Physical Unclonable Functions

from Prof. Dan Holcomb

Hardware

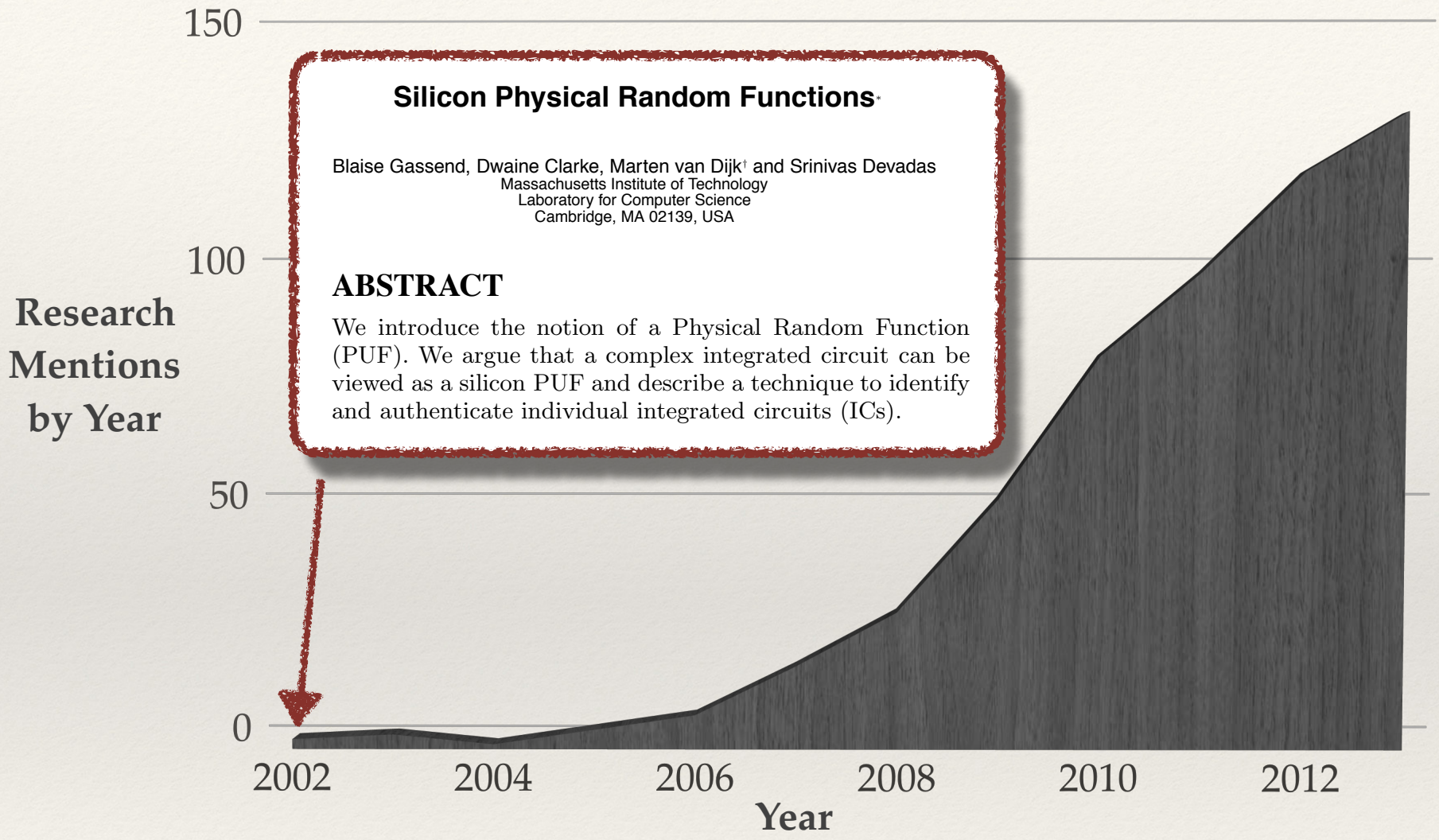
- Design, fabrication, test, distribution, deployment of chips are collaborative efforts involving a huge number of people around the world, with varying levels of trust in each other. This gives rise to a number of interesting security issues, including the following:

    1. Tamper during manufacture ?
    2. Counterfeit parts in supply chain ?
    3. Unauthorized overproduction ?
    4. Invasive reverse engineering of designs ?
    5. Side channel attacks ?

Outline

1. **PUFs**
2. Arbiter PUFs (Strong PUF)
3. SRAM Power-up PUFs (Weak PUF)

Physical Unclonable Functions



Silicon Physical Random Functions*

Blaise Gassend, Dwaine Clarke, Marten van Dijk† and Srinivas Devadas
Massachusetts Institute of Technology
Laboratory for Computer Science
Cambridge, MA 02139, USA

ABSTRACT

We introduce the notion of a Physical Random Function (PUF). We argue that a complex integrated circuit can be viewed as a silicon PUF and describe a technique to identify and authenticate individual integrated circuits (ICs).

Many Applications of PUFs, but today focusing on **anti-counterfeiting**

Counterfeit Electronics

https://www.sparkfun.com/news/364

sparkfun START SOMETHING

SHOP LEARN BLOG SUPPORT

PRODUCT MENU find products, tutorials, etc...

HOME / BLOG / 2010.05.17

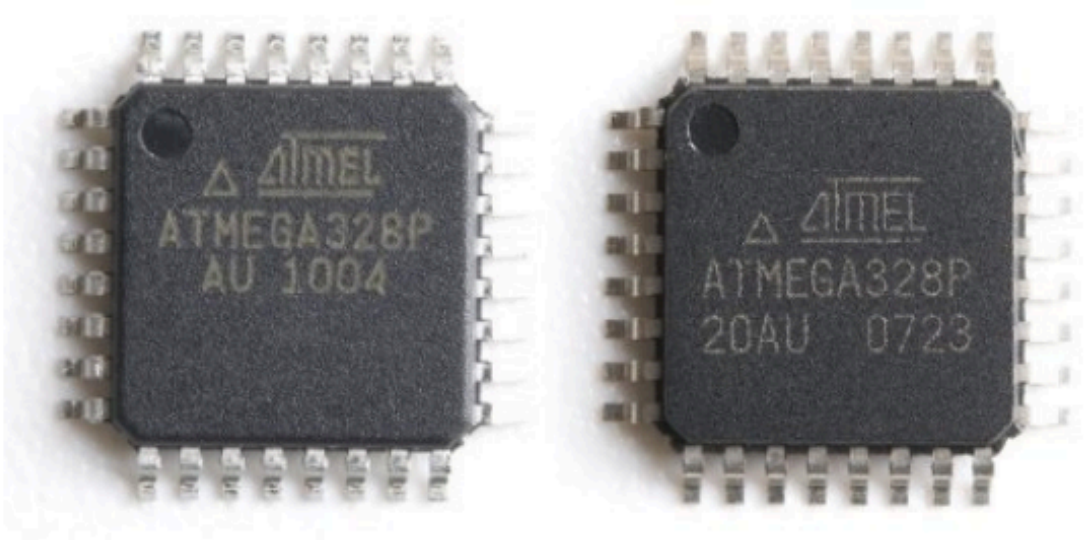
Revisiting the Counterfeit ATMega328s

A SparkFun customer and a a local dentist investigate the ATMega328 "slugs" - with interesting findings.

BY EMCEE GRADY MAY 17, 2010 02:00 EDT 41 BUSINESS TEARDOWN

FAVORITE 0

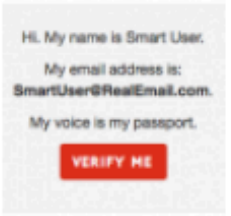
We made a post about the ATmega328 Slugs a little while back. The short version of the story is that we got some ATmegas that we couldn't get working. Are they counterfeit or are we inept at getting a microcontroller running? Hmmm.



A functioning ATMega on the left with a fake one on the right.

Ignoring the obvious questions of why the supplier bothered sending us anything or went through the hassle of creating parts that looked like the real things in some ways (packaged and stamped) but not in others (incorrect numbers and,

Related



Authenticating ICs to Avoid Counterfeits

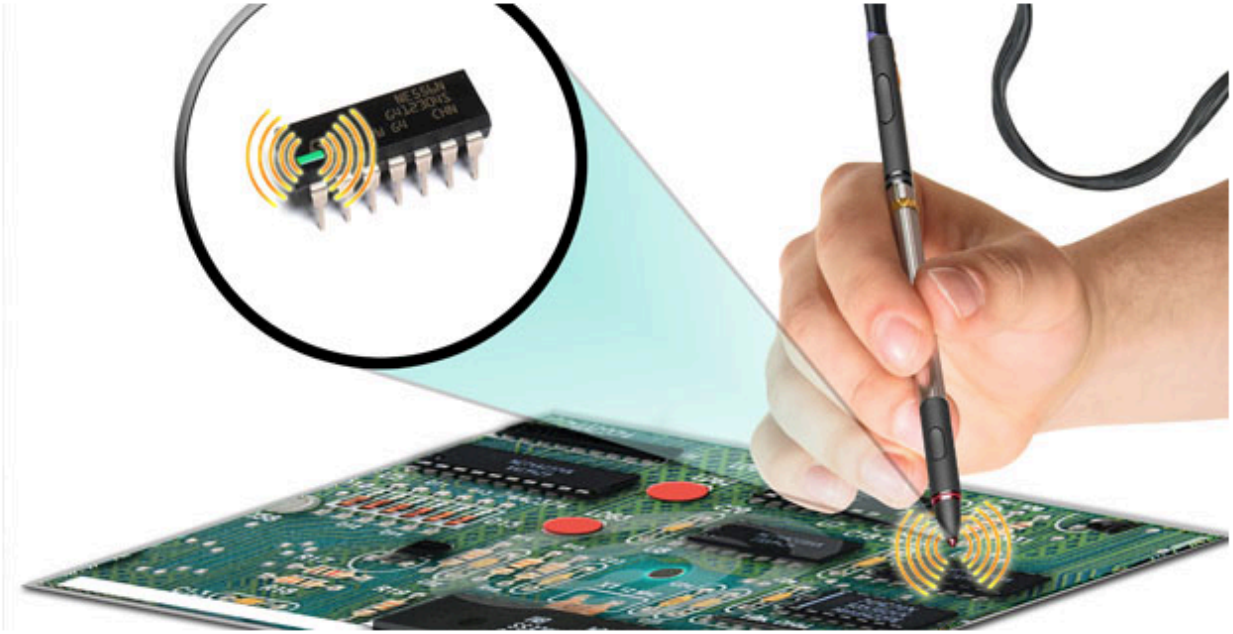
- Ensure authenticity by eliminate all places where counterfeits can enter supply chain
  - Trusted shipping channels
  - Tamper evident shipping containers
  - Trusted delivery and installers

- Alternative approach: authenticate chips using something that is hard to copy

Tiny, Cheap, Foolproof: Seeking New Component to Counter Counterfeit Electronics

New program seeks tool that authenticates electronic components at any step of the supply chain

OUTREACH@DARPA.MIL
2/24/2014



Used and non-authentic counterfeit electronic components are widespread throughout the defense supply chain; over the past two years alone, more than one million suspect parts have been associated with known supply chain compromises [1].

The problem is pervasive, with both expensive and inexpensive electronic parts being targeted. Counterfeit, or otherwise suspect electronic components, present a critical risk for the Department of Defense (DoD), where a malfunction of a single part could lead to system failures that can put

Products and materials are assigned a secure identity via molecular tagging and other visually-identifiable or machine-readable features. Tagging can be implemented anywhere during the lifecycle or supply chain (source, manufacturing, branding).

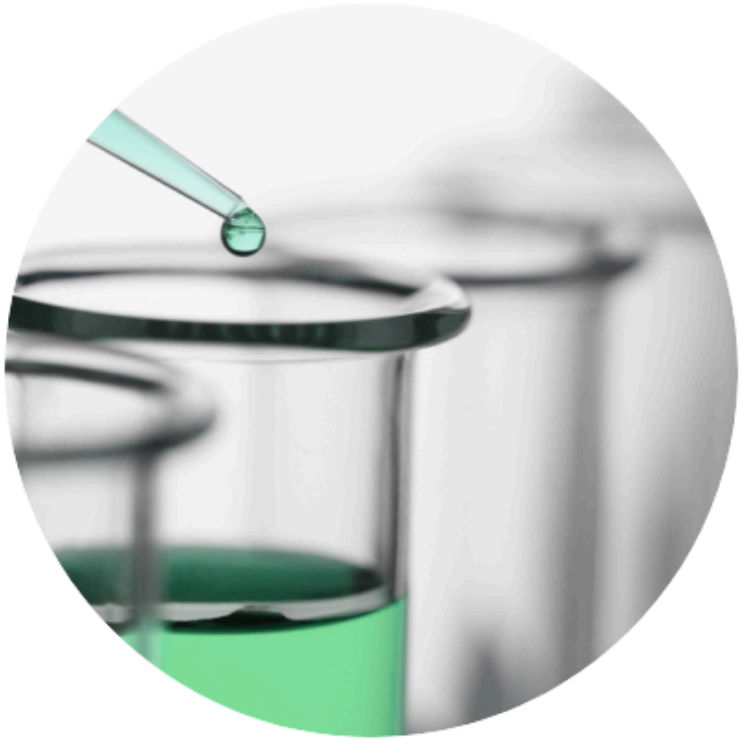
Raw materials, components, finished goods and packaging can be tagged using techniques that verify provenance, integrity and authenticity.

Custom molecular tags are produced to work seamlessly with your process and product. Our team will develop, validate and implement the ideal solution to fit your needs.

Verification of source, authenticity and provenance

The SigNature DNA platform has proven highly resistant to UV radiation, heat, cold, vibration, abrasion and other extreme environmental conditions, and a single SigNature DNA mark will support several authentications in its lifetime. Unsurpassed durability, accuracy and assurance make SigNature DNA markers an ideal foundation or enhancement for any security effort.

Manufacturers, brands, and other stakeholders can ensure their raw materials and products are protected, product claims are authentic and customer expectations are met.



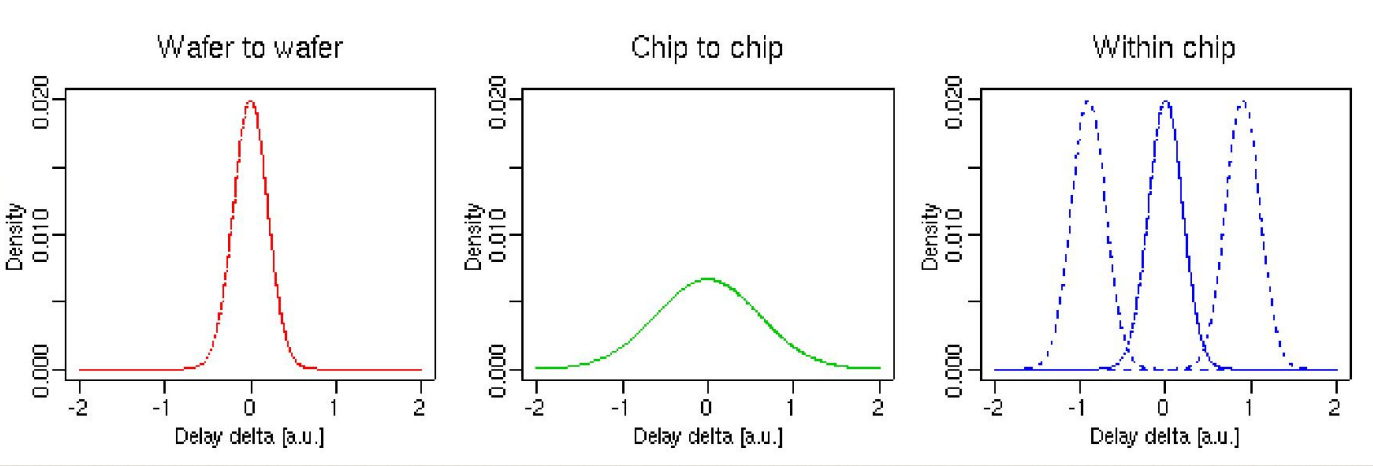
Unique Features of Things

Abstraction: the act of considering something as a general quality or characteristic, **apart from concrete realities**, specific objects, **or actual instances**. [dictionary.com]

Biometrics: the measurement and analysis of **unique** physical or behavioral **characteristics** especially as a means of **verifying personal identity**. [merriam-webster.com]

Unique Features of ICs

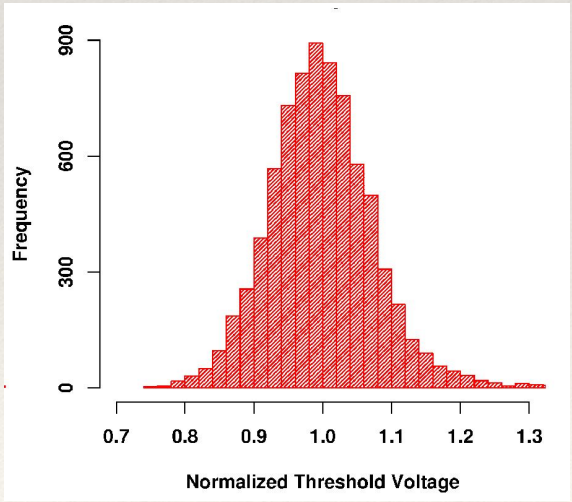
- All ICs have process variation from manufacturing
- Different process variations have different spatial correlation



- Random per wafer
- Random per chip
- Random per transistor
- Which is best for authenticating chips?

- Random dopant fluctuations
  - Concentration of dopant atoms in channel
  - Significantly determines threshold voltage
  - Persistent over life of chip
  - Spatially uncorrelated
  - Worse for small transistors

$$\sigma_{V_{th}} = \frac{A_{V_{th}}}{\sqrt{W * L}}$$



Physical Unclonable Functions (PUFs)

- PUF maps challenges to responses according to uncontrollable process variations

PUF Characterized by Challenge-Response Pairs (CRPs)

- PUFs are circuits with digital outputs that are sensitive to variation
  - PUFs also sensitive to noise
- PUF is like a physical version of a programmable secret key and a cryptographic hash function to map challenges to responses

PUFs vs Non-Volatile Keys

- Keys stored in Fuse or non-volatile memory can be discovered by a sufficiently motivated attacker
  - Battery-backed SRAM instead
- Attackers may probe metal wires on chips to steal secrets
  - Use metal mesh on upper level metals
  - Detect if attacker breaks metal and wipe out secrets
- PUFs are one way to address these problems:
  - Secrets are inherently volatile
    - Cannot be discovered from non-functional chip
  - Secrets are generated by a structure that is sensitive to disturbance
    - Invasive attack may alter PUF behavior
  - Low-cost CMOS process

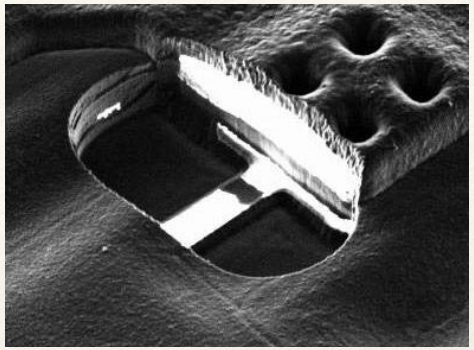


Figure 8: The interrupted white line at the bottom of the cavity in this FIB secondary-electron image is a blown polysilicon fuse next to a test pad (MC68HC05SC2x processor).

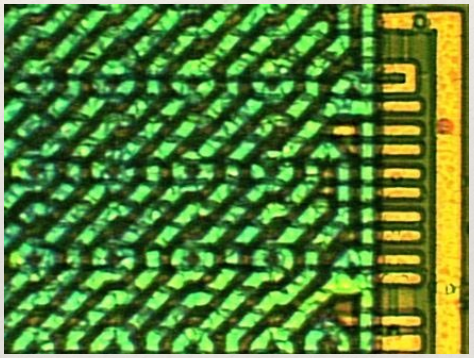


Figure 9: Escape route for imprisoned crypto bits: The ST16SF48A designers generously added this redundant extension of the data bus several micrometers beyond the protected mesh area, providing easy probing access.

Design Principles for
Tamper-Resistant Smartcard Processors

Oliver Kömmerling
Advanced Digital Security Research

Markus G. Kuhn
University of Cambridge

Weak vs Strong PUFs

Weak PUFs

Strong PUFs

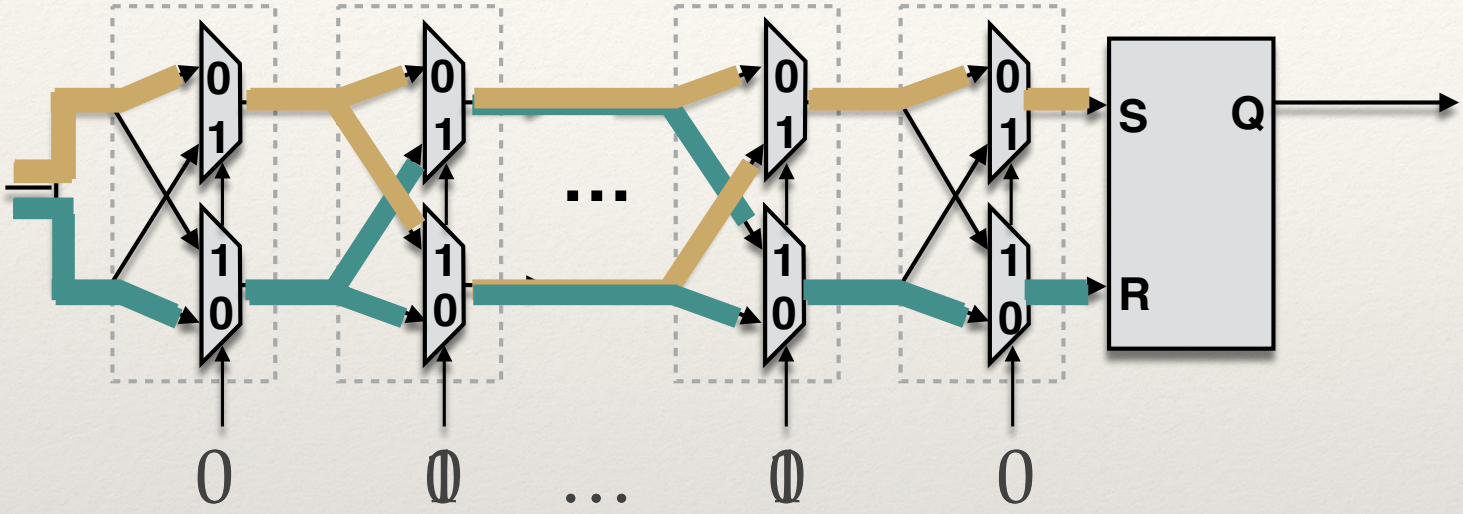
- Weak and strong are two PUF subclasses among many
  - Controlled PUFs
  - Public PUFs
  - SIMPL, etc

Outline

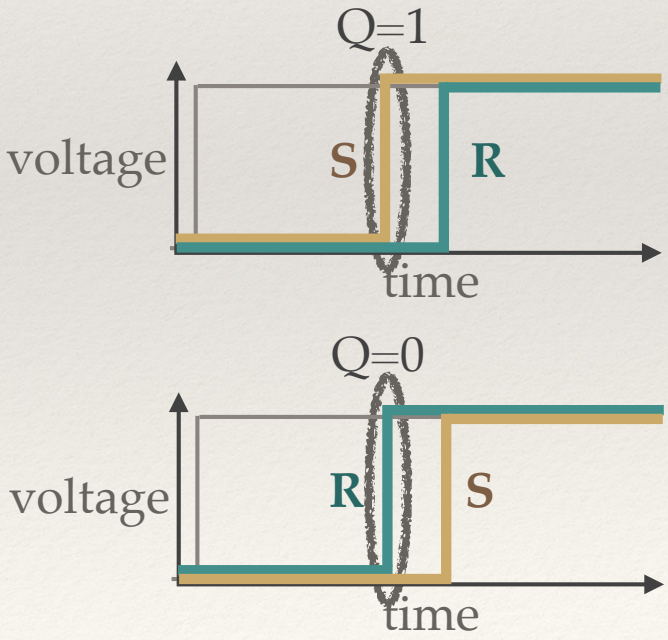
1. PUFs
2. **Arbiter PUFs (Strong PUF)**
3. SRAM Power-up PUFs (Weak PUF)

Arbiter PUF

[D. Lim et al., '05]

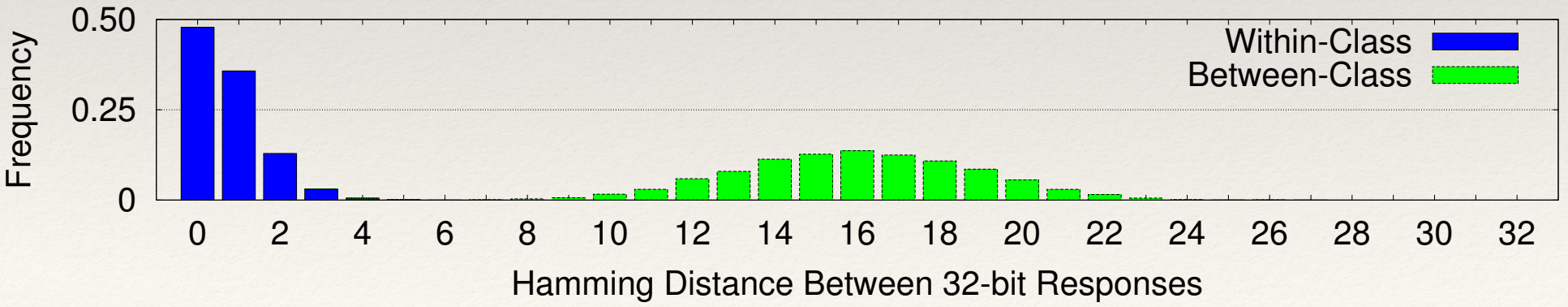


- Challenges: $c_i \in 2^m$ (m= num stages)
- Responses: $r_i \in \{0,1\}$ (for multi-bit output use many Arbiter PUFs in parallel)
- Response uniqueness comes from random delays of 4 paths through each stage



How to use responses?

- For given set of challenges, different PUF instances should produce responses that are very different across instances
  - Large "between-class" distance
- When same set of challenges are applied to same PUF at different times, responses should be the same or similar
  - Small "within-class" distance

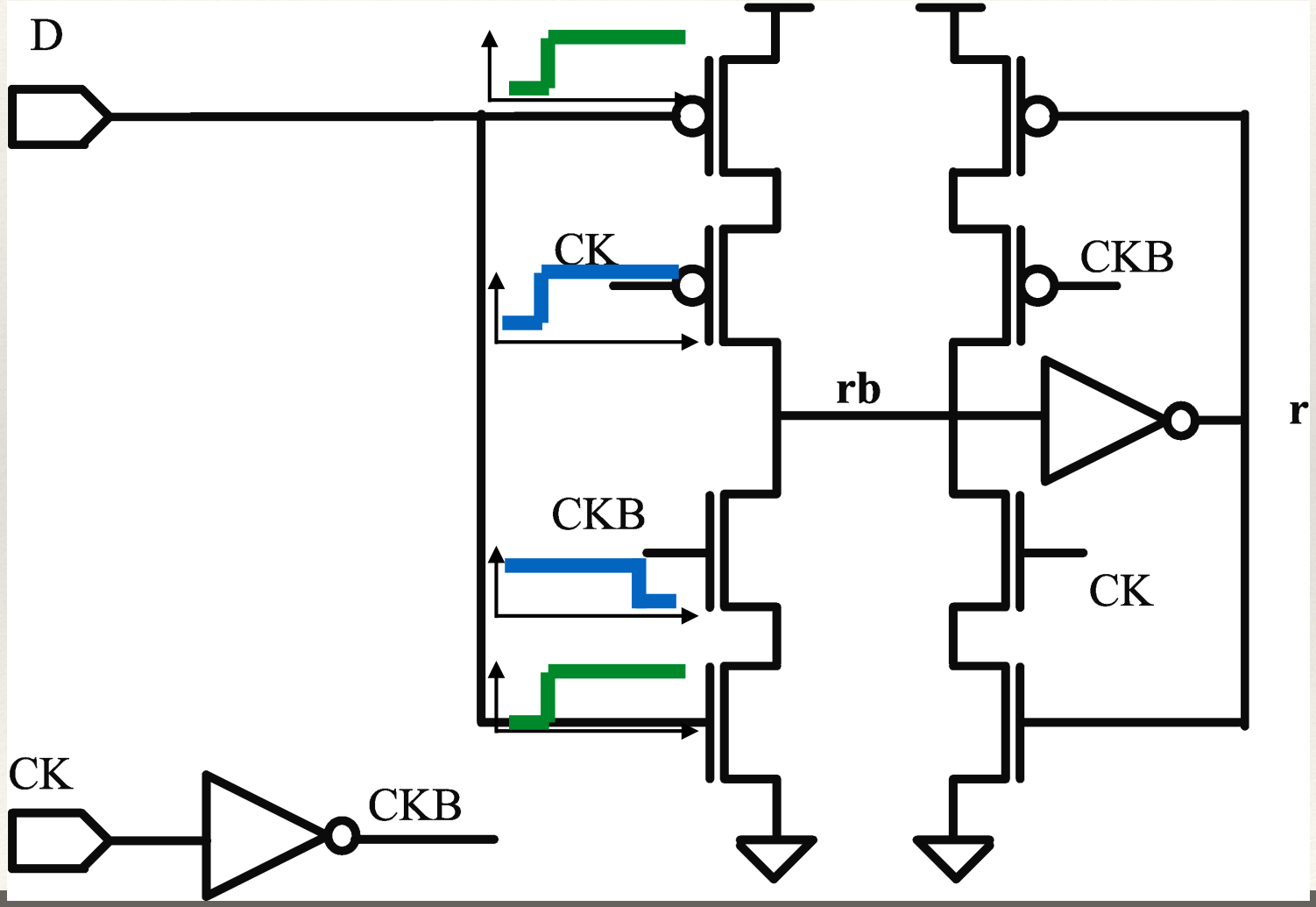


Arbiter PUF Design Considerations

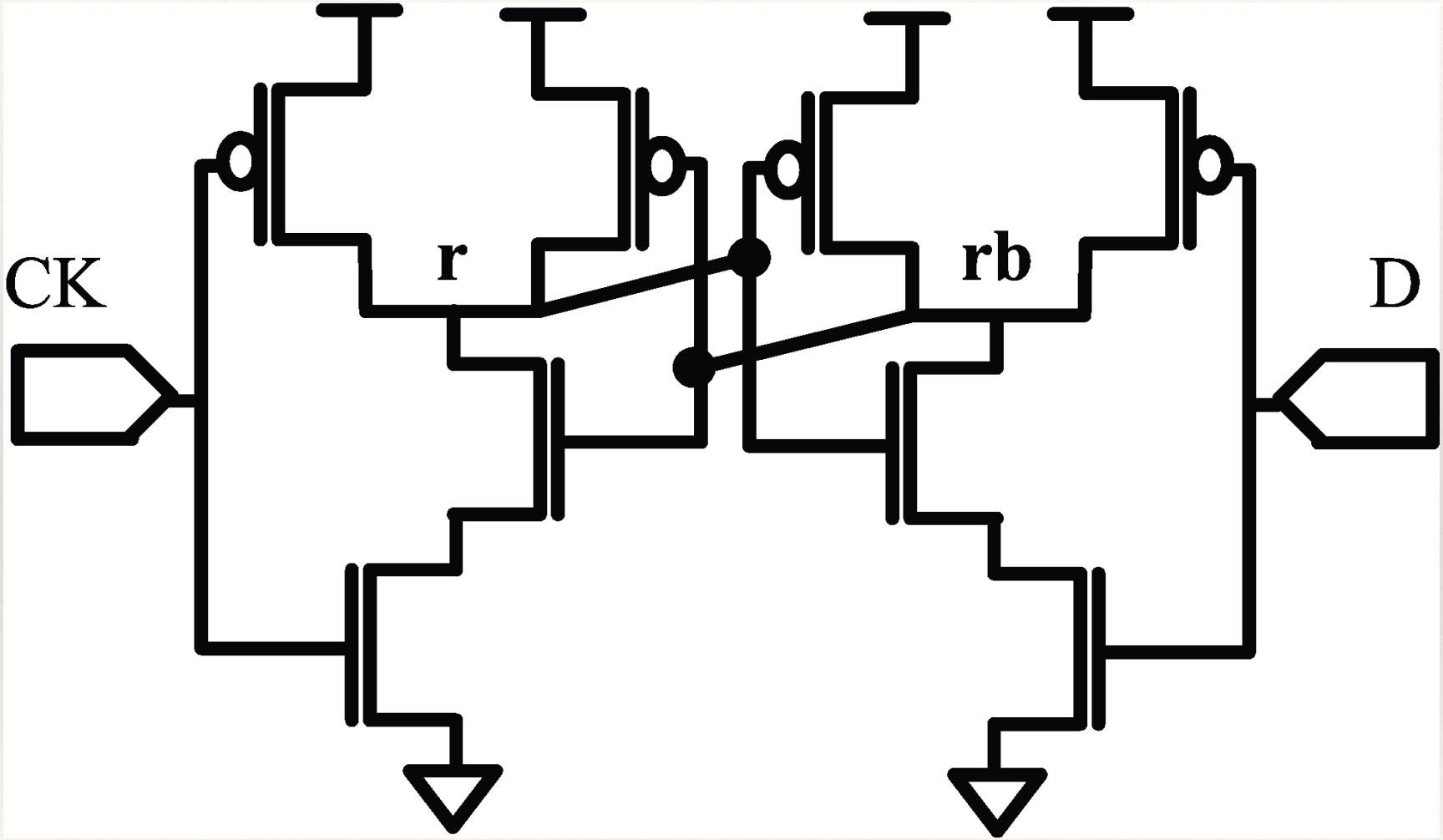
- Want only **random** variation to determine outputs
- Any design skews are consistent across chips
- 4 segments through each stage must have matched nominal delays
  - Careful layout techniques (parasitics)
- Many noise sources are common-mode (temperature, voltage, etc)
  - All 4 paths through stage get slower/faster together
- What type of arbiter to use?

Is This a Good Arbiter?

- What happens if both inputs rise at same time?
- Extra inverter delay biases arbiter towards the rb=0 state



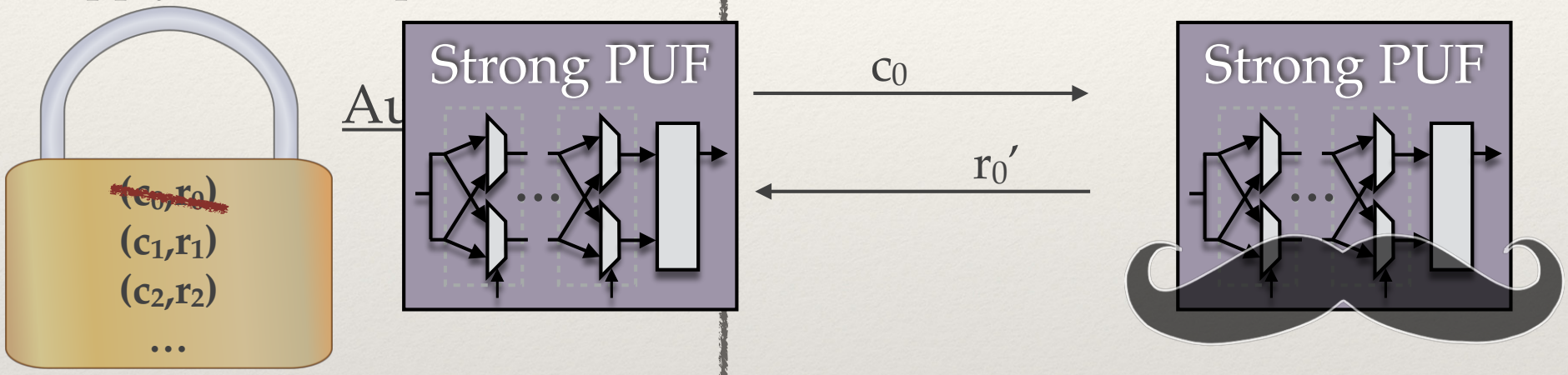
Is This a Good Arbiter?



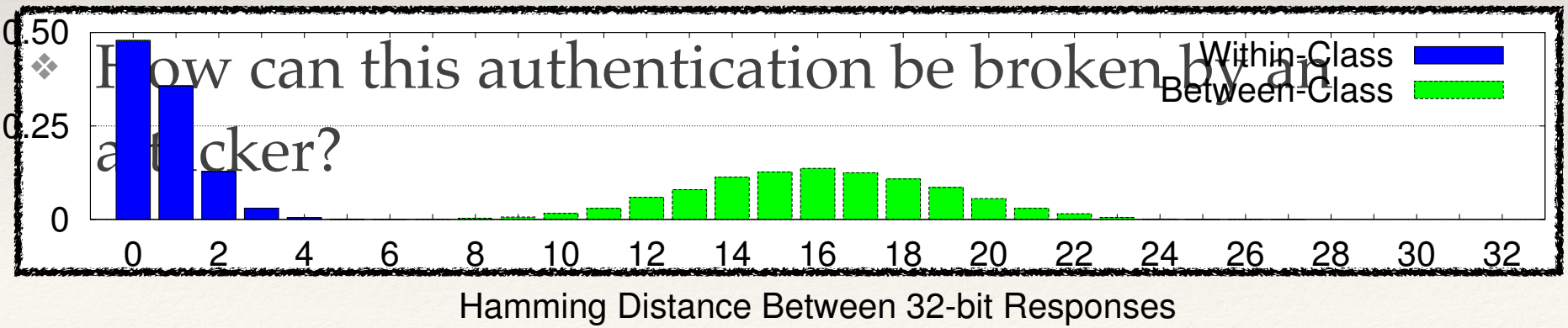
Authentication using Strong PUF

Enroll PUF

- Choose random challenges
- Apply and store private CRPs

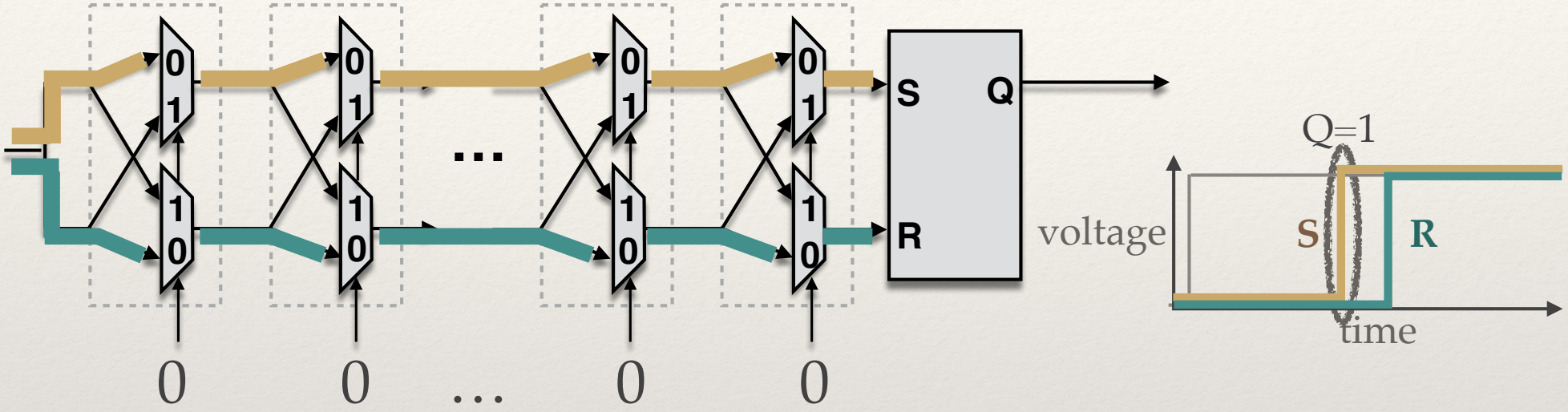


- How can this authentication be broken by an attacker?



Modeling Attacks on Arbiter PUF

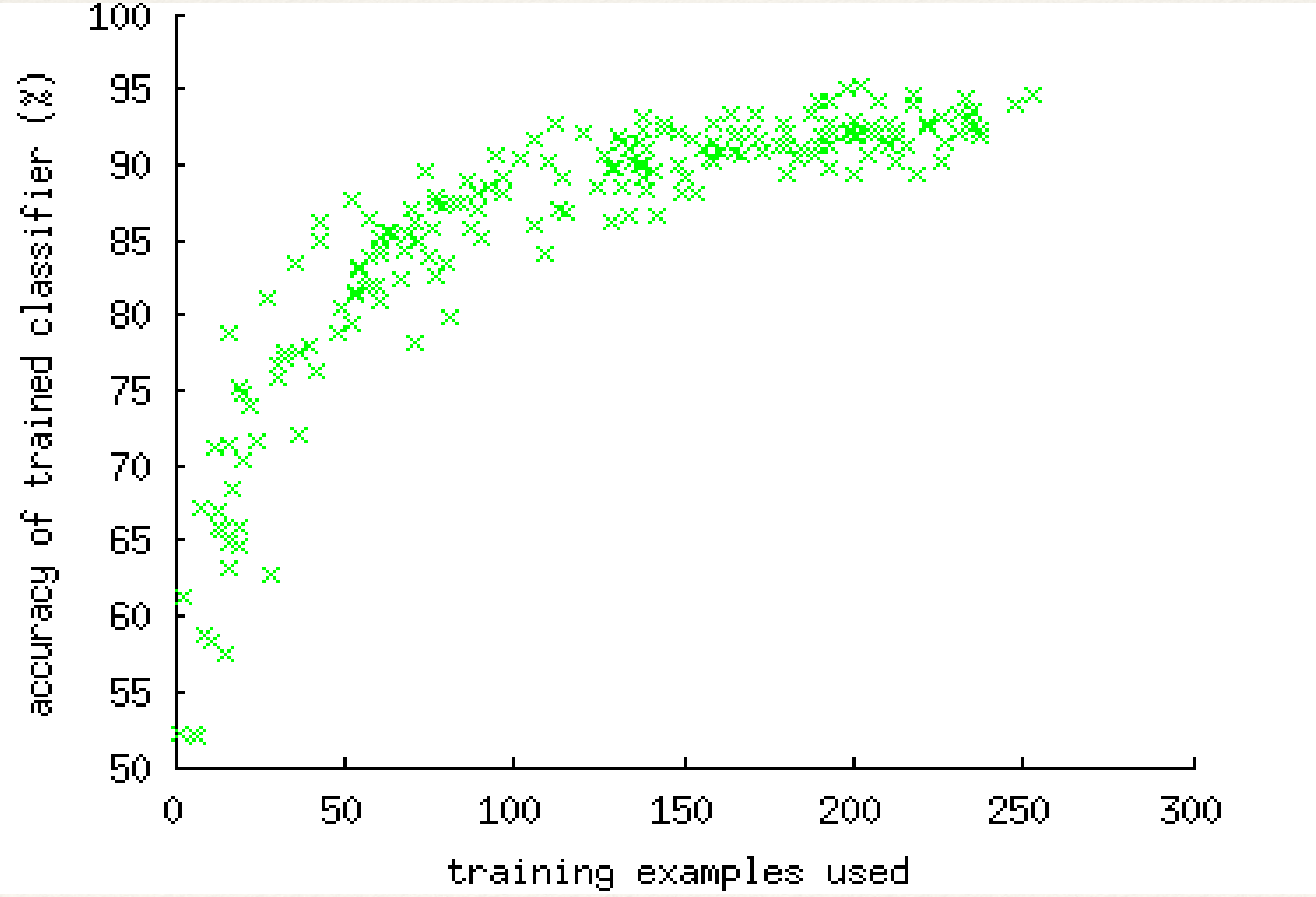
[D. Lim et al., '05]



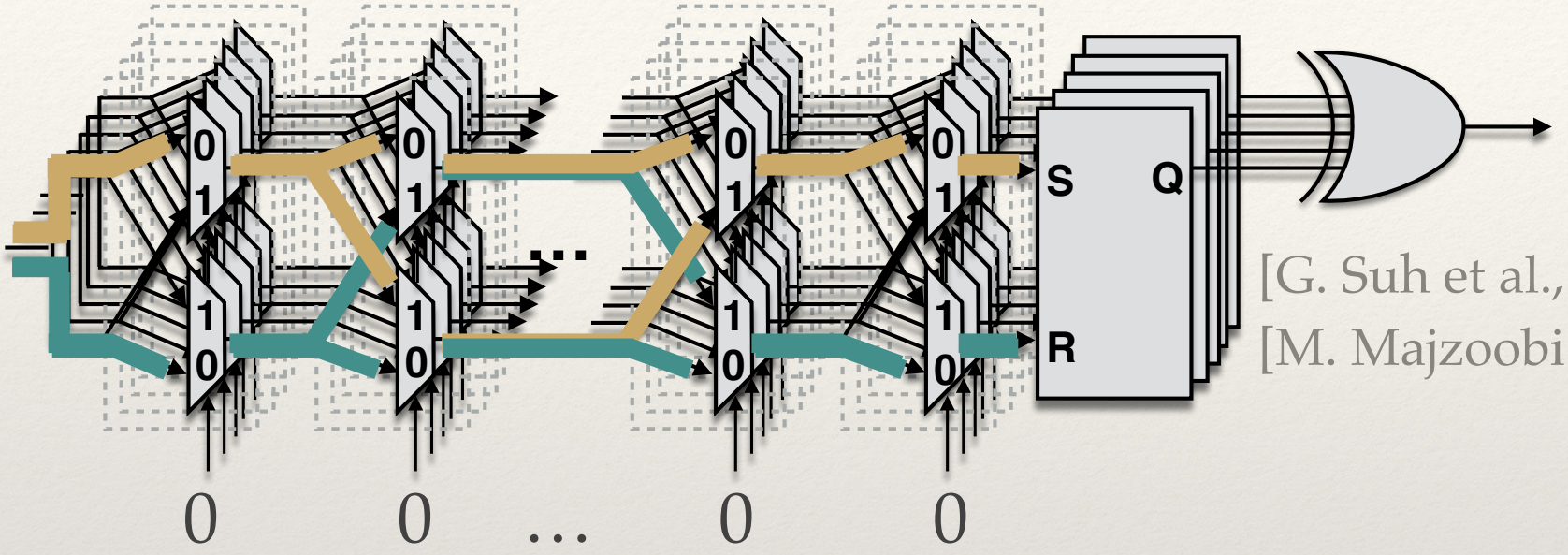
- PUF to a challenge is based on sum of path delays
  - Observing many outputs makes it possible to learn path delays and predict responses
- Many different forms of modeling attacks:
  - Support Vector Machines Classifier
  - Logistic Regression/ Gradient Descent, Linear Programming

Modeling Attack Results

- Basic arbiter PUF insecure
- Need to obfuscate CRPs



Attack-Resistant Arbiter PUF



[G. Suh et al., '07]
[M. Majzoobi et al., '08]

- Obfuscate CRPs to prevent modeling
- Arms race between designers and attackers
- Sufficiently large XOR Arbiter PUF considered secure until recently

Outline

1. PUFs
2. Arbiter PUFs (Strong PUF)
3. **SRAM Power-up PUFs (Weak PUF)**

Weak PUF

- A way to generate secret keys
  - Key must be kept secret (why?)
  - Error correction needed (approximate keys not very useful for crypto)

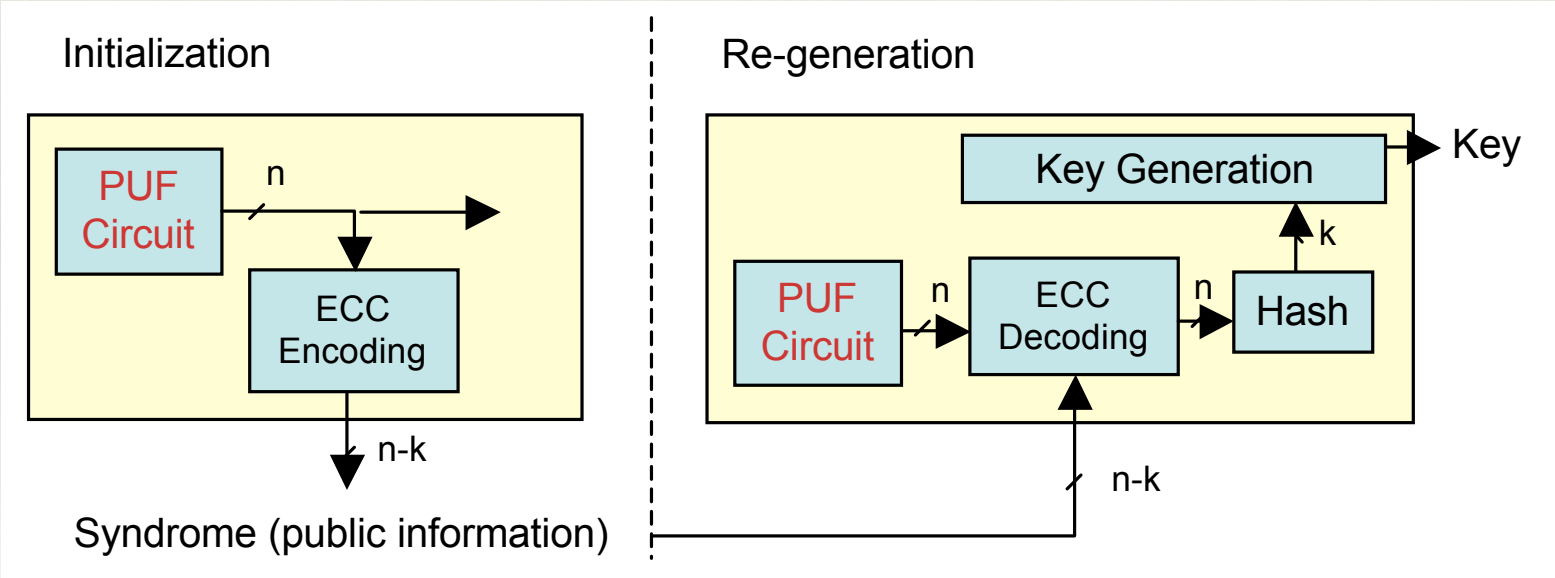
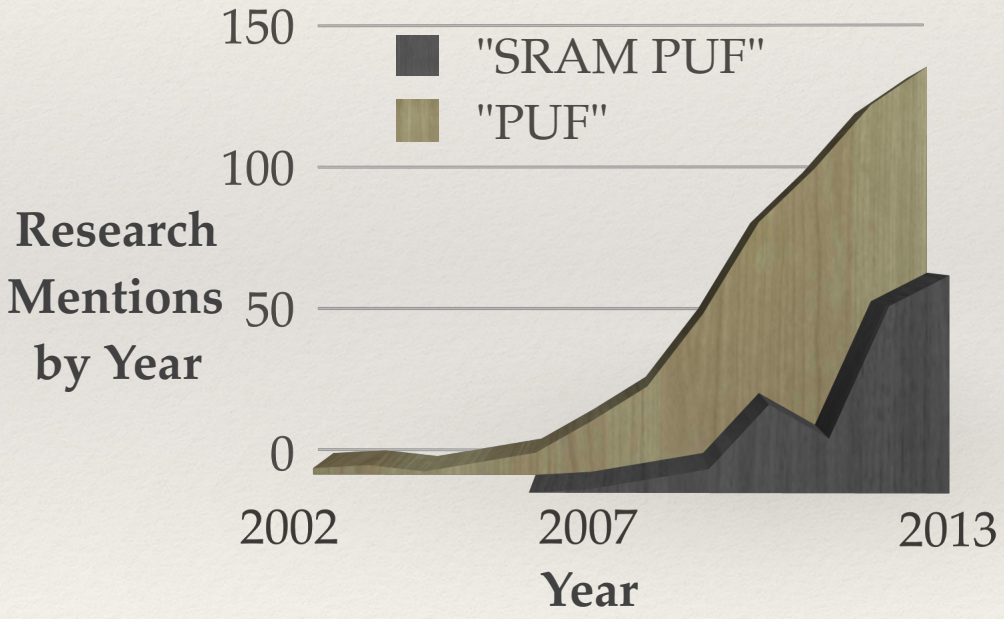


Figure 5: Cryptographic key generation with PUFs.

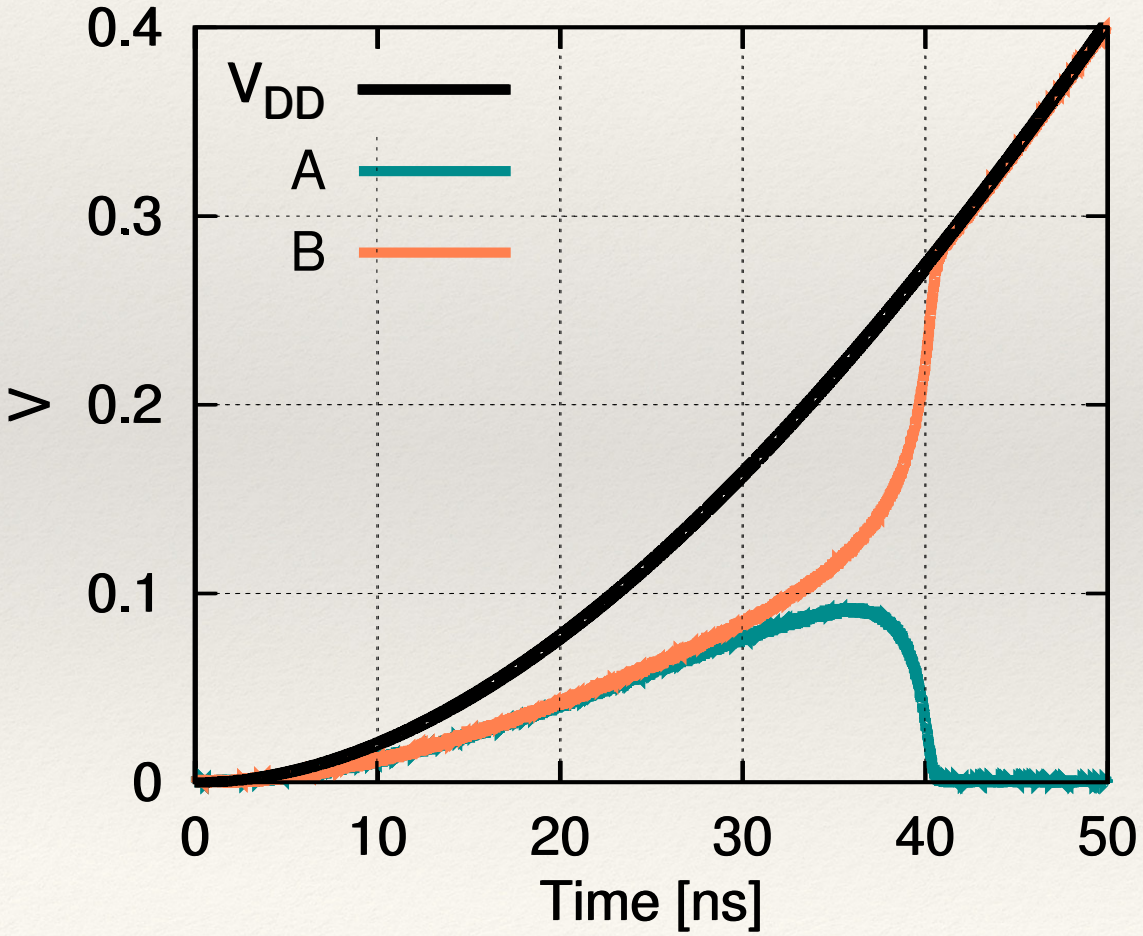
Examples of Weak PUFs

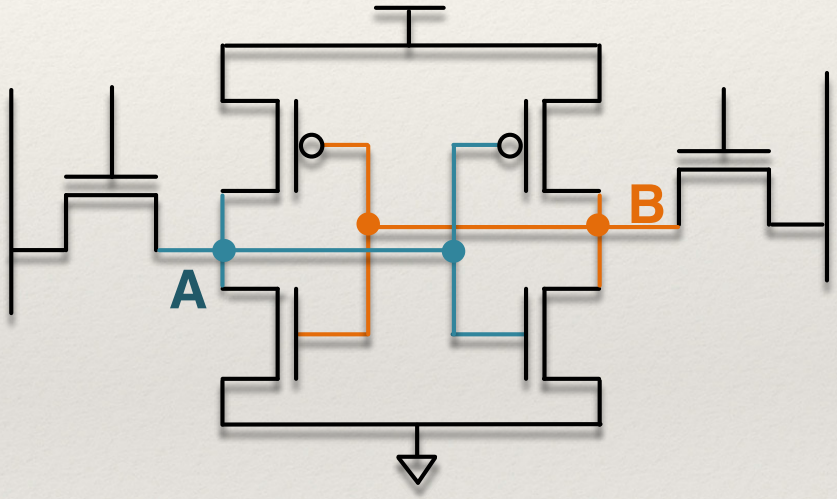
- Using custom circuits
  - Drain currents [Lofstrom et al. '02]
  - Capacitive coating PUF [Tuyls et al. '06]
  - Cross-coupled devices [Su et al. '07]
  - Data Retention Voltage
  - Sense amps [Bhargava et al. '10]
- Using existing circuits
  - Clock skew [Yao et al.'13]
  - Flash latency [Prabhu et al. '11]
  - Power-up SRAM state [Guajardo et al. '07, Holcomb et al. '07]



SRAM Power-up

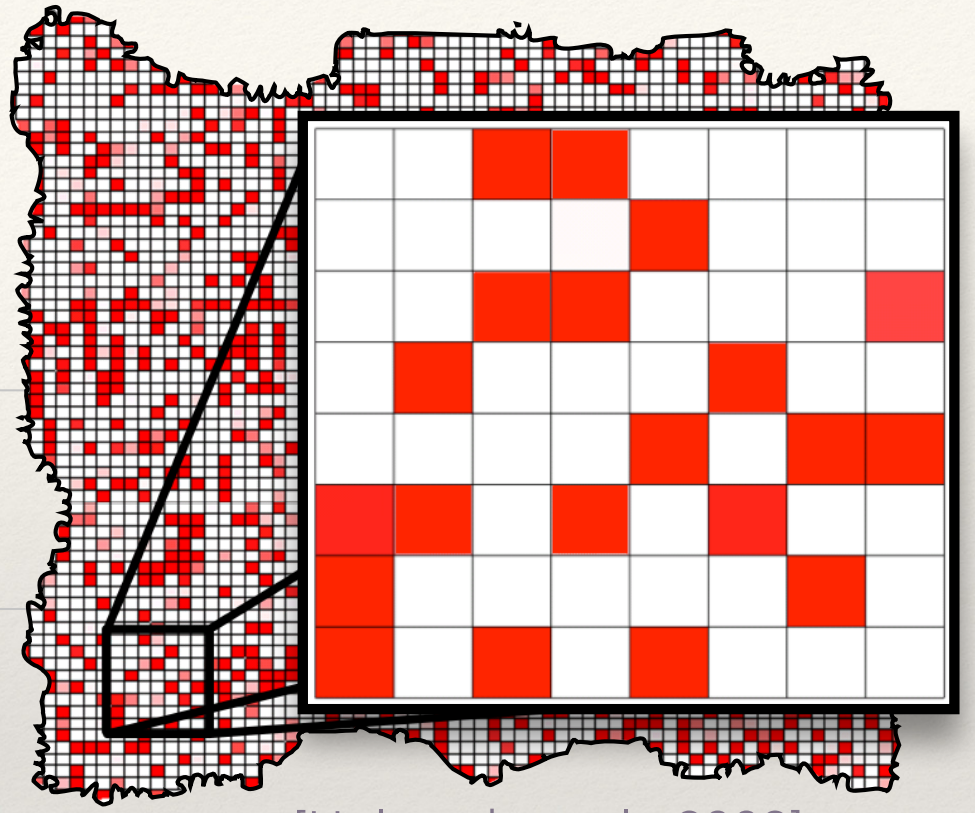
- Power-up sensitive to variations
- Uncorrelated across cells and chips
- Just turn on chip and read out RAM





Power-up Fingerprint

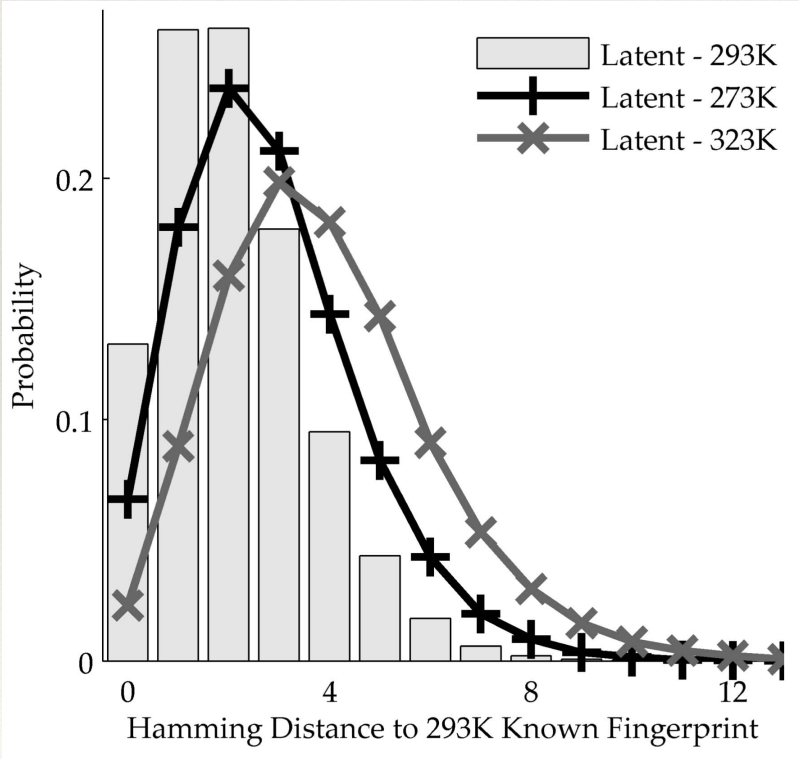
- 64-bit fingerprints
- Population size of 5,120



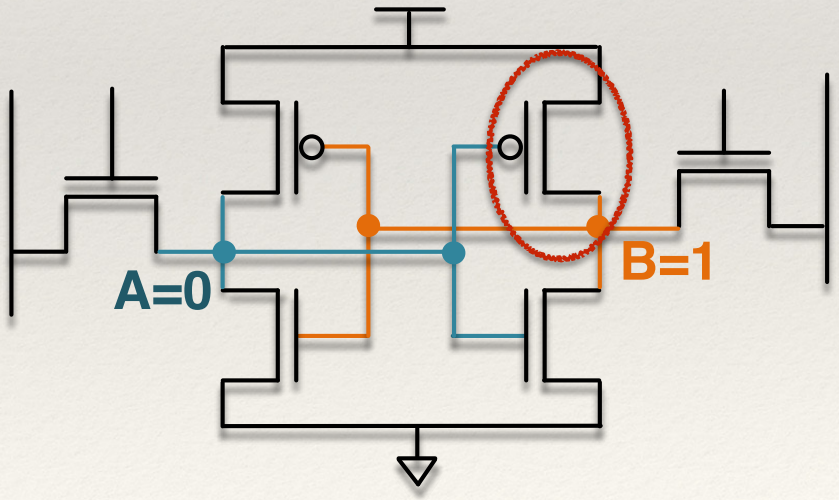
[Holcomb et al., 2009]

Reliability

- Temperature
  - slight increase of within-class distance
  - (but largely common mode)
  - Error correction handles this



- Negative Bias Temperature Instability (NBTI)
  - Increase threshold of stressed PMOS
  - Causes next power-up state to slightly favor the opposite of stored state (storing 0 causes 1)
  - Stressed transistor slow to turn on
  - Recovery occurs

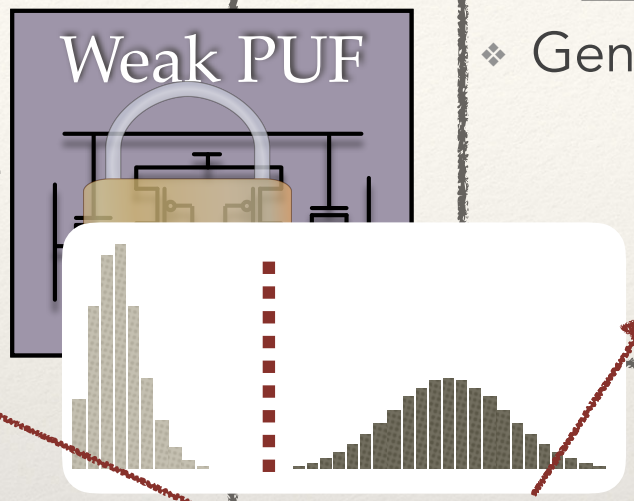


Power-up State PUF as Secret Key

Enroll PUF at Manufacture

- Read power-up state **r**
- Choose key **k** and derive helper data **h**:

  h = r ⊕ Encode(k)
- Store **h** with PUF
- Disable access to power-up state



k is reliable if r ≈ r′

Generate Key in Field

- Generate **x = r′ ⊕ h**

  = r′ ⊕ r ⊕ Encode(k)

  errors

  k = Decode(x)

- Technique to generate a reliable device-tied secret key **k**
- Only nonvolatile value **h** is present when chip is unpowered
- Volatile values **k** and **r** must be protected

code offset construction [Dodis et al. '08]

Possible Attacks

- Inverting cryptographic hash infeasible, so modeling attacks don't work (different than strong PUF)
- Invasive readout of SRAM to get key [Helfmeier et al]
  - Photons emitted from NMOS transistor in saturation
  - Requires many read samples
  - Difficulty unclear on modern SRAM cells
  - What is the impact of bitline capacitance?

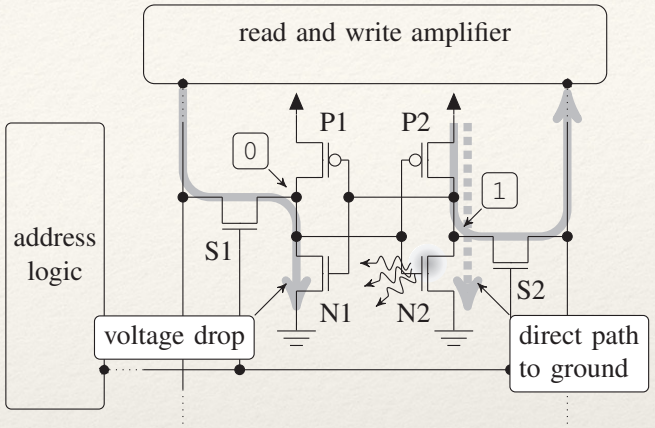


Fig. 1: Schematic and read operation of a 6T SRAM. The solid arrows depict the current injected by the read amplifiers for the specified states. The dotted arrows correspond to the resulting current to ground. During this process, transistor N2 is in saturation and emits light.

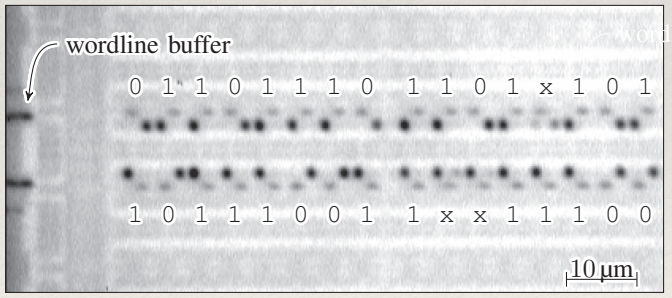
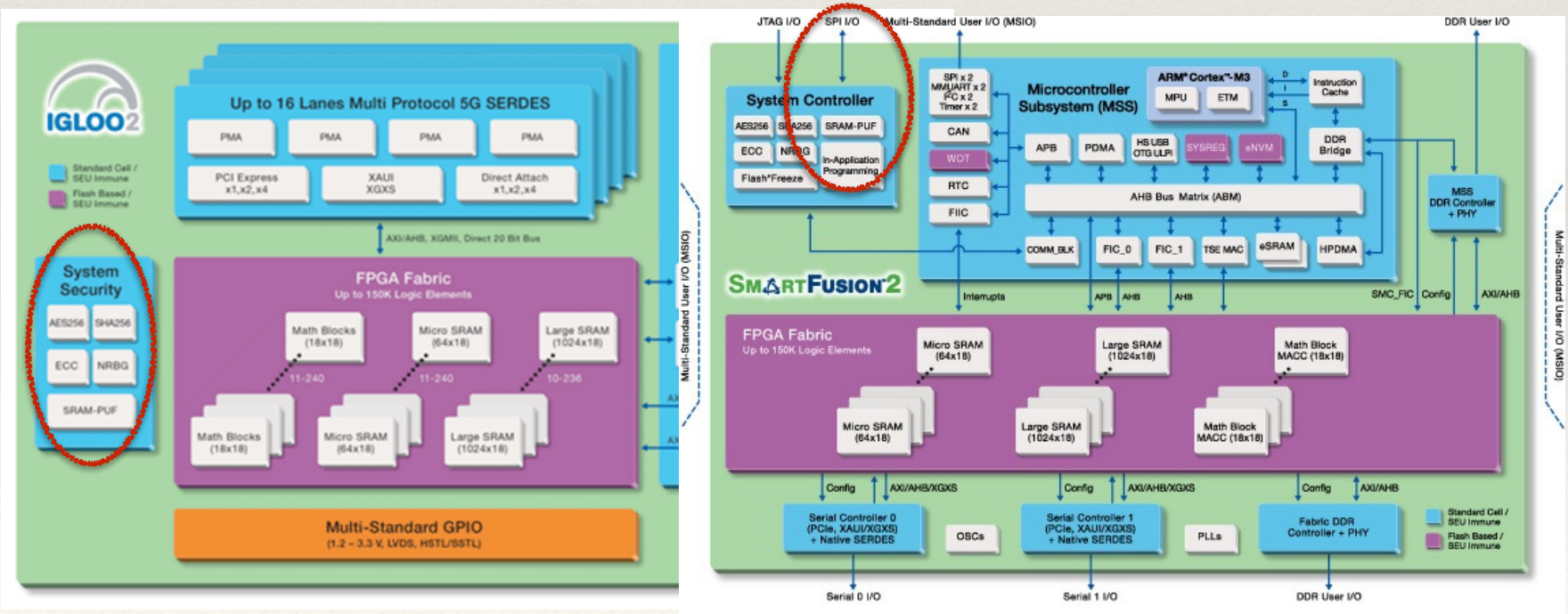


Fig. 4: Reflected image of the SRAM array with overlaid emission image. Bits 9, 10 in the bottom row and bit 12 in the top row are marked "don't care" `x`. These cells started up with both logical states `0` and `1` during integration. Assuming a fuzzy extractor algorithm is used on the PUF response, these bits potentially do not need to be modified [11].

PUF Industry

- Delay-based (Arbiter) PUFs sold by Verayo (from MIT research)
- SRAM power-up PUF IP sold by Intrinsic ID (Philips spinoff)
  - Licensed for use in MicroSemi products (maybe others?)
  - Discovered by both UMass and IID at same time in 2007
  - SRAM ID had been earlier discovered by Paul Layman (not for security)



Review Question

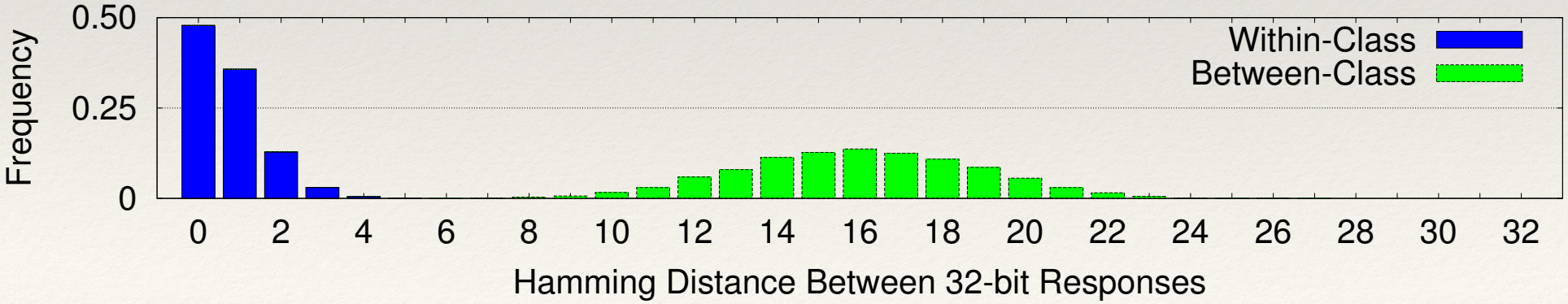
- What happens when between-class and within-class response distances overlap in strong PUFs? Weak PUFs?
  - How would you change the system to prevent this problem?
- SRAM PUFs only generate responses at power-up. What simple circuit change would allow responses to be generated at any time other than power-up?

Review Question

- Why is error correction needed in weak (secret key) style of PUFs?
- How does switching energy per response bit vary with the number of stages in Arbiter PUF? What about switching energy per time (i.e. dynamic power), assuming that the Arbiter PUF has no idle time between one challenge and the next?
- What are 2 ways of generating 64-bit Arbiter PUF responses? Which uses more area? Which uses more energy (ignoring leakage)?

Review Question (658 only)

- Imagine an arbiter PUF with an arbiter that has a timing skew across its two inputs, as we saw in one of the arbiters in slides
  - How does this impact between-class and within-class Hamming distance of responses? Show on histogram.
  - The manufacturer is embarrassed about the timing skew problem, and proposes the following solution: an extra stage is added to the PUF, and the challenge bit of that stage is programmed as a 0 on 50% of chips, and as a 1 on the other 50%. Does this fix the skew problem? Show how this would change the within-class and between-class distances on histogram.



PUFs beyond CMOS

ARTICLES
https://doi.org/10.1038/s41928-018-0146-5
nature electronics

A provable key destruction scheme based on memristive crossbar arrays

Hao Jiang[1,2], Can Li[1,2], Rui Zhang[1], Peng Yan[1], Peng Lin[1], Yunning Li[1], J. Joshua Yang[1*], Daniel Holcomb[1*] and Qiangfei Xia[1*]

Digital keys are commonly used in today's hardware security systems. However, the provable destruction of these keys after use remains a challenging problem. Most security primitives built using traditional complementary metal–oxide–semiconductor transistors are not well suited to address this issue because of their volatility and unreliability at small scales. Here we show that the unique physical fingerprint of a 128 × 64 hafnium oxide memristor crossbar array integrated with transistors is capable of provable key destruction. The fingerprint is extracted by comparing the conductance of neighbouring memristors, and it can be revealed only if a digital key stored on the same array is erased. On the basis of this provable key destruction technique, we propose a protocol for logic locking/unlocking that can support secure outsourcing of integrated circuit manufacturing. By leveraging the unique properties of memristors, including reconfigurability and variability, our chip demonstrates the integration of security, memory and computing functionalities into the same circuits, and could be used to develop more secure, compact and efficient memristive hardware systems.