Physically Unclonable Function and Its Applications

WORTICS 2018, SETS-Chennai

N. Nalla Anandakumar

SETS-Chennai

June 30th, 2018
Presentation Outline

- Introduction to Physically Unclonable Functions (PUFs)
- PUF type analysis
- Use case for PUF technology
- Efficient Implementation of FPGA-Based PUFs with Enhanced Performance
- Testing of PUF behavior
The Need for Secure Hardware

- **Problem**: Cryptographic secrets can be leaked by physical attacks

**Figure**: Non-Invasive Attacks (only exploits externally available information. e.g., timing, power, EM, etc.)

**Figure**: Invasive Attacks (depackaging the chip to get direct access. e.g., mechanical probing, FIB, etc.)

- Algorithmic countermeasures exist
- Requires physical protection mechanisms

Secure hardware required
“Classic” Security Hardware

- **Special** purpose hardware security modules (e.g., Trusted Platform Modules)
  - Often not designed to be secure against hardware attacks
  - Too complex for embedded applications (e.g., RFID and sensors)

- **General** purpose secure co-processors (e.g., physically and electronically protected devices)
  - Too complex and expensive for most commercial and embedded applications (e.g., PCs and mobile platforms)

Alternative (lightweight) solutions required
Promising: Physically Unclonable Functions (PUFs)
Physically Unclonable Functions (PUF)

- **Fingerprint Generator for Devices**
  (It is similar to a biometric thumb impression of humans)

- **Every chip** has slightly different physical characteristics due to its manufacturing process

- **PUF Concept**: to extract unique signature from the manufacturing process variation of silicon devices, using a challenge-response mechanism

- The response in a **unique signature** (random pattern of 0s and 1s) for each chip

- Mathematically **unclonable and unpredictable**
Sources of Variability

Random Dopant

Line Edge Roughness

Gate Oxide Variation


Common Assumptions

- **Unclonability**: PUF is unique due to unpredictable variations of manufacturing process.

- **Robustness**: PUF always returns similar PUF responses when queried with the same challenge.

- **Unpredictability**: Difficult to predict the output of a PUF to a randomly chosen challenge when one does not have access to the device.

- **Tamper-evidence**: Tampering with the PUF should result in physical damage that significantly changes the PUF response.
How to implement a PUF?
Example: Arbiter PUF

- Pair of **identically** designed delay lines
  - Ideally both paths have the same delay
- **Manufacturing variations** affect properties of delay lines
  - Either of the two paths will be faster
  - Depending on which path was faster, a **one bit response** is generated

- '1' if top path is faster, else '0'

![Diagram of Arbiter PUF](image)
Example: Ring Oscillator PUF

Set of identically designed ring oscillators (ROs)
- Two ROs selected by challenge
- Counters count rising edges of signal coming from RO
- Ideally, both ROs are identical and oscillate at same frequency, hence, ideally counters are equal

Manufacturing variations affect properties of ROs
- Either of the two ROs will oscillate with higher frequency
- Depending on which RO was faster, a one bit response is generated
What are PUFs used for?
Typical Applications

- **Device identification/authentication**
  - (e.g., anti-counterfeiting)

- **Binding hardware and software**
  - (e.g., IP protection)

- **Secure key-storage**

- **Building block in cryptographic and security solutions**
  - (e.g., encryption)
Known key storage options:

- E(E)PROM, Flash, ROM, etc.

Problems:

- Non-volatile memory technologies are often vulnerable to physical attacks
- Memory adds additional complexity to manufacturing
Key Storage With PUFs

- In order to protect keys against physical attacks:
  - Do not permanently store a key in non-volatile memory
  - Generate the key only when needed from a PUF in the IC
  - Delete the key
Application: PUF based Encryption

a) $m \rightarrow c$ Encryption $c \rightarrow m$ Decryption

b) $m \rightarrow c$ Public channel $c \rightarrow m$ Carry Device

c) $m \rightarrow c$ Public Key-Pair $(A \oplus B)$ $c \rightarrow m$
Counterfeit Chips:
- Counterfeit chips are typically produced by re-labeling recycled used components and selling them as new.

Problems:
- lower performance
- shorter lifetime
- damaged component
Device Authentication With PUFs

- PUFs enable the identification/authentication of devices based on their physical properties
- Is the hardware genuine?

Security Advantages:
- Protect against counterfeits without using cryptographic operations.
PUFs allow the **remote verification** of software integrity

- allows identification of software configuration and detection of malware.
Efficient implementations of FPGA based PUFs for lightweight applications
Efficient implementations of FPGA based PUFs

- **Ring oscillator-based PUF (RO-PUF):**
  - the difference in oscillator frequencies of selected pairs of ROs generate the PUF response

- **Arbiter based PUF (A-PUF):**
  - the output is composed of two identically configured delay paths which are stimulated by an activating signal.

- **RS Latch-based PUF (RS-LPUF):**
  - the response determines the final state output of selected pairs of latches by applying consecutive rising edges

- These PUFs are investigated on Xilinx Spartan-6 FPGAs over temperature range of $0 - 85^\circ C$ with $\pm 5\%$ variation in the supply voltage ($1.2V$).

- The programmable delays of FPGA LUTs have been used to achieve more random process variations.
Spartan-6 FPGAs from Xilinx, composed of array of configurable logic Blocks (CLBs)

Each CLBs consist of two slice types

A slice contains four 6-input lookup-tables (LUTs) and eight flip-flops (FFs).
The LUT is programmed to implement an inverter whose LUT output (O) is always an inversion of its first input (A₁).

The other inputs A₂ and A₃ of LUT act as don’t-care bits but their values affect the signal proposition path from A₁ to the output (O).

The signal propagation path from A₁ to the output (O) is shortest for A₂A₃ = 00 and longest for A₂A₃ = 11 for 3-input LUTs.

Fine and Coarse PDLs are used.
Multiple Evaluations (ME)

- Multiple Evaluations done 1-1000 times
  - Final bit value based on majority voting
- Errors reduce, but at the cost of increased evaluation time
The proposed RO-PUF: ROs configuration

- Each Ring oscillator (RO) is realized using 3 inverters and 1 AND gate.

- Two ROs are implemented inside a single CLB, where each RO is implemented inside a single slice of CLB.

- In this design, 32 ROs are configured at the center of a chip in a 4 x 4 matrix of CLBs as shown in red.
FPGA device starts with initiation of 8-bit master challenge and generates the 256 subsequent challenges (8-bit Galois LFSR).

Then from each of these sub-challenges two different ROs are chosen for comparison.

The frequency of the selected ROs are then obtained and fed into the 32-bit counters.

The comparison of counter 1 and counter 2 values generates a response bit 0 or 1.

Subsequently, the generated response is stored in a 256-bit shift register.

Therefore, 256 response bits are generated for each 8-bit master challenge.
The proposed RS-LPUF: RS Latch

- Two NAND gates and one FF (for each SR-latch) are implemented in a single slice on one CLB.
- RS latch output is 1 when input is 0 in a stable state.
- When input of the RS latch changes from 0 to 1 (i.e., rising edge), the RS latch temporarily enters a metastable state. It then enters a stable state with either output 0 or 1.
- In this design, 32 RS latches are configured at the center of a chip in a 4x2 matrix of CLBs.
The proposed RS-LPUF

- This design utilizes the proposed ROPUF where PUF response is derived from the difference in counted 1s between the selected pairs of PUF instances from 32 PUF instances by applying rising clock edges.
The proposed A-PUF

- A-PUF is composed of two identically configured delay paths which are stimulated by an activating signal.
- The 32 PUF instances (PUFI) are implemented in 32 X 3 matrix of FPGA CLBs.
- The PUF response is derived from the difference in counting the numbers of 1s of the selected pairs of PUFI by applying a consecutive rising edges.

---

N. Nalla Anandakumar (SETS-Chennai)
### Comparison of FPGA Implementation Results

<table>
<thead>
<tr>
<th>PUF Design</th>
<th>Area (Total Slices)</th>
<th>Processing time (msec)</th>
<th>Response bit length</th>
<th>Target FPGA Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO-PUF</td>
<td>952</td>
<td>4.59</td>
<td>49</td>
<td>Spartan-6 (XC6SLX45)</td>
</tr>
<tr>
<td>RS-LPUF</td>
<td>324 + 1 BRAM</td>
<td>5120</td>
<td>256</td>
<td>Spartan-6 (XC6SLX45)</td>
</tr>
<tr>
<td>A-PUF</td>
<td>4,288</td>
<td>0.64</td>
<td>64</td>
<td>Spartan-6 (XC6SLX45)</td>
</tr>
<tr>
<td>Proposed RO-PUF</td>
<td>101</td>
<td>21.8</td>
<td>256</td>
<td>Spartan-6 (XC6SLX45)</td>
</tr>
<tr>
<td>Proposed RS-LPUF</td>
<td>73</td>
<td>11.2</td>
<td>256</td>
<td>Spartan-6 (XC6SLX45)</td>
</tr>
<tr>
<td>Proposed A-PUF</td>
<td>192</td>
<td>21.8</td>
<td>256</td>
<td>Spartan-6 (XC6SLX45)</td>
</tr>
</tbody>
</table>

* The total slices for the PUF with control logic (without UART).

‡ The total time taken to generate a response at 100 MHz and the total number of clock cycles requires to generate a response = total sub challenges × Ref.counter counts × delay evaluations + control logic.
Enhanced Performance: PUF Properties

In general, **uniqueness, reliability, uniformity and bit-aliasing** are the four major quality metrics (statistical properties) for PUFs.

- **Uniqueness**: each circuit has a unique signature
- **Reliability**: the PUF response is always the same, whatever the noise and environment
- **Uniformity**: determines how uniform the proportion of 0s and 1s are in the PUF response
- **Bit-aliasing**: happens when different chips may produce nearly identical PUF responses, which is an undesirable effect
Total of 10 responses from 10 FPGAs (one response per FPGA) are obtained at 1.2V and temp. of 25°C for each proposed PUFs.

The mean ($\mu$) of the proposed RO-PUF, A-PUF, and RS-LPUF implementations using the coarse PDLs are 49.20%, 49.53% and 48.33%.
Performance Analysis: Reliability

- Spartan-6 FPGA board is placed in temperature controlled chamber
- 0°C to 85°C
- 1.14V to 1.26V (1.2V nominal)

Figure: experimental platform

Figure: Temperature Test

Figure: Voltage Variation Test
## Performance Comparisons with Previous PUFs

<table>
<thead>
<tr>
<th>Design</th>
<th>Unique Value</th>
<th>Reliability</th>
<th>Uniformity</th>
<th>Bit-alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal Value</td>
<td>50%</td>
<td>100%</td>
<td>50%</td>
<td>50%</td>
</tr>
</tbody>
</table>

### RO-PUF

<table>
<thead>
<tr>
<th>Design</th>
<th>Unique Value</th>
<th>Reliability</th>
<th>Uniformity</th>
<th>Bit-alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maiti et al. 2011 [5]</td>
<td>47.24</td>
<td>99.14</td>
<td>50.56</td>
<td>50.56</td>
</tr>
<tr>
<td>Schaumont et al. 2011 [6]</td>
<td>43.50</td>
<td>94.00</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Habib et al. [1] 2013</td>
<td>48.30</td>
<td>97.88</td>
<td>50.13</td>
<td>51.80</td>
</tr>
</tbody>
</table>

### Proposed RO-PUF

<table>
<thead>
<tr>
<th>Design</th>
<th>Unique Value</th>
<th>Reliability</th>
<th>Uniformity</th>
<th>Bit-alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fine PDL</td>
<td>49.02</td>
<td>99.03</td>
<td>49.30</td>
<td>50.63</td>
</tr>
<tr>
<td>Coarse PDL</td>
<td>49.20</td>
<td>99.23</td>
<td>49.60</td>
<td>50.11</td>
</tr>
</tbody>
</table>

### RS-LPUF

<table>
<thead>
<tr>
<th>Design</th>
<th>Unique Value</th>
<th>Reliability</th>
<th>Uniformity</th>
<th>Bit-alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yamamoto et al. 2013 [8]</td>
<td>49.00</td>
<td>99.14</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Bilal et al. 2015 [2]</td>
<td>49.24</td>
<td>98.87</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

### Proposed RS-LPUF

<table>
<thead>
<tr>
<th>Design</th>
<th>Unique Value</th>
<th>Reliability</th>
<th>Uniformity</th>
<th>Bit-alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fine PDL</td>
<td>49.16</td>
<td>99.27</td>
<td>49.14</td>
<td>49.16</td>
</tr>
<tr>
<td>Coarse PDL</td>
<td>49.53</td>
<td>99.15</td>
<td>50.32</td>
<td>49.41</td>
</tr>
</tbody>
</table>

### A-PUF

<table>
<thead>
<tr>
<th>Design</th>
<th>Unique Value</th>
<th>Reliability</th>
<th>Uniformity</th>
<th>Bit-alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPSahoo et al. 2015 [7]</td>
<td>45.25</td>
<td>95.93</td>
<td>48.10</td>
<td>—</td>
</tr>
</tbody>
</table>

### Proposed A-PUF

<table>
<thead>
<tr>
<th>Design</th>
<th>Unique Value</th>
<th>Reliability</th>
<th>Uniformity</th>
<th>Bit-alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fine PDL</td>
<td>48.04</td>
<td>99.17</td>
<td>51.02</td>
<td>52.30</td>
</tr>
<tr>
<td>Coarse PDL</td>
<td>48.33</td>
<td>99.39</td>
<td>50.70</td>
<td>51.05</td>
</tr>
</tbody>
</table>
Thank you!


Abhranil Maiti, Vikash Gunreddy, and Patrick Schaumont.
A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions.


Abhranil Maiti and Patrick Schaumont.
Improved Ring Oscillator PUF: An FPGA-friendly Secure Primitive.


D. P. Sahoo, R. S. Chakraborty, and D. Mukhopadhyay.
Towards Ideal Arbiter PUF Design on Xilinx FPGA: A Practitioner’s Perspective.


Variety enhancement of PUF responses using the locations of random outputting RS latches.