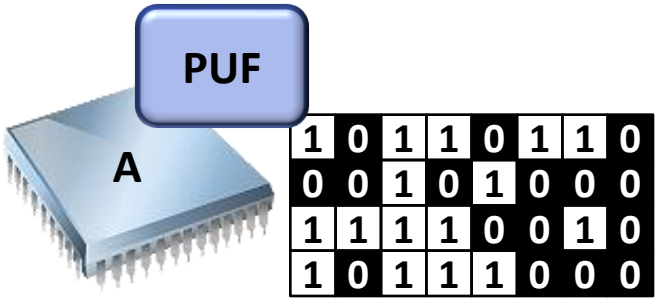


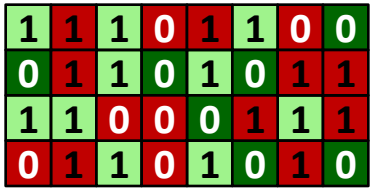


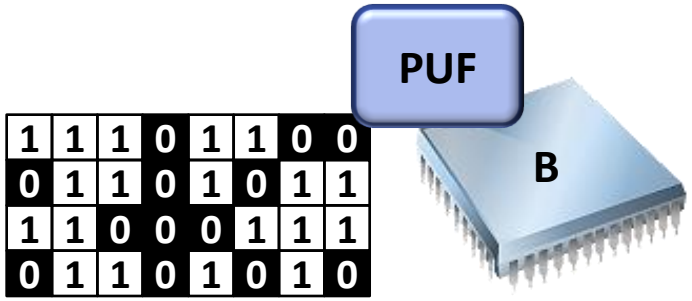


Intra-distance = 2 bit = 6.25%

Basic PUF properties: uniqueness



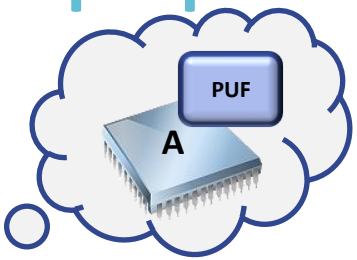




Inter-distance = 15 bit = 46.88%

Basic PUF properties: unpredictability

Insight
+
Guessing



Eve

Chip	PUF response
A	10110110 00101010 10110010 10111000

1	1	0	1	1	1	1	0
0	0	1	0	1	0	0	0
0	1	0	1	1	0	0	1
1	0	1	0	1	0	1	0

1	1	0	1	1	1	1	0
0	0	1	0	1	0	0	0
0	1	0	1	1	0	0	1
1	0	1	0	1	0	1	0

1	0	1	1	0	1	1	0
0	0	1	0	1	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	1	0	0	0

Accurate Prediction = 20 bits = 62.5%

Database

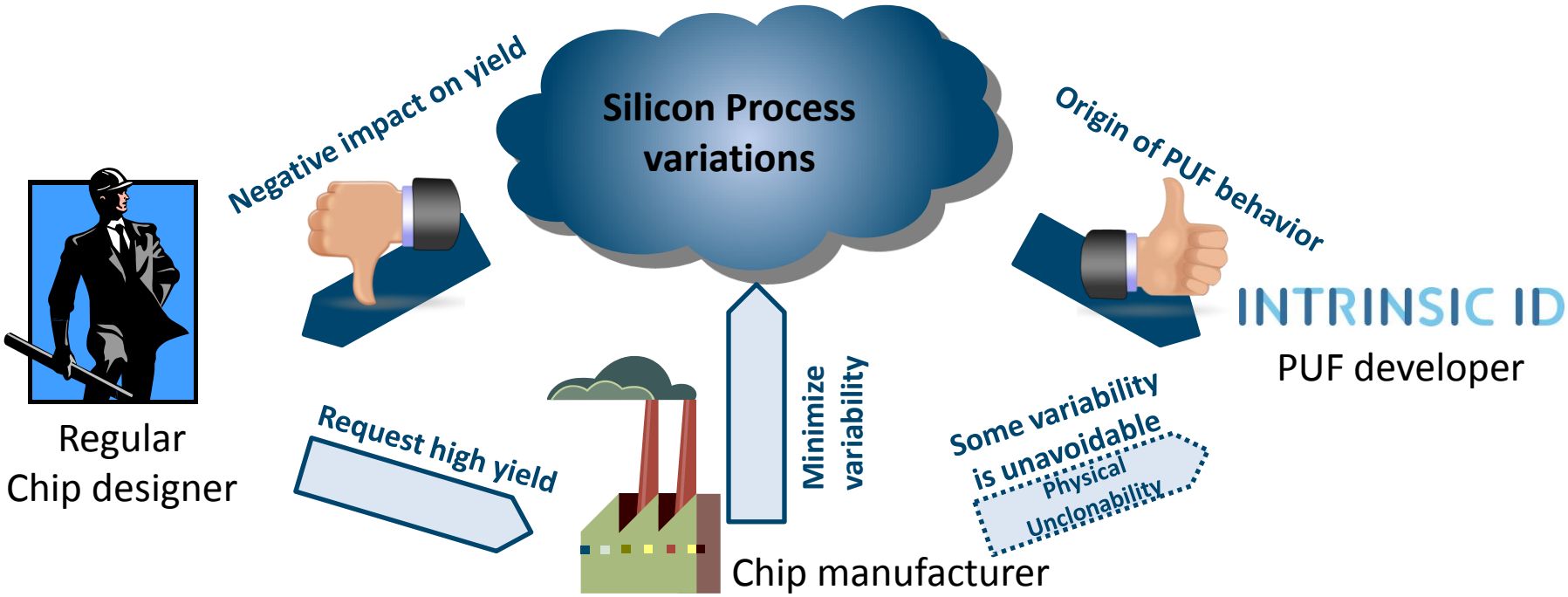
- Complete (100%) unpredictability = guessing every bit
  → 50% prediction accuracy
- Use **entropy** to express unpredictability:
  - 50% accuracy → 100% entropy → 100% *"guessing"* and 0% *"insight"*
  - 62.5% accuracy → 95.4% entropy → 95.4% *"guessing"* and 4.6% *"insight"*

$$H(X) = -\sum_{x} p(x) \log_2 p(x)$$

Unpredictability → 95.4% entropy

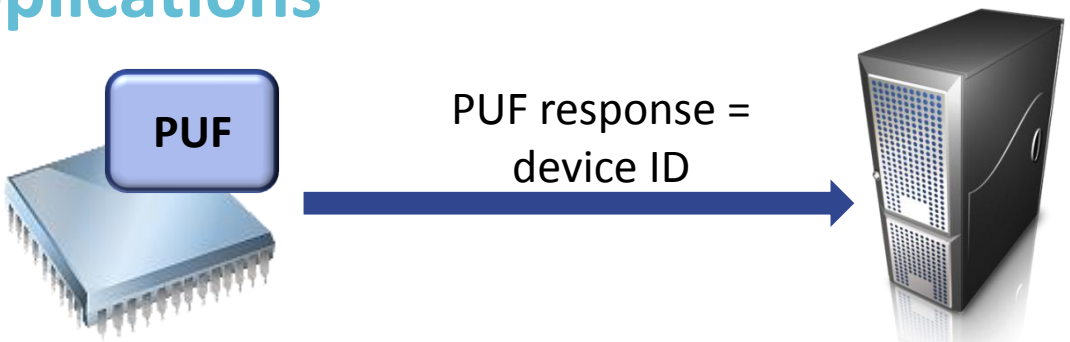
Basic PUF properties: "physical unclonability"

- Technical infeasibility/impossibility to create "non-unique" PUF instantiations
  - Due to **uncontrollable** random process variations

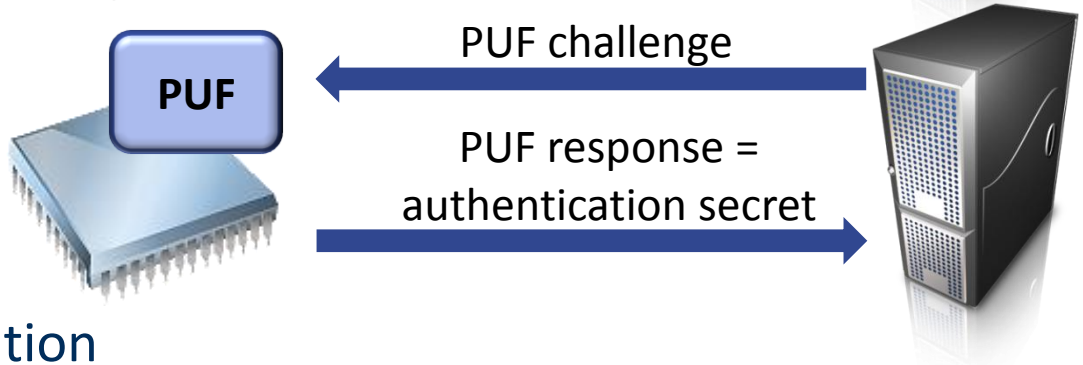


Silicon PUF-based applications

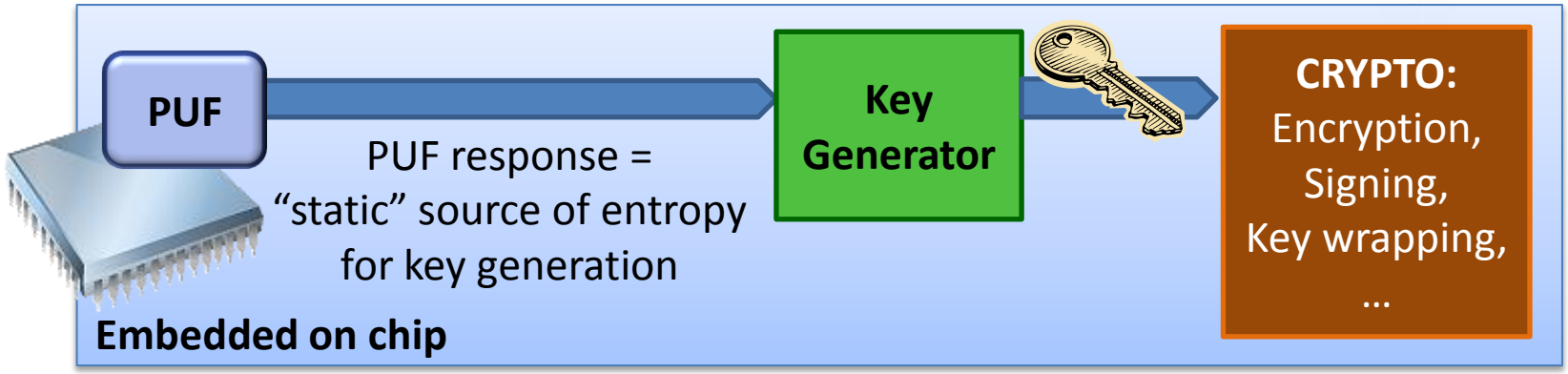
- Device identification



- Device authentication
  - Some variant of:

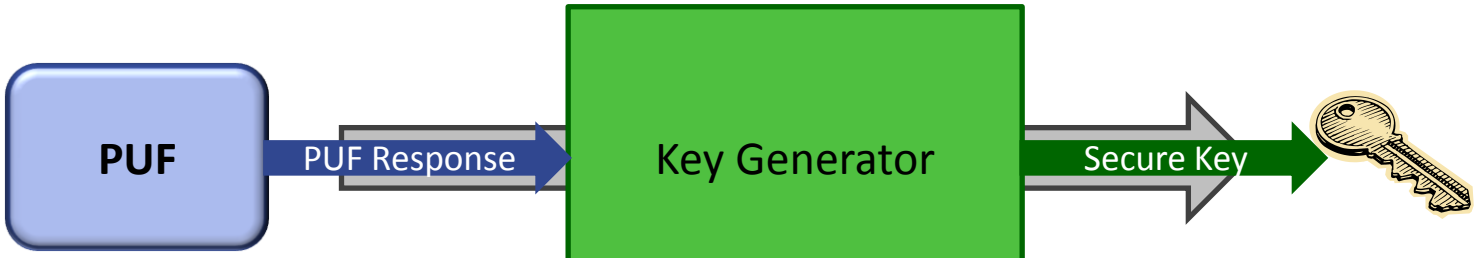


- Cryptographic key generation



Key generation/storage with Silicon PUFs

- Discrepancy between PUF response and crypto key:



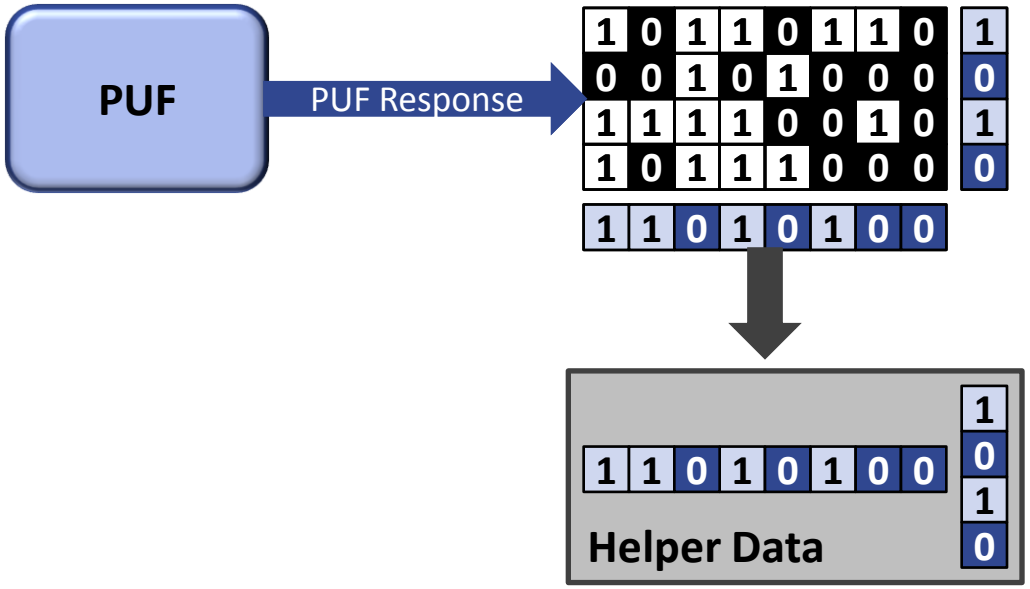
- Reproducible:
  e.g. 3% intra-distance
- Unpredictable:
  e.g. 70% entropy

- Reproducible:
  0% failure rate
- Unpredictable:
  100% entropy

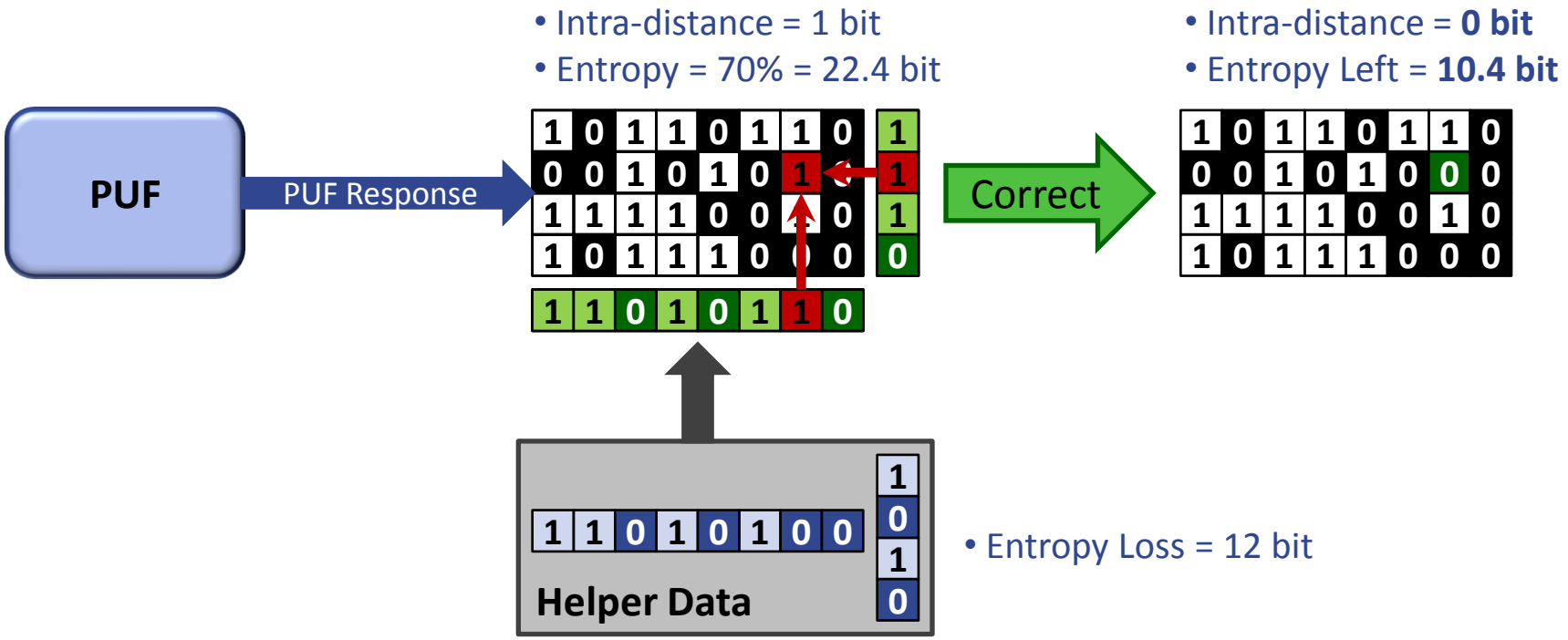
- Key Generator:
  1. Improves *reproducibility* by taking care of intra-distance of response = **correct bit errors**
  2. Improves *unpredictability* by extracting unpredictable part of response = **compress & accumulate entropy**

PUF-based key generation: Error correction

- Intra-distance = 1 bit
- Entropy = 70% = 22.4 bit

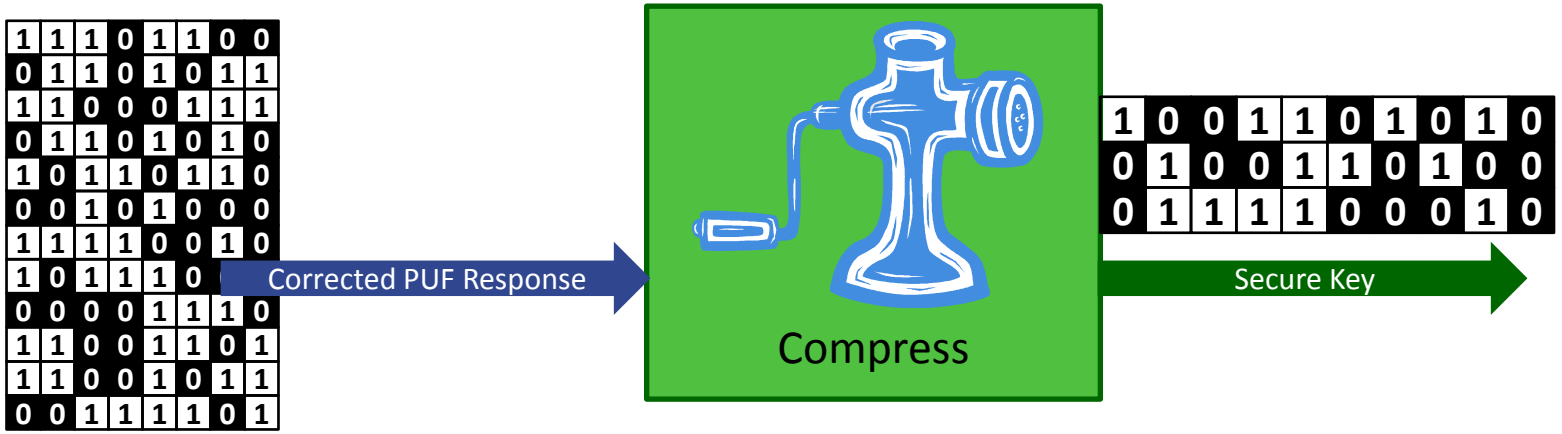


PUF-based key generation: Error correction



- **Result:** *reproducibility improves* drastically,
  but *unpredictability decreases* due to helper data leakage

PUF-based key generation: Entropy extraction



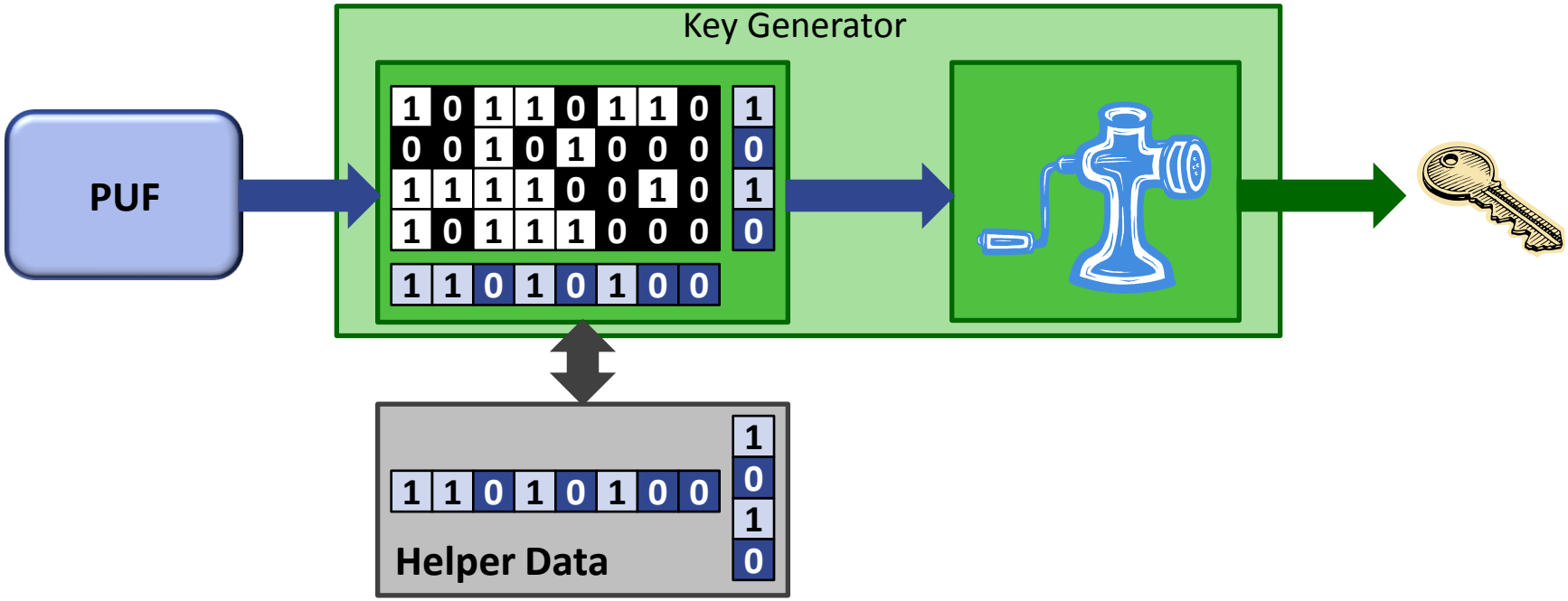
PUF Response Length: **96 bit**
Accumulated Entropy: **31.2 bit**

Key Length: **30 bit**

- **Result:** *Sufficient unpredictability* achieved by accumulating and compressing response bits
- Extracted key length ≤ total accumulated entropy

PUF-based key generation: Fuzzy Extractor

- Combination of error correction and entropy extraction:



Practical PUF-based key generators

- To give you some idea of realistic systems (from literature):
  - All for 128-bit keys:

	PUF type	PUF size	PUF error rate	Error Correction	Key failure rate
Boesch et al., [CHES-2008]	SRAM PUF	3696 bits	15%	Repetition + Golay (hard decision)	10^{-6}
Maes et al., [CHES-2009]	SRAM PUF	1536 bits	15%	Repetition + Reed-Muller (soft decision, multi enroll)	10^{-6}
van der Leest et al., [CHES-2012]	SRAM PUF	2880 bits	15%	Repetition + Golay (soft decision, single enroll)	10^{-6}
Maes et al., [CHES-2012]	Ring Oscillator PUF	848 oscillators	13%	Repetition + BCH (hard decision)	10^{-9}

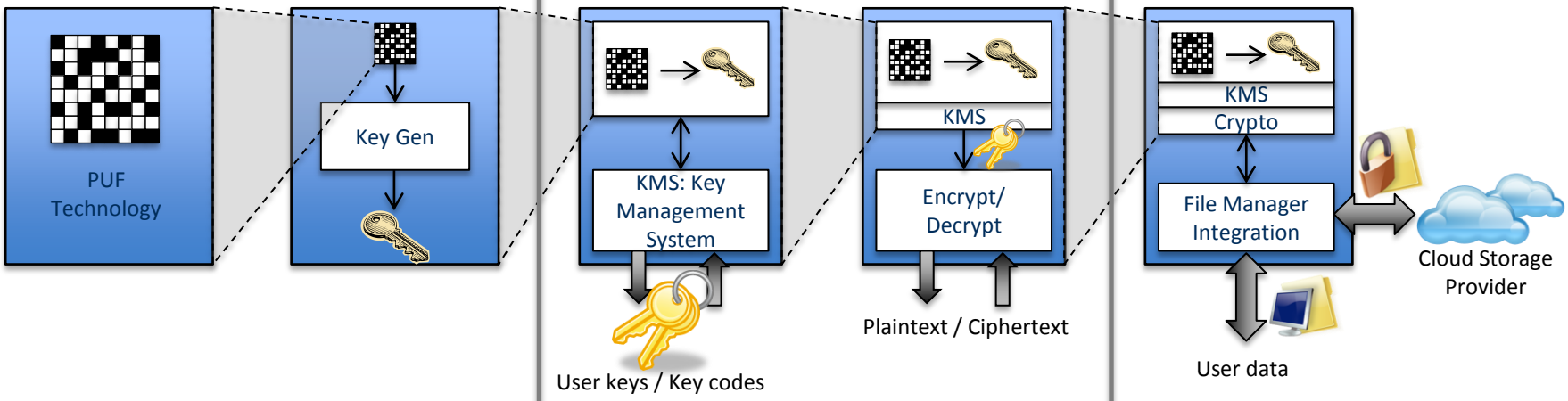
- PUF error rate significantly affects error correction and PUF size
  - Key failure rate has less impact

Towards PUF-based user applications

R&D into PUF constructions and PUF-based key generators

Integration of PUF-based key generators with crypto/security (e.g. key management, encryption, …)

Development of end-user applications using PUF-enabled crypto/security (e.g. secure cloud storage, …)



PUFs: Recent Developments

- Physical Attacks on PUFs
  - PUFs, like all physical crypto primitives, can be susceptible to physical attacks

- E.g.
  - EM analysis on ring oscillator PUFs [Merli et al., TRUST 2011]
  - Remanence decay attack on SRAM PUFs [Oren et al., CHES 2013]
  - Photon Emission Analysis (PEA) on SRAM PUFs [Helfmeier et al., HOST 2013]
  - Invasive attacks [Nedospasov et al., FDTC 2013]

- Countermeasures are possible

Recent Developments: Aging and Anti-Aging (for SRAM PUFs)

- SRAM PUF "natural aging"
  - Power-up behavior: fastest transistor (of matched pair) closes first
  - NBTI aging: closed transistors become slower over time
  - Result: power-up behavior changes over time
    hence: # bit errors increases over time

- SRAM PUF "anti-"aging
  - Long-term storage of the power-up state **inverse** reinforces the power-up behavior
    Result: # bit errors **decreases** over time! [Maes et al., HOST 2014]
  - A similar effect (HCI) can also be applied in an accelerated manner immediately after production to improve the reliability of an SRAM PUF from the start
    [Bhargava et al, CHES 2013]

Summary

- A **silicon PUF** is a *process variation dependent circuit*
  → effectively a **"device fingerprint"**
  - **Delay-based** constructions: arbiter PUF, ring oscillator PUF, …
  - **Memory-based** constructions: SRAM PUF, D flip-flop PUF, …
  - **"Physically unclonable"**: process variations are beyond manufacturer's control

- PUFs are typically *noisy* and *biased*, crypto keys are not…
  → PUF-based key generator: **PUF → KeyGen → Crypto Key**
  - Improve robustness with **error-correction** techniques → **helper data**
  - Improve unpredictability with **entropy accumulation**

INTRINSIC ID

Secure your digital life™