CSL

Flash Memory for Ubiquitous Hardware Security Function

G. Edward Suh

Cornell University

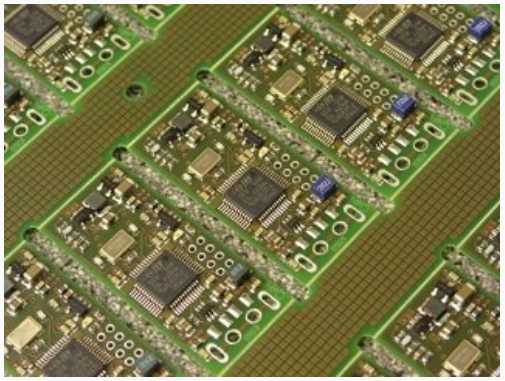
Hardware Security Functions

- Hardware complements software in building more secure systems
  - Provide entropy → true random numbers
  - Tamper resistance → device authentication, SW isolation
  - Efficiency → fine-grained monitoring

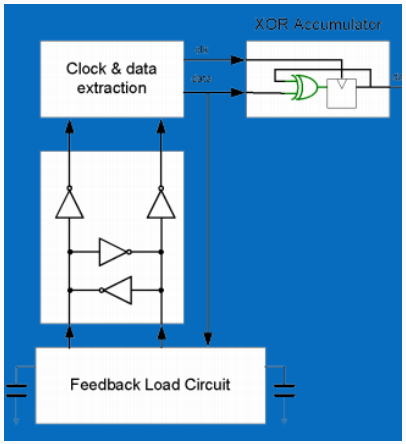
- BUT, often requires custom hardware
  - Expensive to build
  - Not applicable to legacy systems

Today's Hardware Security Functions

- Hardware random number generation (RNG)

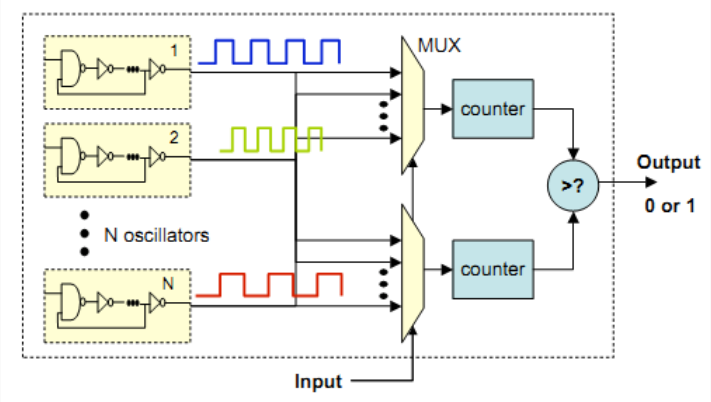


Avalanche noise,
Entropy key product

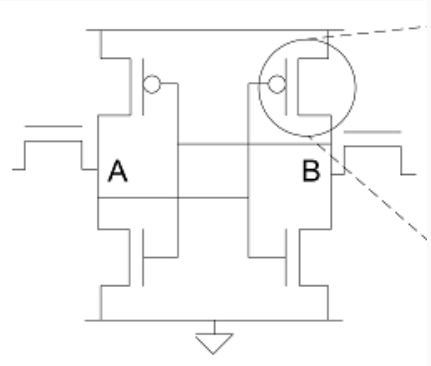


Metastability,
Cox *et al.*, Hot Chips, 2011

- Device Fingerprinting



Identical ring oscillators,
Suh *et al.*, DAC 2007



Initial state of SRAM,
Holcomb *et al.*, RFID security, 2007

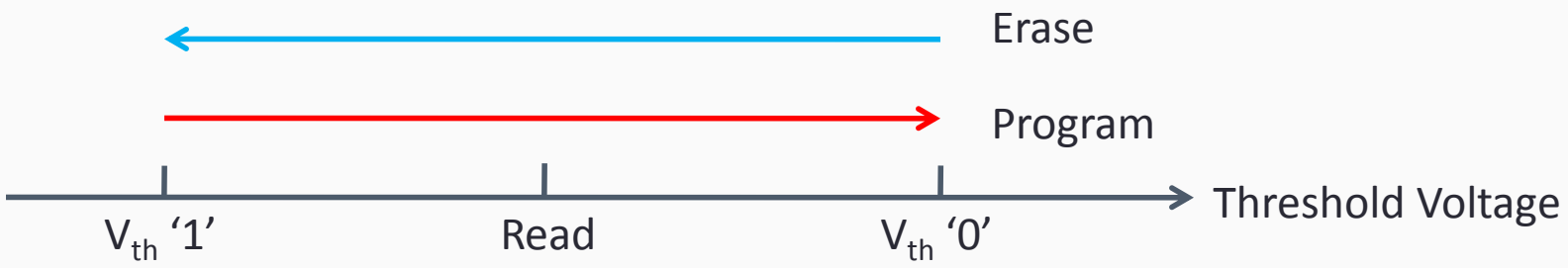
Using Existing Hardware for Security

- Noise and variations are inherent in any HW system
  - Often seen as challenges to overcome
- Turning challenges into features
  - Noise → True random numbers
  - Manufacturing variations → Device fingerprints or secrets
- Flash memory is ubiquitous
  - Mobile devices, SSD, USB, etc

Flash-Based Security Functions [IEEE SP 2012]

- Flash–based security functions
  - True random number generator
  - Device fingerprinting
  - (Hidden storage)
- Use standard chip interface, but more direct access
  - Erase/program/read to chip addresses w/ no ECC
  - Accept RESET command when the chip is busy
- Pure software implementations
  - Works with TI MSP430F2274 Microcontroller(16-bit RISC mixed-signal, used in sensor networks)
  - TI OMAP4430 / NVIDIA Tegra 3 (ARM architecture) should also work (smartphones--android, galaxy, kindle fire)

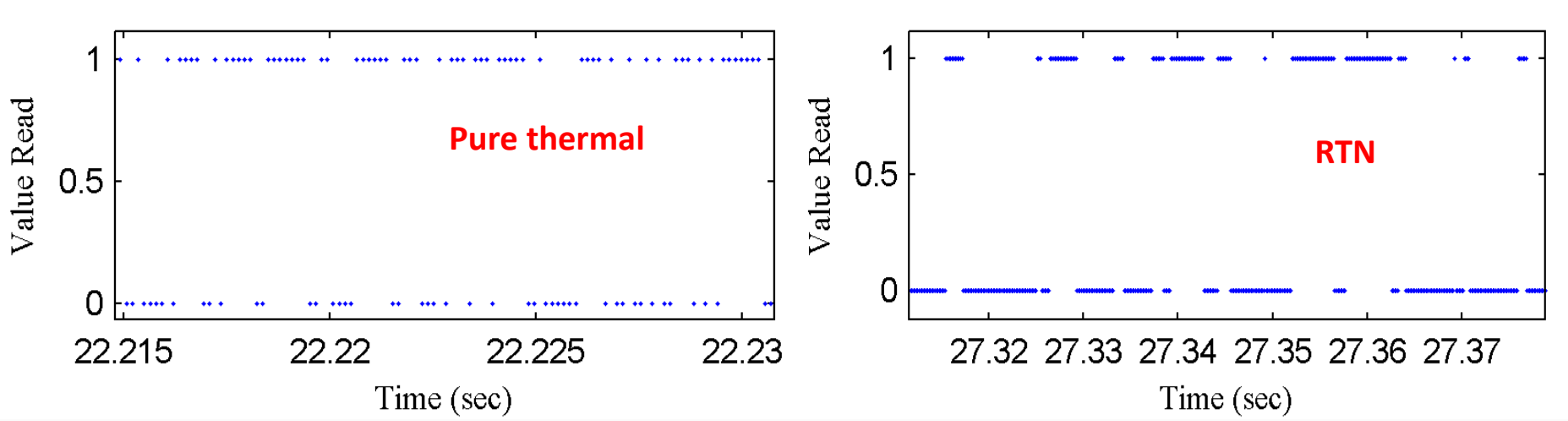
Flash Memory Operations



Page 0	Page 1	Page 2	.....	Block
010011111000	011111111010	010010011001	.....	Original Data
111111111111	111111111111	111111111111	.....	After Erase
111111111111	000000000000	111111111111	.....	Program Page 1

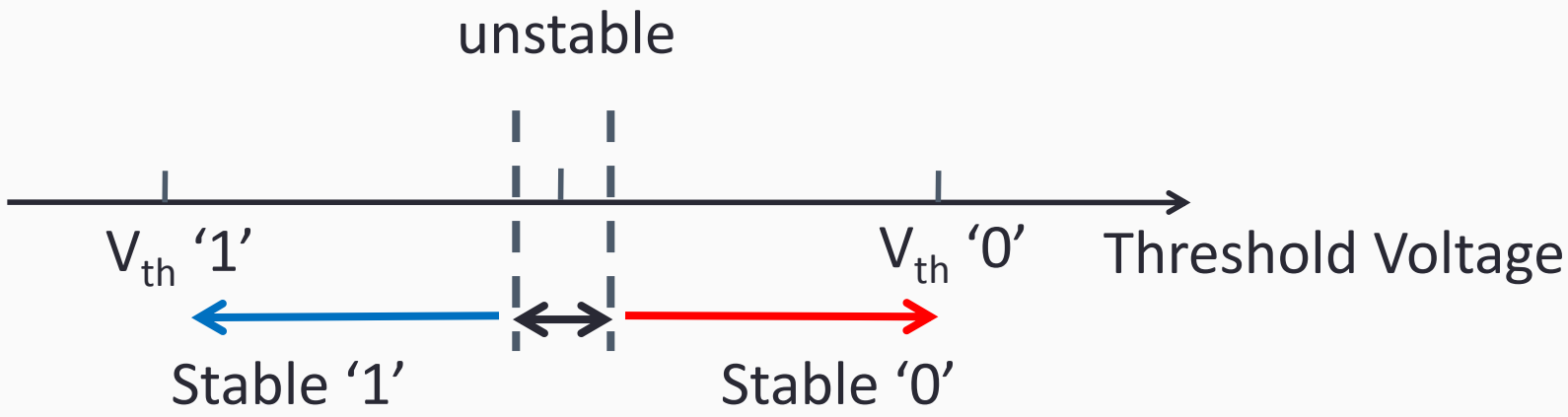
Noises in Flash Memory

- Two types of noises
  - Thermal noise (without quantum property)
  - Random telegraph noise (RTN, caused by single electron capture and emission in the device, quantum noise)



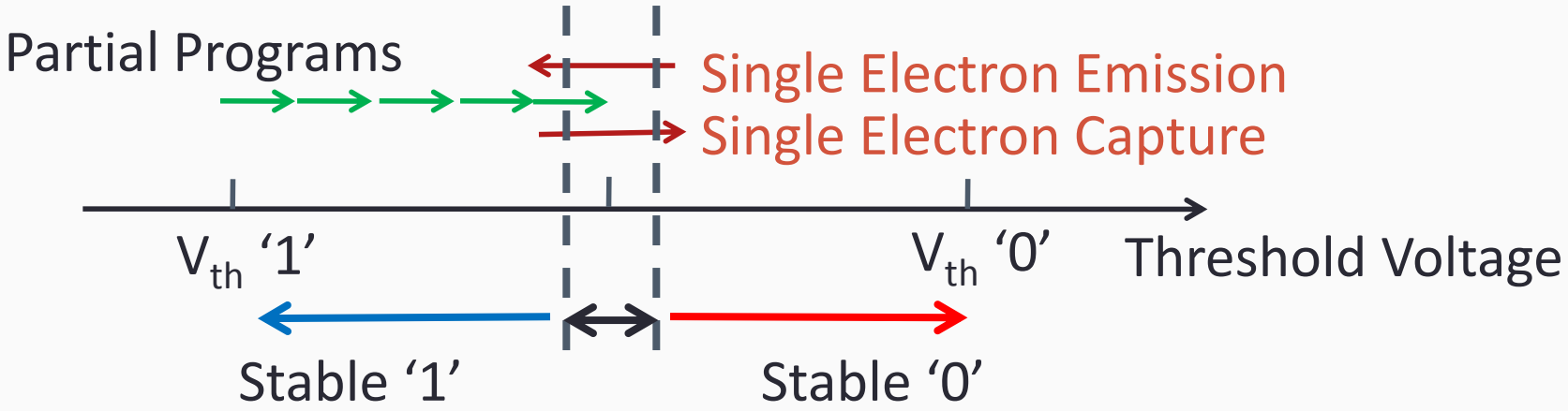
Challenge

- Digital abstraction is built to hide the noise
- Flash bits are programmed to either stable ‘1’ or stable ‘0’
  - Give sufficient noise margins



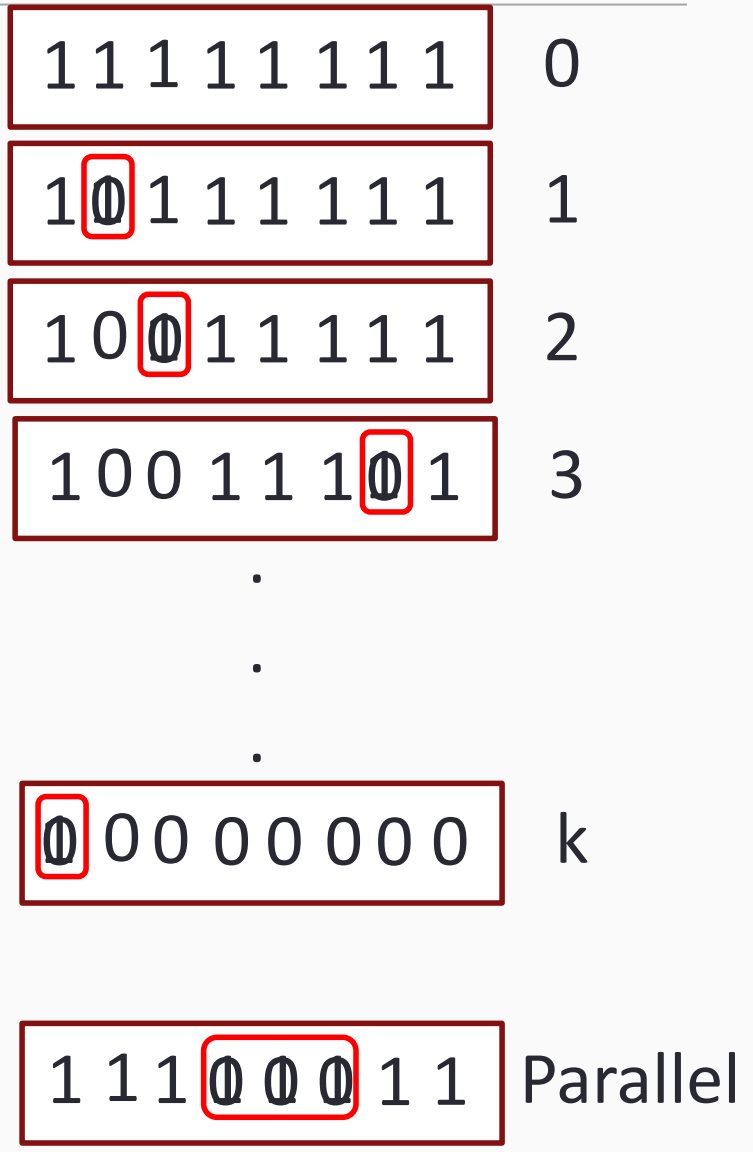
Solution: Partial Programming

- Standard Flash interfaces (such as ONFI – Open NAND Flash Interface) support an abort operation
  - Program/erase can be interrupted
  - Enables partial programming of individual bits
- Put a bit in a "half-programmed" state



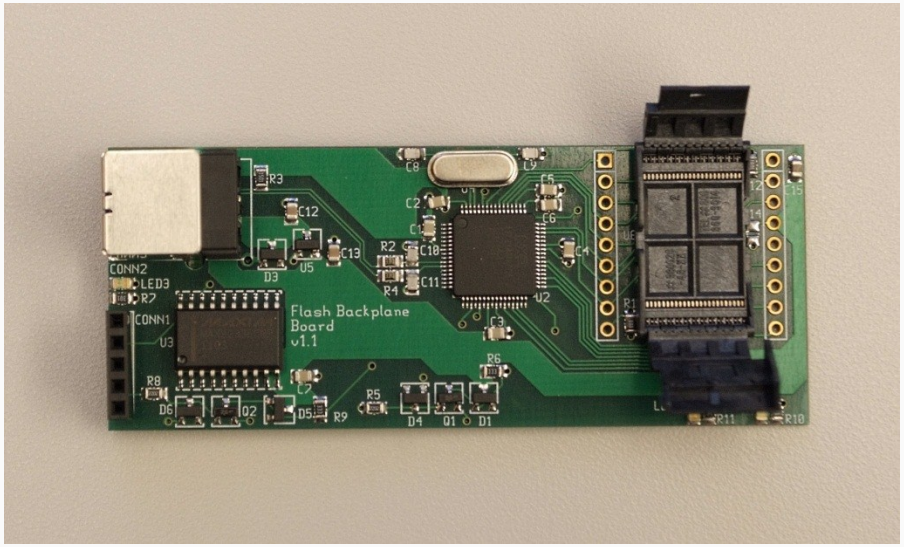
RNG Algorithm

- Erase
- Partial program
- Read the page N times, if one oscillating bit shows RTN, record its position and partial program number
- Repeat above 2 steps until all bits are programmed
- Erase, partial program all RTN bits to proper level
- Read these bits M times
- Debiasing



Experimental Setup

- Flash test board
  - ARM microcontroller
  - Socket for commercial off-the-shelf (COTS) Flash
  - USB output
  - All components available COTS



- Flash chips

Manufacturer	Capacity	Quantity	Technology
Numonyx	4Gbit	3	57nm SLC
Hynix	4Gbit	10	SLC
Micron	2Gbit	24	34nm SLC
Micron	16Gbit	5	MLC

Experimental Results

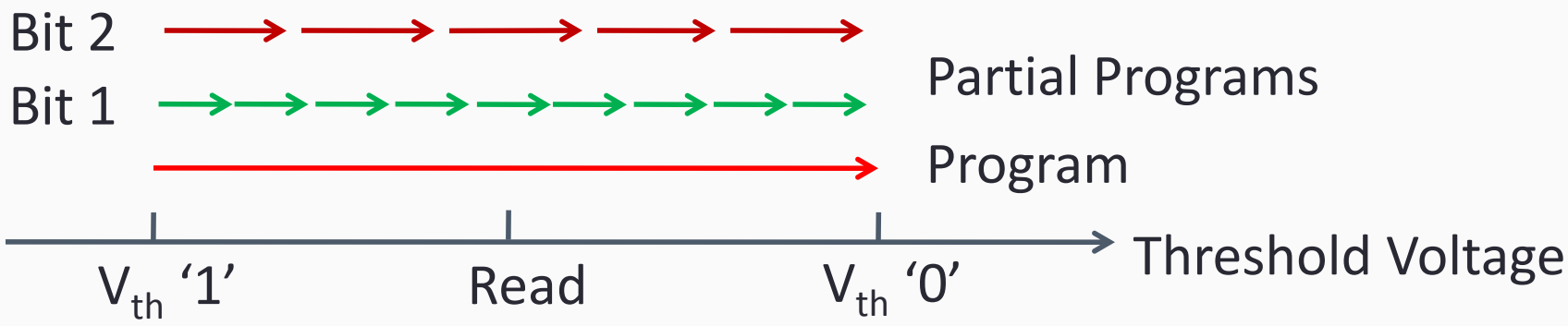
- Use NIST Statistical Test Suite 2.2.1 (Aug. 2010)
  - 15 tests
- Pass all 15 tests in NIST statistical test suite
  - Flash bits with pure RTN: 10 sequences of 200,000 bits
  - Flash bits with RTN+thermal: 10 sequences of 600,00 bits
- Works even at a low temperature and after aging
  - Tested at -5 °C and -80 °C
- 1-10Kbits/s using pure RTN bits

Flash Chip Fingerprints

- Process variation makes every Flash bit unique
  - Threshold voltage (program/erase time)
  - Wear-out from P/E cycles
  - Program/read disturb
  - Quantization margins in sense amplifiers
- Can be used for fingerprints, device-specific keys, etc.
  - No explicit programming is required
  - Difficult to clone
- However, digital interfaces are built to hide such analog variations

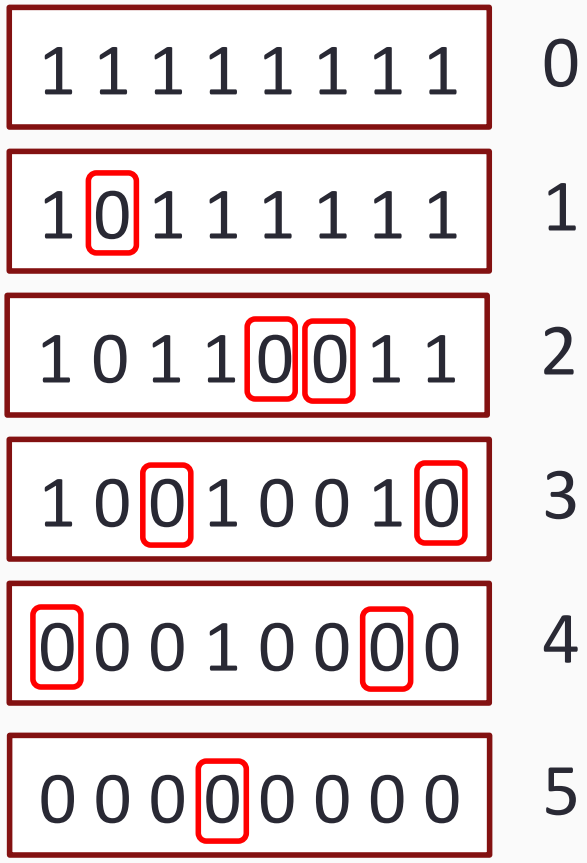
How to Expose the Variations?

- Standard Flash interfaces (such as ONFI – Open NAND Flash Interface) support an abort operation
  - Program/erase can be interrupted
  - Enables partial programming of individual bits



Fingerprinting Algorithm

- Erase a block, pick a page
- Partial program
- Read the page and record the bits flipped in this partial program
- Repeat the above two steps until most bits flipped

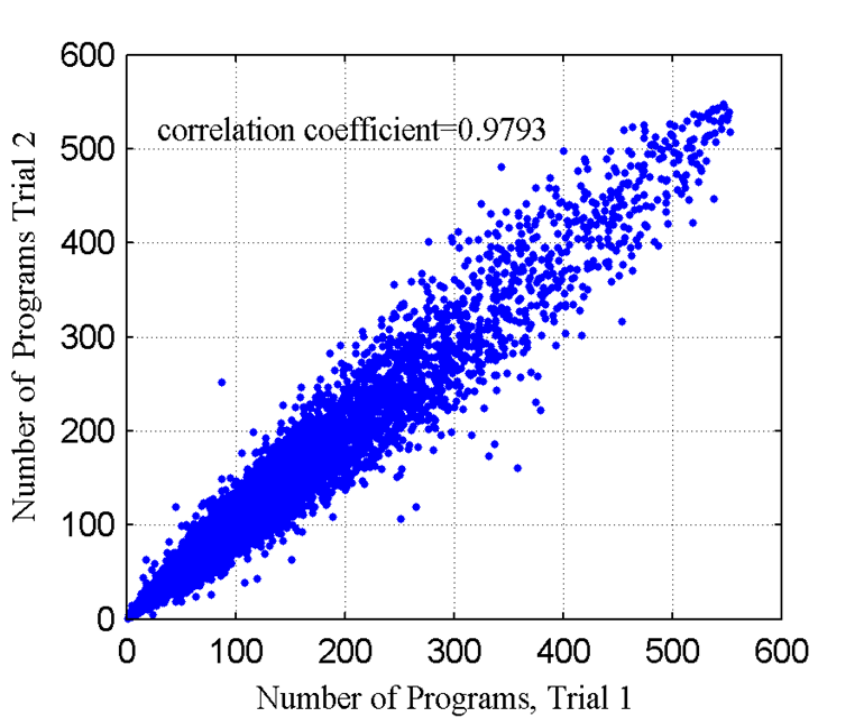


Final results: 4 1 3 5 2 2 4 3

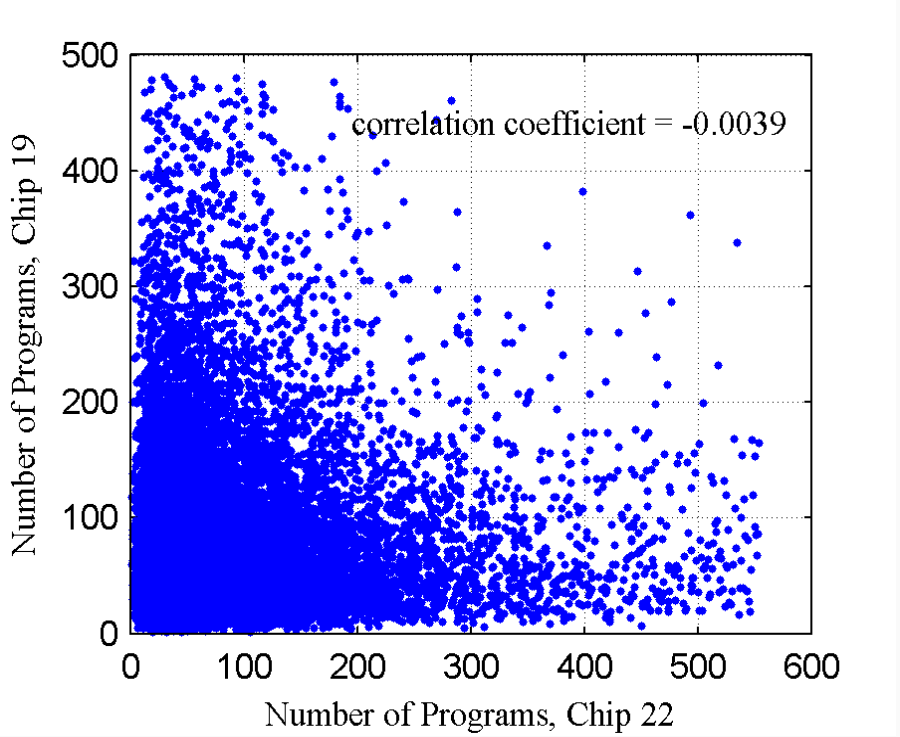
Partial Program Number Fingerprints

Correlation Function: $$P(X,Y) = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

Same page, same chip

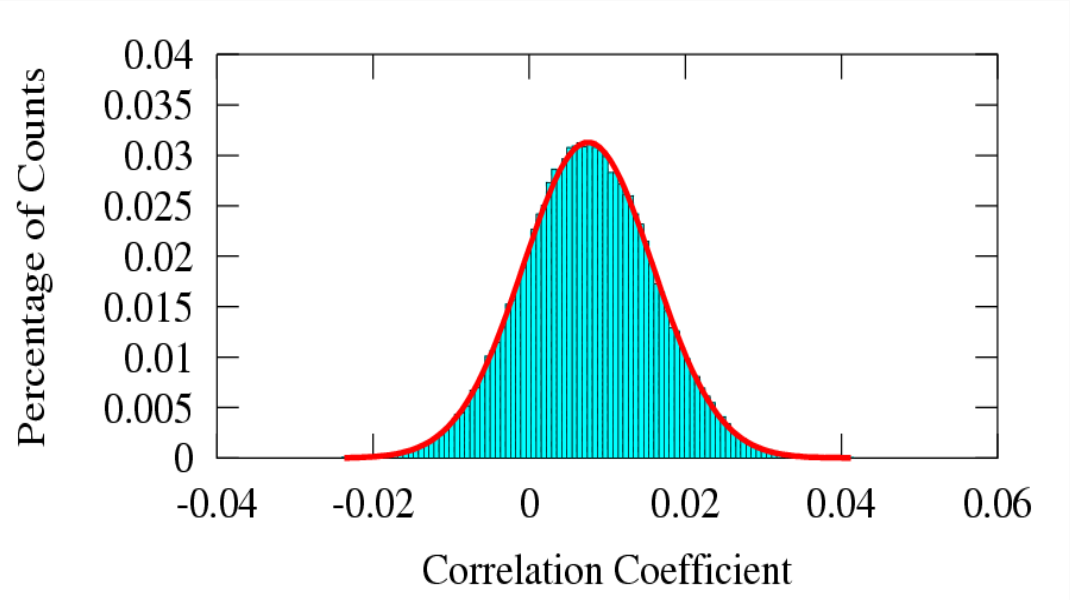


Same page, different chips



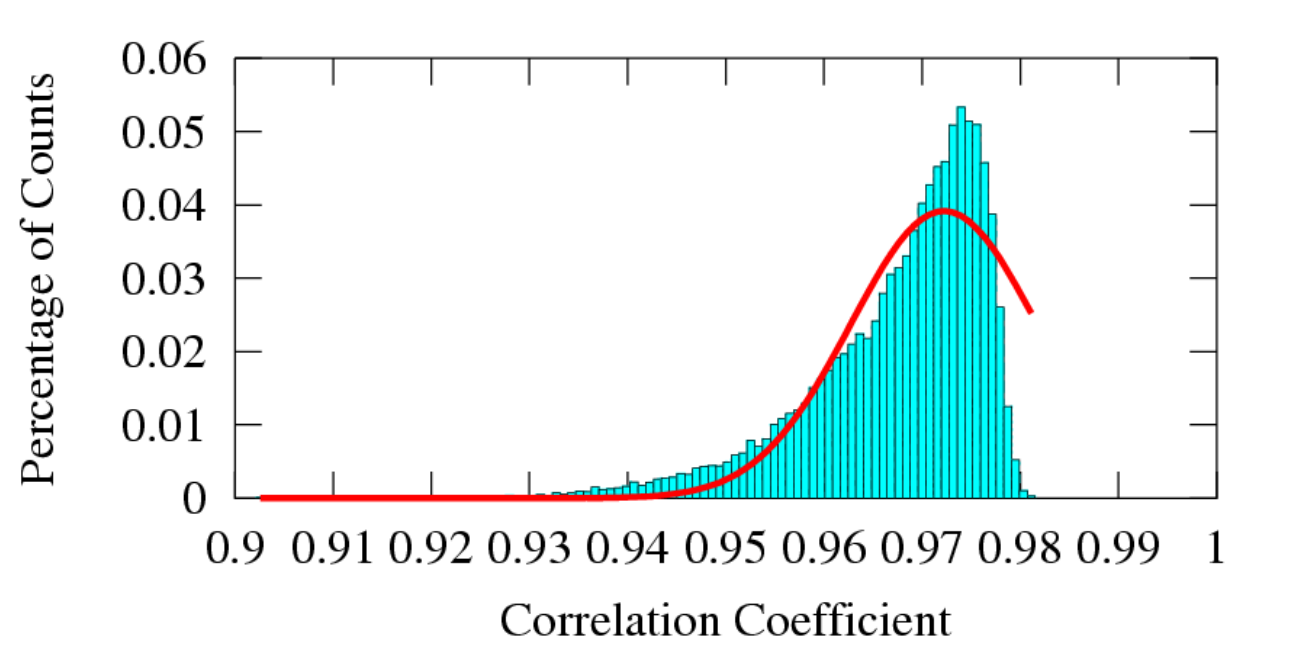
Uniqueness (Inter-Chip Variations)

- Compare measurements of the same page on different chips
  - 66,240 pairs compared
    - (24 chips choose 2) × 24 pages × 10 measurements
  - Histogram with Gaussian fit in red outline



Robustness (Intra-chip variations)

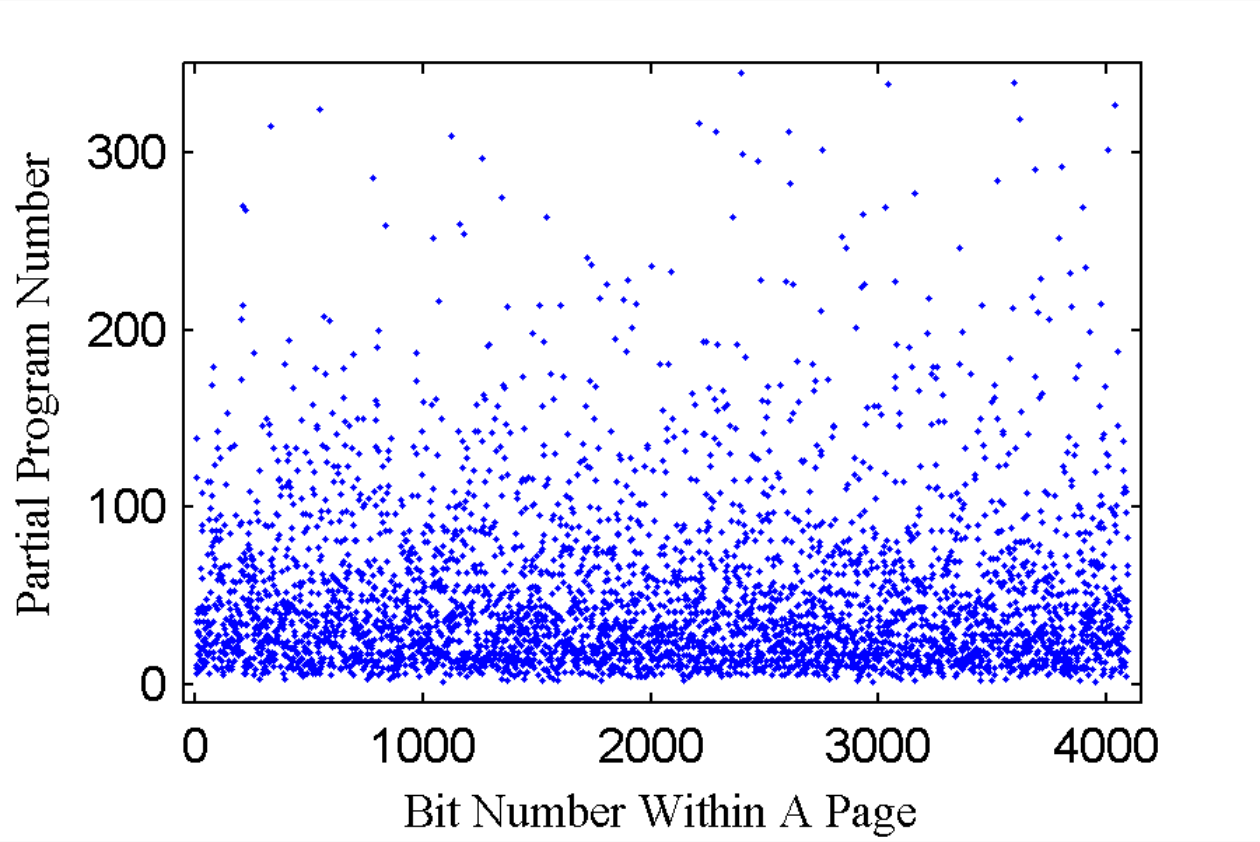
- Compare multiple measurements from the same page on the same chip
  - 25,920 comparisons



Experimental Results

- Low false positive and negative rates
  - Use the fingerprints to identify/authenticate chips
  - Assume Gaussian distribution / using a full page
  - False positive: **10^{-539}**, false negative: **10^{-815}**
- Robust across temperature ranges and aging
  - Tested from -5 °C to 60 °C
  - Up to 500,000 P/E cycles (lifetime < ~100,000)
- Time
  - ~10 seconds for all 16,384 bits in one page
  - **< 1 second** for a 1,024-bit fingerprint

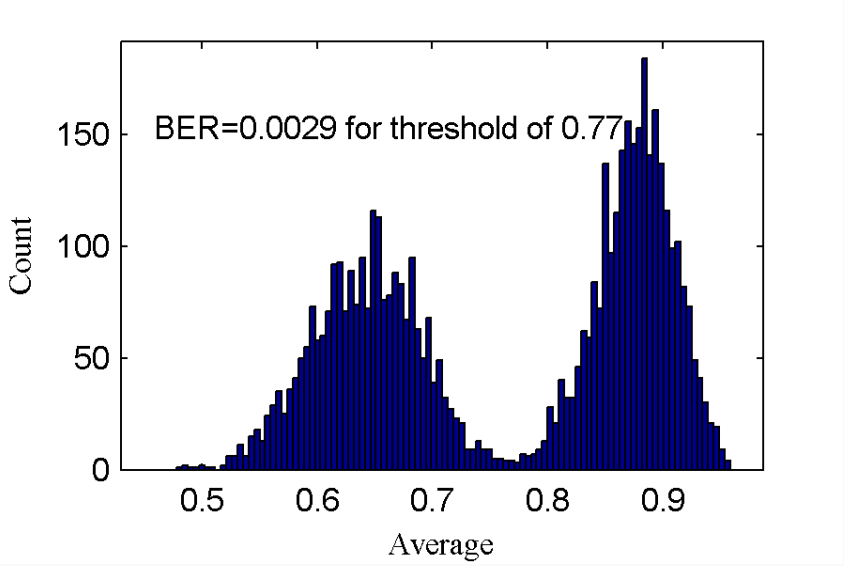
Program Time Distribution

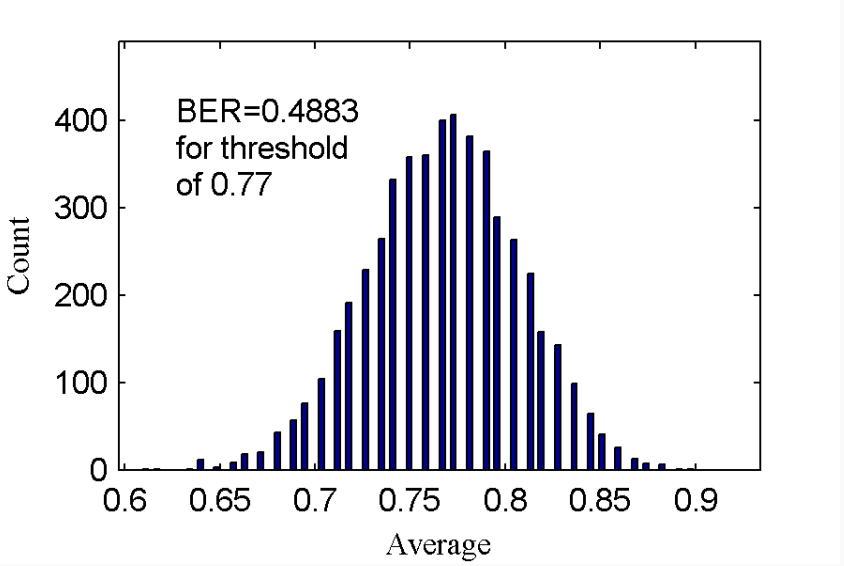


- Program time of a bit gets faster w/ aging
  - Writing ‘0’ stresses a bit more than writing ‘1’

Hiding Information in Program Time

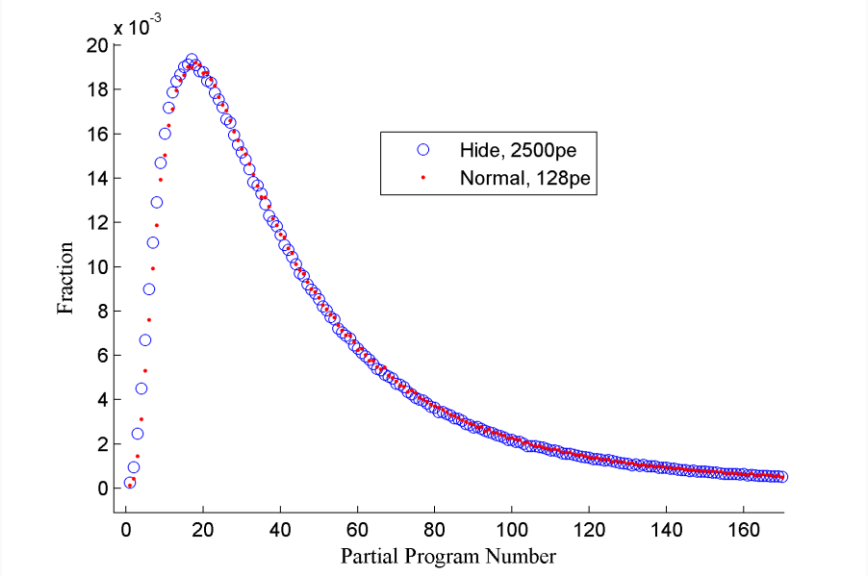
- Select a group of bits (50-100 bits) that will represent one hidden bit
- Stress each group based on a value to store
  - Store '1' → write '0' many times
  - Store '0' → write '1' many times





Detecting and Erasing Hidden Bits

- Timing for normal Flash operations
  - Program, erase, read time
  - Dominated by the number of P/E cycles
- Per-bit program time
  - Still no visible pattern
  - Slow to measure
- Difficult to erase
  - Erasing a page does not erase the hidden information
  - Need to selectively stress locations w/ hidden bit of ‘0’



Summary

- Flash memory is everywhere and can be used for security purposes without hardware changes
- Flash memory as a True RNG
  - Quantum noise (RTN) and thermal noise
  - Viable across temperature ranges, aging
- Flash memory device fingerprinting
  - Robust and unique signatures
  - Resistant to temperature variations and aging
- Hiding information in Flash program time
  - Analog characteristics can be intentionally affected
  - Difficult to detect, difficult to erase