Extracting Unique Fingerprints From Flash Memory Devices

Pravin Prabhu¹, Ameen Akel¹, Laura M. Grupp¹, Wing-Kei S. Yu², G. Edward Suh², Edwin Kan², and Steven Swanson¹

¹Non-volatile Systems Laboratory, UCSD CSE
²Cornell University
The Flash Juggernaut
Flash Device Authentication

• Can we authenticate each flash chip?
  – Distinguish genuine flash chips from counterfeits
  – Authenticate a device with a flash chip
Physical Unclonable Functions (PUFs)

- Because of random manufacturing variations, no two Integrated Circuits are identical - even those using same mask
  - Hard to remove or predict in advance
  - Relative variation increases as feature sizes shrink
  - Variation persists, despite $ billions spent to control it

- We can generate fingerprints from unique analog characteristics of each IC: Response = PUF(Challenge)
  - Inexpensive; intrinsic to each device; effectively unclonable

- This work introduces a PUF based on flash chips
Outline

• Flash memory overview
• Experimental infrastructure
• Flash-based Physically Unclonable Functions (FPUFs)
  – Usage Model
  – Desiderata
  – Our FPUFs
• Conclusions
Flash Operations

Program 0V 5V 20V
0V 1V
Floating Gate
20V Erase
0V 0V

Read

SLC: Single Level Cell
== 1 bit

MLC: Multi Level Cell
== 2 bits
NAND Flash Basics

Page: 0 1 2 3 4  n-4 n-3 n-2 n-1 n

Block 0

Block 1

Block 2

Block n

Erase Blocks

Program Pages
Flash Failure Mechanisms

- Program/Erase (PE) Wear
  - Permanent damaged to the gate oxide at each flash cell
  - After 3000 (MLC) – 100,000 (SLC) PE cycles, a cell becomes unreliable
- Program disturb
  - Data corruption caused by interference from programming adjacent cells.
  - No permanent damage
Experimental Setup

- Custom-Built Daughter Board
- Xilinx XUP Board
- EZ to integrate similar capabilities into existing systems
## The Test Subjects

<table>
<thead>
<tr>
<th>Chip Name</th>
<th>Node (nm)</th>
<th>Bytes</th>
<th>Pages</th>
<th>Planes</th>
<th>Dies</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-SLC2</td>
<td>2048</td>
<td>64</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>A-SLC4</td>
<td>2048</td>
<td>64</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>A-SLC8</td>
<td>2048</td>
<td>64</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>B-SLC2</td>
<td>50</td>
<td>2048</td>
<td>64</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B-SLC4</td>
<td>72</td>
<td>2048</td>
<td>64</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>E-SLC8</td>
<td>2048</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chip Name</th>
<th>Node (nm)</th>
<th>Bytes</th>
<th>Pages</th>
<th>Planes</th>
<th>Dies</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-MLC16</td>
<td>4096</td>
<td>128</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>B-MLC8</td>
<td>72</td>
<td>2048</td>
<td>128</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B-MLC32</td>
<td>50</td>
<td>4096</td>
<td>128</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>B-MLC32-2</td>
<td>34</td>
<td>4096</td>
<td>256</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>B-MLC128</td>
<td>34</td>
<td>4096</td>
<td>128</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>B-MLC128-2</td>
<td>34</td>
<td>4096</td>
<td>256</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>C-MLC64</td>
<td>43</td>
<td>8192</td>
<td>128</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>D-MLC32</td>
<td>4096</td>
<td>128</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>E-MLC8</td>
<td>4096</td>
<td>128</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>F-MLC16</td>
<td>41</td>
<td>4096</td>
<td>128</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Outline

• Flash memory overview
• Experimental infrastructure
• Flash-based Physically Unclonable Functions (FPUFs)
  – Usage Model
  – Desiderata
  – Our FPUFs
• Conclusions
Flash-based Signatures

• Cell-level variation in flash devices makes each chip unique.
• Unique, unforgeable flash chip signatures have several uses
  – Device identification
  – Supply chain integrity
Challenge-Response Based Authentication

- Create CRPs for IC with PUF when IC is in your possession
- Use CRPs to subsequently authenticate IC throughout the supply-chain and post-deployment
  - Use each CRP only once → prevent “replay”
Signature Characteristics

• Selectivity – an FPUF should be able to reliably distinguish between flash devices
• Speed – Computing an FPUF should be fast
• “Unforgeable” – It should be prohibitively difficult to forge the FPUF
• Non-Destructive – Extracting an FPUF should not wear out the flash device.
Basic Recipe for an FPUF

1. Identify an aspect flash chip behavior that varies based on manufacturing inconsistencies
2. Measure the variation at a bit, page, or block level
3. Use the sequence of measured values as a signature
4. Use statistical correlation to determine whether two signatures are for the same device.

Signatures Correlated?
Program Disturb FPUF

• Program one page repeatedly
• For each bit in the adjacent page
  – How many programs before the bit flips?
• The sequence of counts is a signature.
Signature Selectivity

Signatures from Different Pages

Signatures from the Same Page

Error Cycle (E-MLC-2 Trial 1)

Error Cycle (E-MLC-1 Trial 1)

Error Cycle (E-MLC-1 Trial 2)
Selectivity for Program Disturb

- Same Chip, Same Page: $R > 0.9$
- Different Chip, Same Page: $R < 0.1$
- Different pages, Different chips: $R < 0.1$
- Same Page, Same Chip: $R > 0.9$
Program Disturb Latency

• Extract an program-disturb signature takes up to 5 minutes
  – Some usage models require many signatures from each chip
  – 5 minutes is prohibitively slow in these cases.
• Can we extract a useful signature with fewer program operations?
Reducing Programs/Signature

5000 Programs = ~ 50 sec
Forging Program Disturb FPUFs

• Forging an FPUF would require storing the signature in the flash device.
• If the signature contains more than one bit of information per flash cell, storing the signature in the chip is not possible.
• However, our signatures are noisy, so precise forgery is not required.
  – It is possible to lossily compress signatures.
Compressing the Signatures

• Raw signatures need 10 bits of program count information per flash cell

• We can quantizing program counts in to 4 values (i.e., the top two bits)
  – Quantized signatures correlate well ($R = 0.8$) with raw signatures

• The quantized signatures are not very compressible (entropy/bit is near 1)
  – It is still impossible to store the signature for every page in a flash chip
Program Disturb FPUF

• Selectivity: Very Good
• Speed: 1-5 Minutes per page
• Wear: 10,000 programs of the target page
• Forgeability: low
Other FPUFs

• Usable FPUFs
  – Per-bit program latency
  – Read disturb

• Unusable FPUFs
  – Per-block erase latency
  – Per-page read latency
  – Full page program latency (rather than bit-by-bit)
Conclusions

• FPUFs can provide a robust mechanism for identifying individual flash devices.
• Flash’s ubiquity makes them an attractive method for device identification
  – Inexpensive
  – Easy to implement
• FPUFs will become even more useful as flash manufacturing variation grows
Questions?