Authenticache: Harnessing Cache ECC for System Authentication

Anys Bacha and Radu Teodorescu
Department of Computer Science and Engineering
The Ohio State University
http://arch.cse.ohio-state.edu
Security and Everyday Computing

Security is now crucial to all computing markets, especially with the advent of IoT.
Security Challenges

- Password management
- Complexity due to different accounts having policies
Security Challenges

• Password management
• Complexity due to different accounts having policies

• Secure key storage
• Increases complexity for low cost IoT devices
Security Challenges

- Password management
  - Complexity due to different accounts having policies

- Secure key storage
  - Increases complexity for low cost IoT devices

- Software as a Service
  - Personal device at workplace increasing security risks
Physical Unclonable Functions (PUF)
Physical Unclonable Functions (PUF)
Physical Unclonable Functions (PUF)

- Exploit randomness in silicon
Physical Unclonable Functions (PUF)

- Exploit randomness in silicon
- Systematic outputs unique to device
Enrollment

Silicon Fingerprints
PUF System Authentication

Enrollment

Silicon Fingerprints
Enrollment

Silicon Fingerprints
PUF System Authentication

Enrollment

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>01100</td>
<td>00110</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>00010</td>
<td>11001</td>
</tr>
</tbody>
</table>

Silicon Fingerprints
PUF System Authentication

Enrollment

Authentication

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>01100</td>
<td>00110</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>00010</td>
<td>11001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>01100</td>
<td>00110</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>00010</td>
<td>11001</td>
</tr>
</tbody>
</table>
PUF System Authentication

Enrollment

Authentication

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>01100</td>
<td>00110</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>00010</td>
<td>11001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>01100</td>
<td>00110</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>00010</td>
<td>11001</td>
</tr>
</tbody>
</table>
PUF System Authentication

**Enrollment**

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>01100</td>
<td>00110</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>00010</td>
<td>11001</td>
</tr>
</tbody>
</table>

**Authentication**

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>01100</td>
<td>00110</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>00010</td>
<td>11001</td>
</tr>
</tbody>
</table>
Enrollment

Challenge | Response
---|---
01100 | 00110
... | ...
00010 | 11001

Authentication

Challenge | Response
---|---
01100 | 00110
... | ...
00010 | 11001

Challenge | Response
---|---
01100 | 00110
... | ...
00010 | 11001
PUF System Authentication

Enrollment

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>01100</td>
<td>00110</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>00010</td>
<td>11001</td>
</tr>
</tbody>
</table>

Authentication

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>01100</td>
<td>00110</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>00010</td>
<td>11001</td>
</tr>
</tbody>
</table>

Authenticache: Harnessing Cache ECC for System Authentication

Anys Bacha
Related Work

- Arbiter PUF (Lee et al. VLSI’04)
- Signal traversing maze of cascaded switch blocks
Related Work

- Arbiter PUF (Lee et al. VLSI’04)
  - Signal traversing maze of cascaded switch blocks

- Ring Oscillator PUF (Suh et al. DAC’07)
  - Delay loops feeding oscillations into counters
Related Work

- Arbiter PUF (Lee et al. VLSI’04)
  - Signal traversing maze of cascaded switch blocks

- Ring Oscillator PUF (Suh et al. DAC’07)
  - Delay loops feeding oscillations into counters

- SRAM PUF (Guajardo et al. CHES’07)
  - Power-on states of 6T SRAM cell
Related Work

- Arbiter PUF (Lee et al. VLSI’04)
  - Signal traversing maze of cascaded switch blocks

Authenticache: No custom hardware
On-chip error correction logic in processor caches

- SRAM PUF (Guajardo et al. CHES’07)
  - Power-on states of 6T SRAM cell
Cache Errors as Silicon Fingerprints
Cache Errors as Silicon Fingerprints

- Caches optimized for density
Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
Cache Errors as Silicon Fingerprint

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

Intel 9560 Processor
Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

![Intel 9560 Processor]
Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

![Intel 9560 Processor](image-url)
Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

![Intel 9560 Processor]
Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

![Intel 9560 Processor](image_url)
Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

![Intel 9560 Processor](image)

![Graph](image)

- Repeatable
- 2 new errors/mV
- Relative Correctable Error Range (mV)

- L2
- L2
- L2
- L2

- L2
- L2
- L2
- L2
Cache Errors as Silicon Fingerprints

• Caches optimized for density
• Sensitive to process variation
• Itanium processor 8 L2 caches

Intel 9560 Processor

![Graph showing relative correctable error range (mV) vs cache lines and error count vs cache line address.]

- 2 new errors/mV
- Repeatable

Cache Line Address:
- L2
- L2
- L2
- L2
- L2
- L2
- L2
- L2
Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

Intel 9560 Processor

 повторное выполнение

- Repeatability of errors
- 2 new errors/mV

Relative Correctable Error Range (mV) vs. Cache Line Address

Error Count vs. Cache Line Address
Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

Intel 9560 Processor

![Graph showing cache lines and relative correctable error range](image)

- Repeatable
- 2 new errors/mV
- <1 overlap/cache
Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

Intel 9560 Processor

- Relative Correctable Error Range (mV)
- Cache Line Address
- Error Count
- Repeatable
- Unique
- <1 overlap/cache
The Authenticache System
The Authenticache System

Cache Layout

- Exploit process variation in LLC for randomness
• Exploit process variation in LLC for randomness

• Construct cache maps as a function of voltage and correctable errors
The Authenticache System

- Exploit process variation in LLC for randomness
- Construct cache maps as a function of voltage and correctable errors
Challenge and Response

Error Map

```plaintext
 e e e e
 e e e e
 e e e e
 e e e e
```

Anys Bacha

Authenticache: Harnessing Cache ECC for System Authentication
Challenge and Response

Error Map

\[ \text{Challenge} = (x_1, y_1, V_1), (x_2, y_2, V_2) \]
Challenge and Response

Error Map

$\text{Challenge} = (x_1, y_1, V_1), (x_2, y_2, V_2)$

A

B
Challenge and Response

Error Map

$V_1 = V_2$

$\text{Challenge} = (x_1, y_1, V_1), (x_2, y_2, V_2)$
Challenge and Response

**Challenge** = \((x_1, y_1, V_1), (x_2, y_2, V_2)\)

**Response** = \[
\begin{align*}
0, & \text{dist}(A, e_a) < \text{dist}(B, e_b) \\
1, & \text{dist}(A, e_a) \geq \text{dist}(B, e_b)
\end{align*}
\]

**Manhattan Distance**

Error Map

\(V_1 = V_2\)
Challenge and Response

**Challenge** = \((x_1, y_1, V_1), (x_2, y_2, V_2)\)

**Response** = 
\[
\begin{cases} 
0, & \text{dist}(A, e_a) < \text{dist}(B, e_b) \\
1, & \text{dist}(A, e_a) \geq \text{dist}(B, e_b)
\end{cases}
\]

Manhattan Distance

\(V_1 = V_2\)
**Challenge and Response**

**Authenticache: Harnessing Cache ECC for System Authentication**

**Challenge** = \((x_1, y_1, V_1), (x_2, y_2, V_2)\)

**Response** = \[
\begin{cases} 
0, & \text{dist}(A, e_a) < \text{dist}(B, e_b) \\
1, & \text{dist}(A, e_a) \geq \text{dist}(B, e_b)
\end{cases}
\]

**Manhattan Distance**

\(\text{dist}(A, e) = 5\)

\(\text{dist}(B, e) = 4\)

\(V_1 = V_2\)
Challenge and Response

Challenge = \((x_1, y_1, V_1), (x_2, y_2, V_2)\)

Response = \(\begin{cases} 
0, & \text{dist}(A, e_a) < \text{dist}(B, e_b) \\
1, & \text{dist}(A, e_a) \geq \text{dist}(B, e_b) 
\end{cases}\)

Manhattan Distance

\(V_1 = V_2\)
Challenge and Response

**Challenge** = \((x_1, y_1, V_1), (x_2, y_2, V_2)\)

- **A**
- **B**

**Response** = \(
\begin{cases} 
    0, & \text{dist}(A, e_a) < \text{dist}(B, e_b) \\
    1, & \text{dist}(A, e_a) \geq \text{dist}(B, e_b)
\end{cases}
\)

**Manhattan Distance**

\(\text{dist}(A, e) = 5\)

\(\text{dist}(B, e) = 4\)

\(V_1 = V_2\)

5 > 4

\[\text{Error Map}\]
Experimental Framework

- System:
  - BL860c-i4 Integrity Server from HP
  - 2x 9560 Itanium II CPUs

- Prototype in System Firmware
  - Thermal experiments through power virus

- Monte Carlo simulations
  - Different cache sizes
  - Different error maps and noise profiles
Identification and Noise

Identification in presence of environmental and measurement noise
Identification and Noise

Identification in presence of environmental and measurement noise
Identification in presence of environmental and measurement noise
Identification and Noise

Identification in presence of environmental and measurement noise
Identification and Noise

Identification in presence of environmental and measurement noise

Intra-chip (10% Noise)  Intra-chip (150% Noise)  Inter-chip

Code Distance (bits)

Misidentification

< 2 ppm
Identification and Noise

Identification in presence of environmental and measurement noise
Identification and Noise

Observe 6% intra-chip variation after +25° C
Resiliency to Noise

Max Tolerable Noise (%)

- Expected Errors Removed
- Unexpected Errors Injected

CRP Size

- 64-bit
- 128-bit
- 256-bit
- 512-bit
Resiliency to Noise

Enrollment Phase

- Expected Errors Removed
- Unexpected Errors Injected

Max Tolerable Noise (%) vs CRP Size

- 64-bit
- 128-bit
- 256-bit
- 512-bit
Resiliency to Noise

### Enrollment Phase

- **Expected Errors Removed**
- **Unexpected Errors Injected**

#### CRP Size

<table>
<thead>
<tr>
<th>CRP Size</th>
<th>Max Tolerable Noise (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-bit</td>
<td>12%</td>
</tr>
<tr>
<td>128-bit</td>
<td>28%</td>
</tr>
<tr>
<td>256-bit</td>
<td>42%</td>
</tr>
<tr>
<td>512-bit</td>
<td>62%</td>
</tr>
</tbody>
</table>
Resiliency to Noise

### Enrollment Phase
- **Expected Errors Removed**
- **Unexpected Errors Injected**

### Environmental Conditions

<table>
<thead>
<tr>
<th>CRP Size</th>
<th>Max Tolerable Noise (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-bit</td>
<td>12%</td>
</tr>
<tr>
<td>128-bit</td>
<td></td>
</tr>
<tr>
<td>256-bit</td>
<td></td>
</tr>
<tr>
<td>512-bit</td>
<td>62%</td>
</tr>
</tbody>
</table>
Resiliency to Noise

Environmental Conditions

Enrollment Phase

- Expected Errors Removed
- Unexpected Errors Injected

Max Tolerable Noise (%)

CRP Size

- 64-bit
- 128-bit
- 256-bit
- 512-bit

12% 14% 62% 142%
Repeatability and Performance
Repeatability and Performance

- Repeatable cache line errors
• Repeatable cache line errors
• Repeatable cache line errors
• Repeatability of cache line errors
• Repeatable cache line errors

• Linear increase in runtime as a function of self-test attempts
Model Building Attack Case Study
Model Building Attack Case Study

![Graph: Prediction Rate vs. Observed CRP]

- Prediction Rate vs. Observed CRP

- X-axis: CRP Count
- Y-axis: Prediction Rate (bits/response)

Graph shows a trend where the Prediction Rate increases as the CRP Count increases.
Model Building Attack Case Study

Prediction Rate vs. Observed CRP

Prediction Rate (bits/response)

CRP Count

0.4
0.5
0.6
0.7
0.8
0.9
1.0

50000
100000
150000
200000
250000
300000
350000
400000

Model Building Attack Case Study

Prediction Rate vs. Observed CRP

10% Noise
Model Building Attack Case Study

- Prediction Rate vs. Observed CRP

10% Noise

Regenerate logical error map
• Observe that correctable errors in caches can be used as silicon fingerprints

• Introduce a challenge-response design that can sustain large number of authentications (10 year lifetime)

• Demonstrate robustness of technique to noise (up to 142%)

• Realize a proof-of-concept to show system is practical
Thank you!

Questions?
Authenticache: Harnessing Cache ECC for System Authentication

Anys Bacha and Radu Teodorescu
Department of Computer Science and Engineering
The Ohio State University
http://arch.cse.ohio-state.edu