IEEE SecDev | 2021

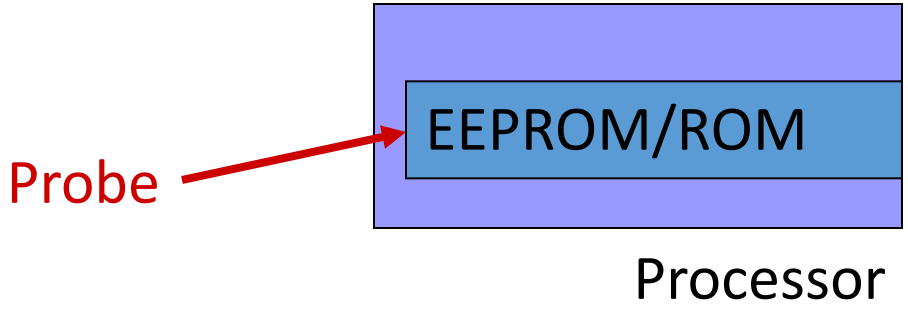
IEEE COMPUTER SOCIETY

Physical Unclonable Functions

Srini Devadas
MIT

Problem

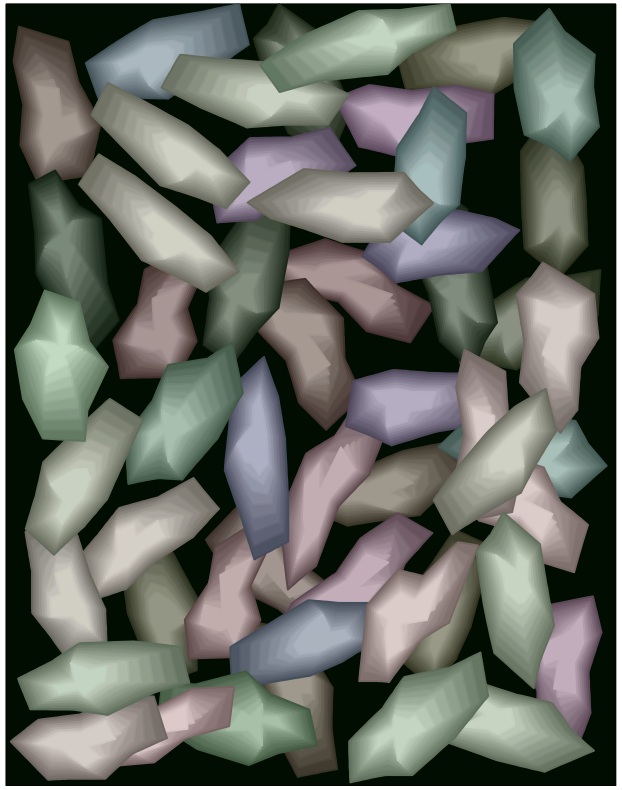
Storing digital information in a device such that it is resistant to physical attack is difficult and expensive



- Adversaries can physically extract secret keys from EEPROM while processor is off
- Trusted party must embed and test secret keys in a secure location

A Solution

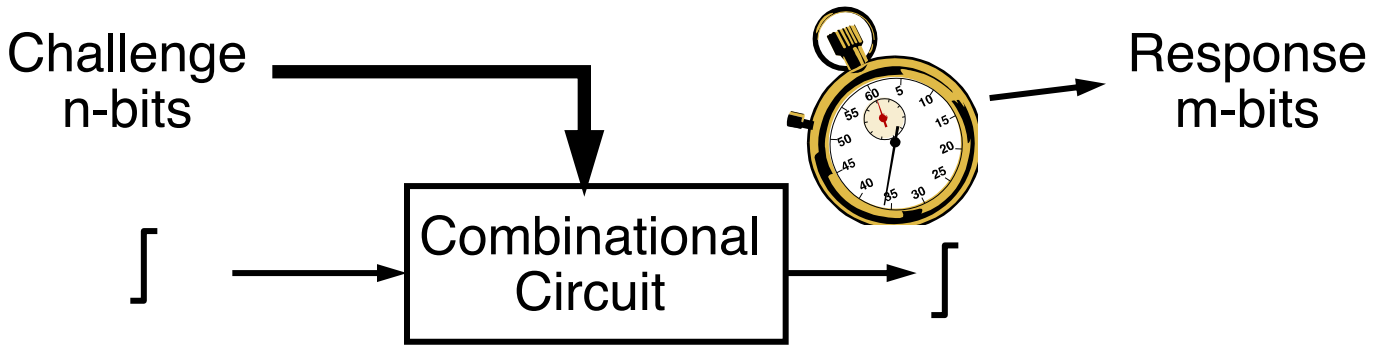
- Extract key information from a complex physical system.



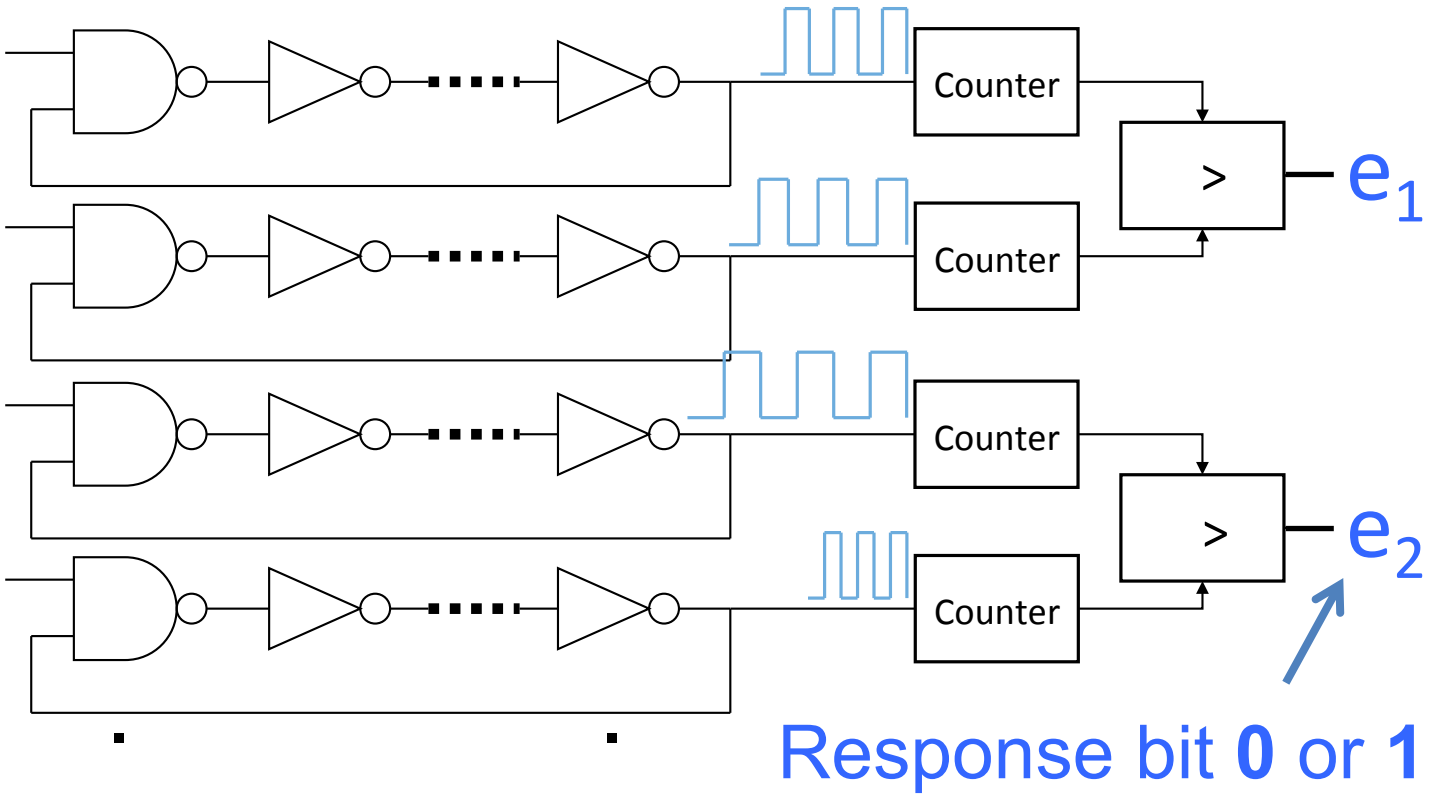
Physical Unclonable Functions

- PUFs are a silicon biometric
- Because of random process variations, no two Integrated Circuits even with the same layouts are identical

- Delay-Based Silicon PUF concept (2002)
  - Generate keys from unique delay features of chips



Ring Oscillator PUF

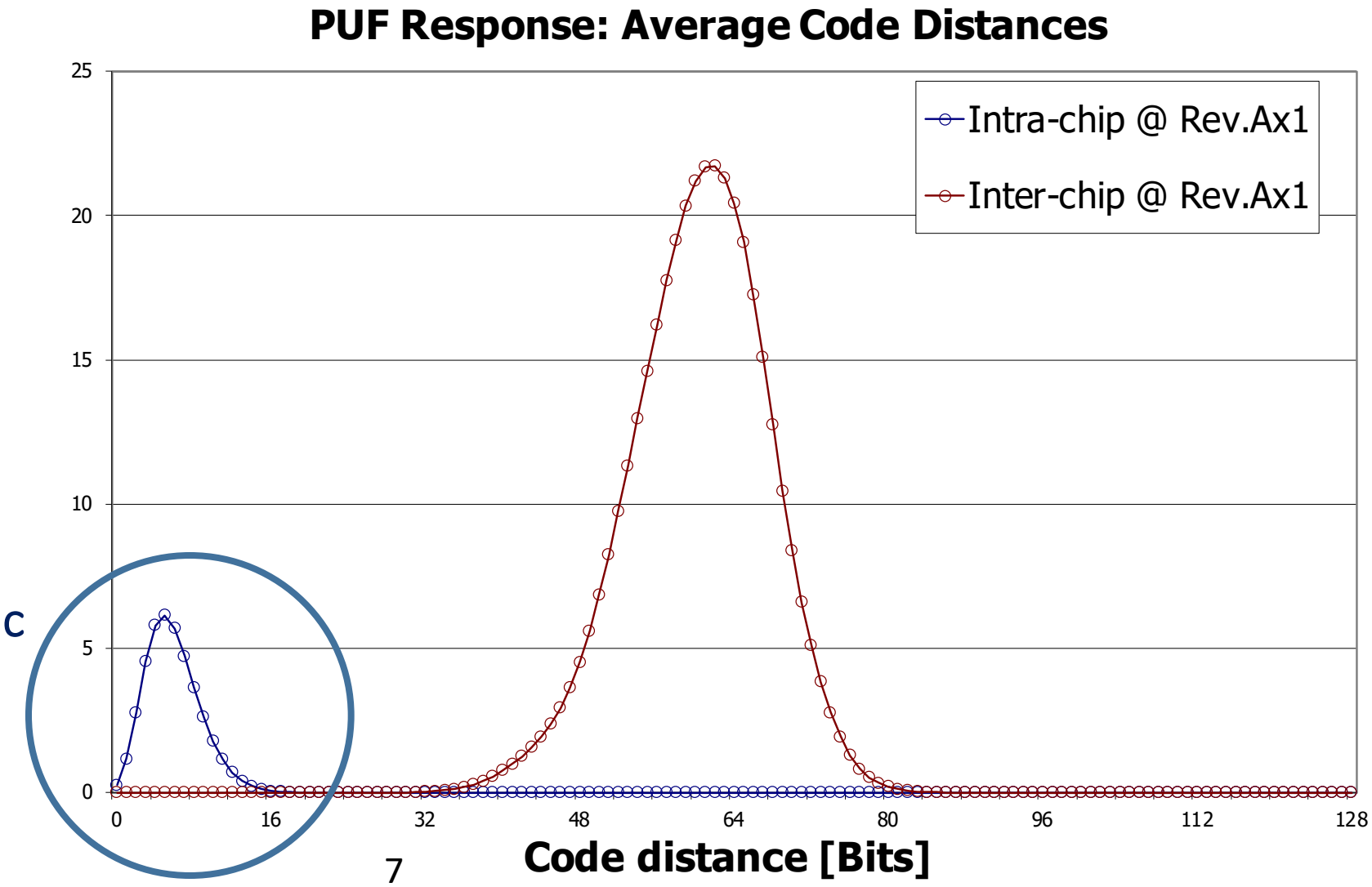


Metrics

- Security: Show that different PUFs (ICs) generate different bits
  - *Inter-chip* variation: how many PUF bits (in %) are different between PUF A and PUF B?
  - Ideally, inter-chip variation should be close to 50%

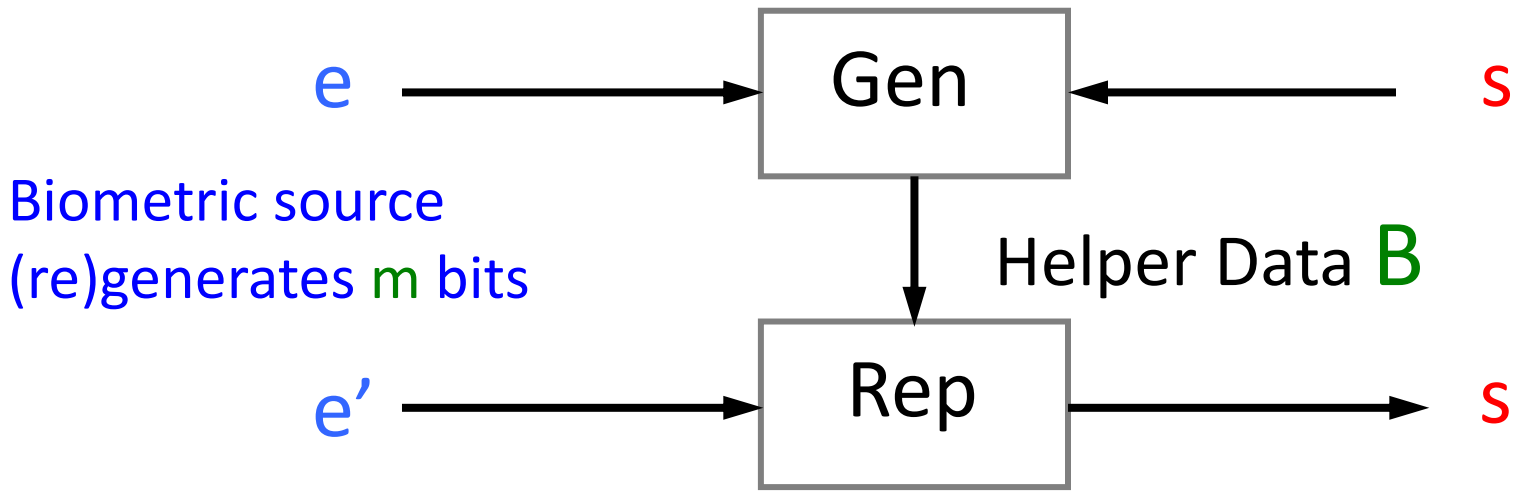
- Reliability: Show that a given PUF (IC) can re-generate the same bits consistently
  - *Intra-chip* variation: how many bits flip when re-generated again from a single PUF
  - Environments (voltage, temperature, etc.) can change
  - Ideally, intra-chip variation should be 0%

128 RO Pairs: Many Chips and Environments



Cryptographic keys need zero error rate

Fuzzy Extractors Produce Stable Keys



- Public Helper Data B leaks information about e, s
- Information-theoretically speaking, number of remaining secret bits = |s| - |B|
- Problem: error rate ↑ ⇒ |B| ↑ ⇒ secret bits ↓

Learning Parity with Noise (LPN)

$$b_1 = a_1 \,.\, s + e_1$$

$$b_2 = a_2 \,.\, s + e_2$$

$$\ldots$$

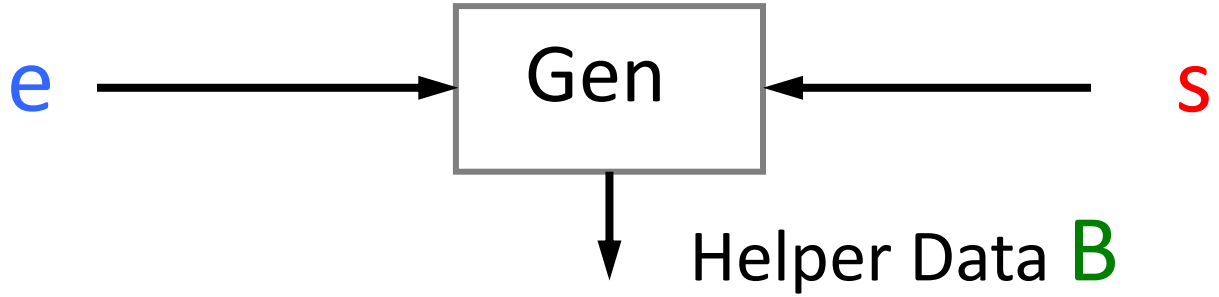
$$b_m = a_m \,.\, s + e_m$$

s secret, a_i and b_i public, e_i is hidden noise

a_i, s are n-bit vectors, b_i, e_i are bits

Hard to discover s given a_i and b_i
for any $m > n$ for any non-zero noise level

Gen Step



Helper Data B

$$b_1 = a_1 \,.\, s + e_1$$

$$b_2 = a_2 \,.\, s + e_2$$

$$\ldots$$

$$b_m = a_m \,.\, s + e_m$$

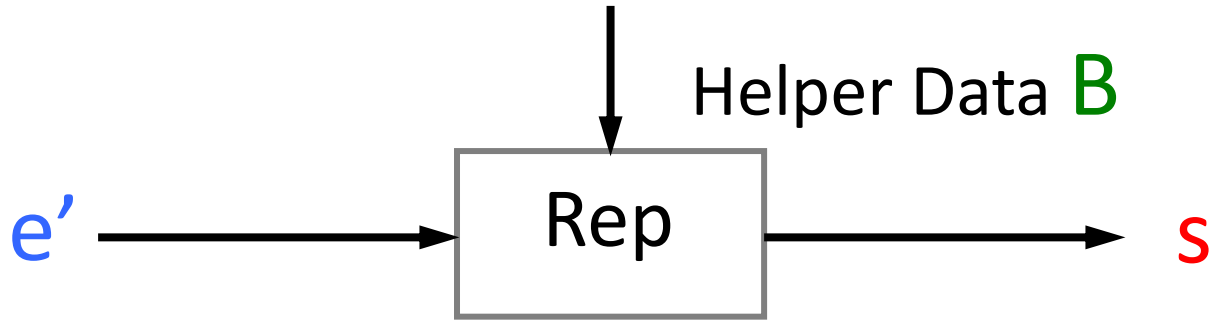
Computed and is public helper data

PUF generates these values

Choose randomly. Same and public for all instances

Choose randomly and is secret

Rep Step



$$b_1 = a_1 \cdot s + e'_1$$

$$b_2 = a_2 \cdot s + e'_2$$

$$\ldots$$

$$b_m = a_m \cdot s + e'_m$$

known

known

PUF regenerates these values

unknown

Rep Step

$$b_1 = a_1 \,.\, s + e'_1$$

$$b_2 = a_2 \,.\, s + e'_2$$

$$\ldots$$

$$b_m = a_m \,.\, s + e'_m$$

known

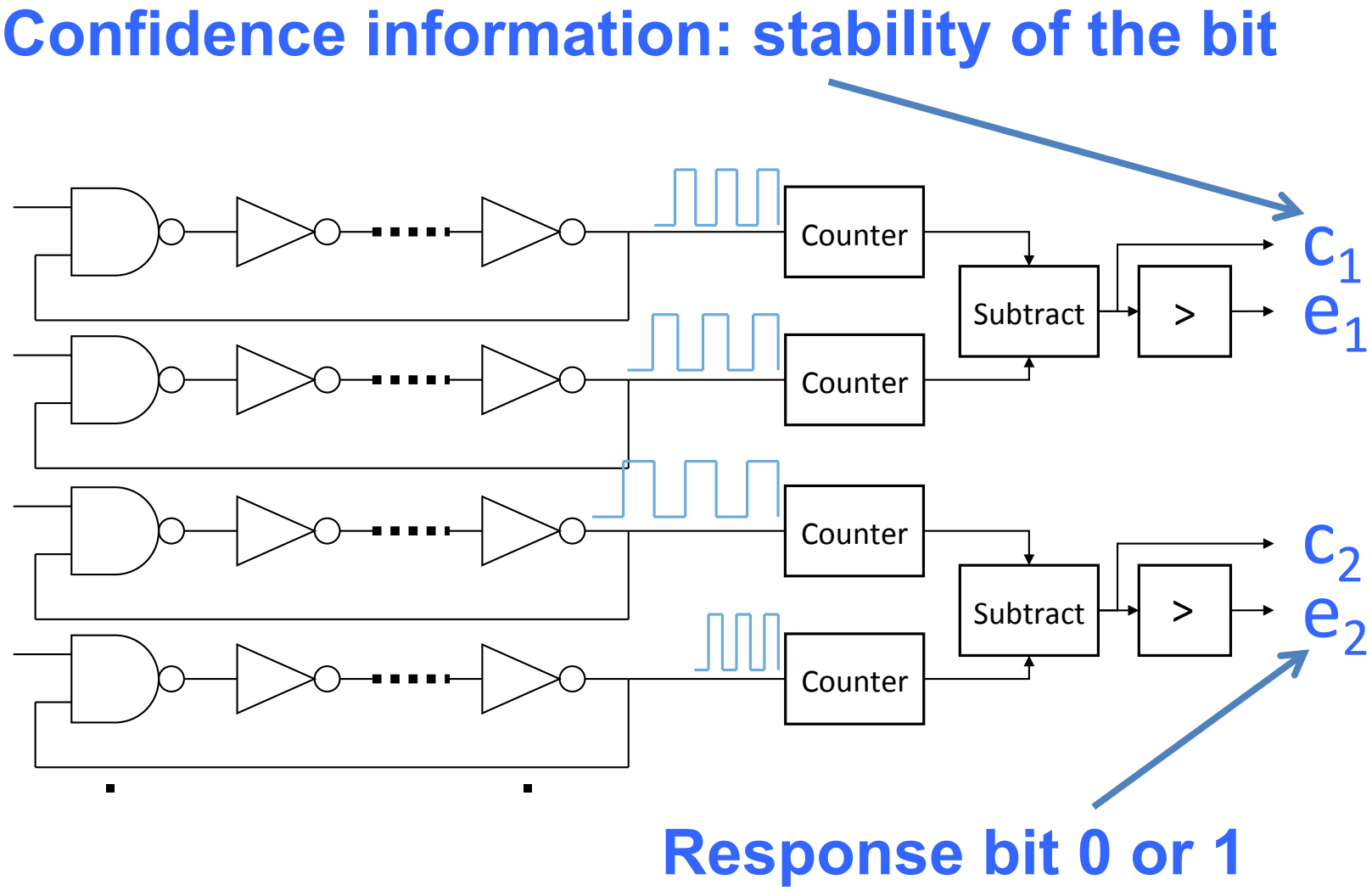
known

PUF regenerates these values

unknown

- Problem: e'_i values not the same as e_i values
- LPN is hard even for small amount of noise

Ring Oscillator PUF



Confidence Information

$$b_1 = a_1 \cdot s + e'_1$$

$$\ldots$$

$$b_m = a_m \cdot s + e'_m$$

Need only n out of m e'_i values to be correct to solve for s

Can use c_i confidence information associated with each e'_i value to find n correct values!

Rep Step: Key Extraction

$$b_1 = a_1 \cdot s + e'_1$$

$$b_5 = a_5 \cdot s + e'_5$$

$$\ldots$$

$$b_{m-1} = a_{m-1} \cdot s + e'_{m-1}$$

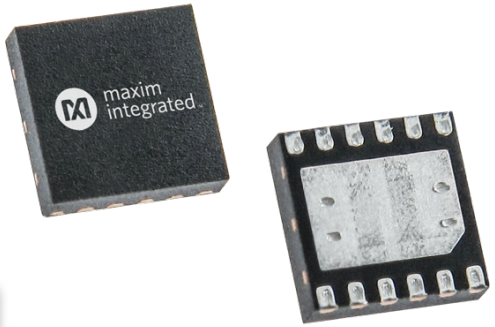
Set m to be Kn, and choose the most stable n bits out of the m using confidence information c_i

Obtain s through Gaussian elimination

K grows with error rate but doesn't affect security!

Adversary doesn't know $e_i/e'_i/c_i$ – faces LPN

Summary



- Various types of PUFs with a variety of error correction schemes generate secret keys in FPGA and crypto processor products