A Novel Microprocessor-intrinsic Physical Unclonable Function

Abhranil Maiti
Patrick Schaumont

ECE Department, Virginia Tech

FPL 2012, Oslo, Norway

08/30/2012
• What is a microprocessor-intrinsic PUF? Why do we need it?

• Detail of the microprocessor-intrinsic PUF

• Results

• Conclusion and future work
Motivation

SW IP is valuable and needs to be protected.

One way of preventing illegal use of SW IP is to restrict its execution to an authenticated processor. Additionally, in FPGAs, we need to authenticate the HW too.

How can we do it cheaply?

Idea: extracting the characteristics of a microprocessor using software programs.
Microprocessor-intrinsic PUF

- Extracts variability in a microprocessor pipeline to identify a chip.

- Accepts a software instruction as a challenge and produces the delay in a data path or a control path as the response.

- The delay is measured by over-clocking the microprocessor.
Why do we need a new PUF?

- Majority of the proposed PUFs require significant hardware resources e.g. RO-PUF requires a pair of ROs to generate a single response bit.

- There are PUFs that require no additional hardware \textit{(intrinsic)}: memory-based PUFs.


  They require power-cycling to generate the CRPs.

- The proposed PUF is intrinsic but requires no power-cycling.
Benefits of the proposed PUF

- A microprocessor is a ubiquitous circuit element, present in almost every embedded application.

- It provides an easy way of integrating low-level hardware information with the high-level software applications.

- Any post-processing of the PUF data can be flexibly done in software obviating any need of costly error-correction hardware.

• What is a microprocessor-intrinsic PUF? Why do we need it?

• **Detail of the microprocessor-intrinsic PUF**

• Results

• Conclusion and future work
Basic Idea: a pipelined delay path

Clock (frequency = f, period = $T_c$, $T_c = 1/f$)

$T_c \geq t_{c\_q} + t_{\text{path}} + t_{\text{setup}}$
Frequency Failure Points

\[ T_c \]

\[ \text{Chip}_1 \rightarrow t_{\text{path}_1} \]
\[ \text{Chip}_2 \rightarrow t_{\text{path}_2} \]
\[ \text{Chip}_3 \rightarrow t_{\text{path}_3} \]
\[ \text{Chip}_4 \rightarrow t_{\text{path}_4} \]
\[ \text{Chip}_m \rightarrow t_{\text{path}_m} \]

FFP = Frequency Failure Points
Failure transition information (FTI) = \{ f_{\text{start}}, f_{\text{end}}, pc \text{ values in the transition region} \}
CRP formation

- **Frequency**
  - $f_{\text{start}}$ = 0.1k
  - $f_{\text{end}}$ = 0.9k
- **Pass count**
  - 0.1k
  - 0.5k
  - 0.9k
  - k
- **Points**
  - pc: 95, 87, 46, 17 and 5
  - r: 11, 10, 01, 01 and 00
  - $110010100$
1. Claims identity

2. Searches the db and defines a challenge by selecting one or more instructions and a set of frequencies

3. Asks the chip to reproduce the responses

4. The chip generates the responses and sends back

5. If it matches with the stored one, the verifier authenticates the chip
• What is a microprocessor-intrinsic PUF? Why do we need it?

• Detail of the microprocessor-intrinsic PUF

• Results

• Conclusion and future work
Measurements were taken at an interval of 0.5 MHz
Add 0x7FFFFFFF, 0x1

94 response bits

No of sampling points = 1 + \((f_{\text{max}} - f_{\text{min}}) / \text{sampling steps}\)

No of response bits = No of sampling points \times 2
Mult 0xFFFFFFFF, 0xFFFFFFFF

- Chip1: $f_{\text{min}} = 128.5$ MHz, $f_{\text{max}} = 154.5$ MHz
- Chip2: $f_{\text{min}} = 131$ MHz, $f_{\text{max}} = 156.5$ MHz

Mult 0xFFFFFFFF, 0x80000001

- Chip1: $f_{\text{min}} = 131$ MHz, $f_{\text{max}} = 156.5$ MHz
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Operation</th>
<th>Uniqueness</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>0x7FFFFFFF + 1</td>
<td>38.7 %</td>
<td>97.4 %</td>
</tr>
<tr>
<td>Multiplication</td>
<td>0xFFFFFFFF × 0xFFFFFFFF</td>
<td>36 %</td>
<td>98.1 %</td>
</tr>
<tr>
<td>Division</td>
<td>0xFFFFFFFFE00000001 ÷ 0xFFFFFFFF</td>
<td>38.1 %</td>
<td>99 %</td>
</tr>
<tr>
<td></td>
<td>0x00000000000000FA0 ÷ 0x000000014</td>
<td>37.3 %</td>
<td>95.6 %</td>
</tr>
<tr>
<td>Logic</td>
<td>0xFFFFFFFF AND 0xFFFFFFFF</td>
<td>36 %</td>
<td>99 %</td>
</tr>
<tr>
<td>Control</td>
<td>BGE</td>
<td>40.6 %</td>
<td>98.3 %</td>
</tr>
</tbody>
</table>
• What is a microprocessor-intrinsic PUF? Why do we need it?

• Detail of the microprocessor-intrinsic PUF

• Results

• Conclusion and future work
Variability in a microprocessor pipeline can identify a chip.

Multiplication and division instructions showed more variability and produced responses that are based on input operands.

Uniqueness of the proposed PUF deviates from the ideal value. It needs further improvement.

Though the PUF showed high reliability at normal operating condition, it needs to be tested under varying temperature and supply voltage.
This research was supported in part through NSF grant no 0964680 and NSF grant no 0855095.

Thank you

Questions ??