Physically Unclonable Function

Jean-Luc Danger
PUF Agenda

- What is a PUF?
- PUF Taxonomy
- Silicon PUF architectures
- PUF characteristics
- PUF robustness
- Conclusions, Questions
PUF Agenda

- What is a PUF?
- PUF Taxonomy
- Silicon PUF architectures
- PUF characteristics
- PUF robustness
- Conclusions, Questions
What is a PUF?

- Function returning a signature of the device
  - Physical function,
  - which exploits material randomness,
  - and is unclonable: same structure for each device

- The signature can be:
  - A unique identifier
  - A list of responses for a given set of challenges

Introduced by Lofstrom (2000), Pappu (2001),…


PUF – why?

- **Needs for security reasons:**
  - To use the signature as a Key for **cryptography**
  - To perform **authentication**
    - Use of challenge–response protocol (CRP)

- **Application examples**
  - IP protection
  - Secure boot
  - Secret Key storage
  - RFID tag
PUF – why?

PROS

• No programmation (no human intervention)
• Not clonable (no Reverse Engineering) as the structure is the same
• Can be build in standard CMOS process (silicon PUF)
Why not an OTP NVM?

CONS:

- Needs a write operation (can be forged)
- Clonable
- Attackable by Reverse Engineering
- Needs a specific process
- Or patented CMOS (Sidense, Kilopass)
PUF phases of use

1. **Enrollment**
   - To do once after manufacturing
   - In order to get a "reference PUF" and possibly a "helper" to rebuild it

2. **Measurement (or reconstruction)**
   - Carried out when the circuit is in use
   - To obtain the reference back
   - Can use a "helper" for greater reliability
Anti-Counterfeiting, licensing

- The PUF returns the ID of a device
  - An ID could be a unique number, or a list of Challenge/Response pairs
- The device ID should be member of a valid list
- An authentication protocol with the PUF is used to enable the device usage.
  - CRP Protocol
    - Challenge(server) => expected Response(PUF)
  - Used of Cipher
    - Nonce => Cipher(K_{PUF}, Nonce) OK ?
PUF Application example #2

Reverse Engineering protection

Program the encrypted bitstream:

1. PUF

Challenge

Encrypt(BS, $K_{PUF}$)

Bitstream

0111
0110
1001

Secure Module

UNTRUSTED ZONE

Dump Attack

Impossible to reverse or modify

Program Memory

TRUSTED ZONE

Then Decrypt at boot and load:

Bitstream

0111
0110
1001

Secure Module

Load

Decrypt(EncBS, $K_{PUF}$)

Program Memory
Secure Boot:

1. The PUF is accessed by the Boot loader. It gives a key to the Bootloader.
2. The Bootloader loads and verifies the integrity of the ciphered OS.
3. If OK, the Bootloader deciphers the OS with the key given by the PUF.
4. The PUF is disabled by the Bootloader to avoid access by untrusted SW.
PUF Agenda

- What is a PUF?
- PUF Taxonomy
- Silicon PUF architectures
- PUF characteristics
- PUF robustness
- Conclusions, Questions
PUF Taxonomy

Non silicon PUF
- Optical PUF
- Coating PUF
- RF DNA

Silicon PUF
- Delay PUF
- Memory PUF
- Other types
  - Glitch, Mecca,


Non Silicon PUF

- **Use of a specific process**
  - Optical PUF:
    - particles scattered in a transparent material
    - needs a light source (laser)
  - Coating PUF:
    - dielectric particles randomly distributed
    - extra coating on a silicon wafer

---


# Silicon PUF

## Principle

- To take advantage of the CMOS process dispersion at manufacturing stage

Variation in dopant concentration

Variation in Gate oxide thickness

1.9

1.6
SILICON PUF main principles

- **Delay PUF**
  - Differential delay measurement of "identical" delay lines

- **Memory PUF**
  - Memory state at Reset

- **Other Principles**
  - *Glitch PUF*: Propagation of glitches in combinationnal logic
  - *Mecca PUF*: Duration of Write pulse in SRAM
  - *TERO PUF*: Number of oscillations after Reset of RS latch
  - …
PUF Agenda

- What is a PUF?
- PUF Taxonomy
- Silicon PUF architectures
- PUF characteristics
- PUF robustness
- Conclusions, Questions
Delay PUF

- Differential Delay measurement

\[ D = \sum_{i=1}^{N} t_i \]

\[ D' = \sum_{i=1}^{N} t'_i \]

Delay PUF: Arbiter PUF

- The first silicon PUF (Gassend et al.)

- Special care needed at P/R
- Many derivatives to enhance security against modeling attack

Feed-Forward PUF

- Proposed by Gassend et al. to fight ML attacks
  - Some challenge bits are generated by intermediate arbiters

Ring Oscillator PUF

- **Introduced by Suh et al**
  - Identical Ring Oscillators are compared pairwise

Loop PUF

- Based on a single Loop
  - Delay chain with not necessarily balanced delay elements

- Huge quantity of challenges
  - very « strong » PUF
  - Sorting of sequential measurements with "equivalent" challenges
- Flexible: returns an integer (not boolean) value

Memory PUF

- **Principle**
  - Convergence towards a steady state from an unstable state:
    - Metastable
    - Power on

What happens when Reset goes from 1 to ‘0’?

---

SRAM PUF

- SRAM state at power-up

(a) SRAM cell CMOS circuit.

(b) SRAM cell logic circuit.

SRAM PUF
uniqueness and steadiness

Hardware Intrinsic Security, from Theory to Practice
Vincent van der Leest - Intrinsic-ID http://www2.lirmm.fr/journees_securite/material/j7/VanDerLeest.pdf
## Brief comparison

<table>
<thead>
<tr>
<th>Pros</th>
<th>Delay PUF</th>
<th>Memory PUF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Unique</td>
<td>• Unique</td>
</tr>
<tr>
<td></td>
<td>• Many challenges</td>
<td>• Implementable in DFF or non initialized memory</td>
</tr>
<tr>
<td></td>
<td>• Implementable in any technology</td>
<td>• Available in many circuits</td>
</tr>
<tr>
<td>Cons</td>
<td>• Sensitive to <strong>modeling attack</strong></td>
<td>• Few challenges =&gt; &quot;weak PUF&quot;</td>
</tr>
<tr>
<td></td>
<td>• Needs effort at P/R</td>
<td>• Not steady</td>
</tr>
<tr>
<td></td>
<td>• Not steady</td>
<td></td>
</tr>
</tbody>
</table>
PUF Agenda

- What is a PUF?
- PUF Taxonomy
- Silicon PUF architectures
- **PUF characteristics**
- PUF robustness
- Conclusions, Questions
Main Properties to meet

- **Uniqueness**
  - Each device has its own signature
  - Can be "intra" or "inter"

- **Steadiness (or reliability)**
  - The PUF response should not be sensitive to:
    - Noise
    - Environmental change T°C, Vdd
    - Aging

- **Randomness**
  - Good bit entropy

- **Robustness**
  - physical attack
  - mathematical attack

- **Other types**:
  - Diffuseness
    - Not the same response for two challenges
    - Relates to strong PUF only
  - Correctness
    - Faulty bit detection
Causes of PUF variability

- **Process**
  - global
  - local mismatch

- **Environment**
  - $V_{dd}$
  - $TC$

- **Noise**
- **Aging**

**GOOD for UNIQUENESS**

**BAD for STEADINESS**
PUF variability

A great variance is good for uniqueness!

Impact of T°C, Vdd, aging

Bad for steadiness!

Measurement of Response 'i' with noise
Virginia Tech Metrics (Maiti et al.)

- Randomness

\[ \text{Rand}_k = \frac{1}{L} \sum_{l=1}^{L} b_l \times 100\% . \]

- Steadiness

\[ \text{Stead}_n = \frac{1}{T} \sum_{t=1}^{T} \frac{\text{HD}(R^\text{ref}_n, R^t_n)}{L} \times 100\% . \]

- Uniqueness

\[ \text{Uniq} = \frac{2}{N(N-1)} \sum_{u=1}^{N-1} \sum_{v=u+1}^{N} \frac{\text{HD}(R_u, R_v)}{L} \times 100\% . \]

Impact of aging

Main phenomenon

• Oxide Wearout
  – Negative Bias Temperatue Instability (NBTI)
    • Vth increase, especially for PMOS transistors
    • Accelerate with temperature and high Vdd
  – Hot Carrier Injection (HCI)
    • Carriers collision with the gate oxide
    • Accelerate with high switching rate and high Vdd
  – Time dependent dielectric Breakdown (TDDB)
    • Accelerate with high switching rate and high Vdd

• Interconnect cut
  – Electromigration: High density current remove conductor atoms

Impact of aging on delay-PUF

- Study performed on R0-PUF by Maiti et al.
  - Spartan 3E, T stress: 85°C, V stress: 2v instead of 1.2V

The Impact of Aging on an FPGA-based Physical Unclonable Function by A. Maiti et al. (FPL 2011)
Steadiness Enhancement example

- Secured sketch: Use of a **witness** and Error correcting code (ECC)

Key Generation example

- **Fuzzy extraction**
  - Key extracted by means of Secure Sketch and Hash function

Impact of the temperature

Impact of the core activity

Loop PUF Key generation

- Low complexity method;
  - Use optimal challenges
  - Enlarging the measurement window
  - Removing the unreliable bits indicated in the "helper"

Low complexity: ECC is not necessary

http://www.proofs-workshop.org/slides/PROOFS2014_JeanLuc_DANGER_PUF.pdf
Key generation: optimal challenges

- **Maximum distance**
  - Key bits are obtained by using "equivalent" challenges which should have the maximal distance
  
  The challenges are equivalent if their $\text{HW}=N/2$

  - Case if $N=2$ => use two complementary challenges

\[
\text{KEY bit} = \text{sign}(T_1 - T_2) \quad \Delta_T = T_1 - T_2 = \sum_{i=1}^{N} t_{i,C1_i} - t_{i,C2_i}
\]

HD $(C,C')$ bits are involved

Key generation: optimal challenges

Key bit independance

- Each key bit use challenges which should have the maximal distance with challenges for other key bits
- Use of Constant Weight Codes $A(n,d,w)$

Table: Lower Bound of Constant Weight Codes

<table>
<thead>
<tr>
<th>$(n,w)$</th>
<th>d</th>
<th>$n/2$</th>
<th>$n/3$</th>
<th>$n/4$</th>
<th>$n/5$</th>
<th>$n/6$</th>
<th>$n/7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(12,6)</td>
<td></td>
<td>22</td>
<td>132</td>
<td>?</td>
<td>-</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td>(16,8)</td>
<td></td>
<td>30</td>
<td>-</td>
<td>1170</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(18,9)</td>
<td></td>
<td>34</td>
<td>424</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td>(20,10)</td>
<td></td>
<td>38</td>
<td>-</td>
<td>?</td>
<td>13452</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(24,12)</td>
<td></td>
<td>46</td>
<td>2576</td>
<td>15906</td>
<td>-</td>
<td>151484</td>
<td>-</td>
</tr>
<tr>
<td>(28,14)</td>
<td></td>
<td>54</td>
<td>-</td>
<td>?</td>
<td>-</td>
<td>-</td>
<td>1535756</td>
</tr>
<tr>
<td>(30,15)</td>
<td></td>
<td>58</td>
<td>19210</td>
<td>-</td>
<td>?</td>
<td>?</td>
<td>-</td>
</tr>
</tbody>
</table>

Key generation: Enlarging the time measurement

- **Reduce the noise variance**
  - Two ways:
    - Increase the measurement window
    - Repeat the measurement

- **But**
  - A time boundary exists *
    - It is due the environmental noise
    - Is enlarged with differential measurement

\[
\Delta_T = T_1 - T_2 + n(t) \\
n(t) \sim \mathcal{N}(0, \frac{s^2}{mw})
\]
Key generation: Keeping the most reliable bits

- **Use the reliability information**
  - The bits corresponding to low reliability are removed, and indicated in an helper word.
  - NO ECC is needed to correct the bits
Key generation results

- Bit Error Rate vs Measurement time and unreliable bits removal
PUF Agenda

- What is a PUF?
- PUF Taxonomy
- Silicon PUF architectures
- PUF characteristics
- PUF robustness
- Conclusions, Questions
PUF Attacks

- **No Reverse Engineering Attack**
  - The strength of PUF, but other potential attacks:

- **Brute force**
  - To save every Challenge/Response (CRP)
  - Impossible in a reasonable time

- **Replay**
  - Sniffing CRPs and play them back
  - Can be countered at protocol level (authentication)

- **Mathematical**
  - Reconstruct the PUF model: Modeling Attack
  - Applies to delay-PUF

- **Physical attack**
  - Side-channel
  - Faults
Modeling Attacks

Based on Machine Learning algorithms

- Take advantage of relationships between the challenge / response
- Methods based on machine learning to build a model by trial and error
- Set of CRP needed to train ML algorithm
- Very powerful to attack delay-based PUF

* Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. “Modeling attacks on physical unclonable functions”. In Proceedings of the 17th ACM
Results from Rühmair et al:

- Logistic Regression technique: success rate
  - Arbiters
    - 99.9% using 18K CRPs in 0.6 sec. (64 taps)
  - XOR Arbiter
    - 99% using 12K CRPs in 3 min 42 secs (4 XOR, 64 taps).
  - Lightweight Arbiters
    - 99% using 12K CRPs in 1 hour and 28 mins (4 XORs, 64 taps).
  - Feed-forward Arbiters
    - 99% using 5K CRPs in 47 mins and 7 secs (7 FF, 64 taps).
  - Ring Oscillators
    - 99% using 90K CRPs (1024 bit value).

* Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. “Modeling attacks on physical unclonable functions”. In Proceedings of the 17th ACM
ML Countermeasures

- **Hide the Challenge or Response**
  - Cipher the challenge or the response
    - Cryptography is generally present near the PUF

- **Do not use a CRP protocol for authentication**
  - For example:
    - the server sends a Nonce
    - PUF sends Cipher(K_{PUF}, Nonce) to the server
    - the server checks it is OK
PUF side-channel attack

**Direct attack**
- Observe the raw oscillating frequency
- Applies to RO PUF\(^1\) and Loop PUF

**Indirect attack**
- Attack on the Fuzzy extractor \(^2\)
  - Simple Power Analysis has been carried out on a FE with error correcting code using conditional branches.
  - Template attacks have been implemented on error correcting codes (w/o conditional branches)

---

PUF SCA countermeasures

- **Against direct Attack**
  - Use sequential measurement (case of Loop PUF)
    - An internal Random variable indicates the sequence order

- **Against indirect attack**
  - Countermeasures used for cryptographic blocs
    - Masking (applies well for ECC linear codes)
    - Differential logic
Enhanced SCA attack

- **Combination with ML**
  - Use of noise distribution of the arbiter PUF \(^1\)
  - Use unsupervised ML- techniques \(^2\)
    - SCA is performed first
    - The ML technique proposes a model for classification (like for instance the "k-means" algorithm).

PUF fault injection attack

Potential attacks

- Pulse attack (laser, EMI, …)
  - The PUF output is forced
- Harmonics attack*
  - The PUF frequency can be locked on external EM carrier injection (Fault injection attack)

Countermeasures

- Detection
  - Measure online the entropy of the PUF response

* Poucheret, F., Tobich, K., Lisarty, M., Chusseau, L., Robisson, B., & Maurine, P. (2011, September). Local and direct em injection of power into cmos integrated circuits. In Fault Diagnosis and Tolerance in Cryptography (FDTC), 2011 Workshop on (pp. 100-104). IEEE.
PUF invasive attack

- Read out SRAM PUF
  - Laser stimulation techniques exploiting the Seebeck effect
    - the off-transistor becomes to conduct under laser shot
    - Provides a current increase
  - Attack performed on AVR microcontrollers

PUF Agenda

- What is a PUF?
- PUF Taxonomy
- Silicon PUF architectures
- PUF characteristics
- PUF robustness
- Conclusions, Questions
PUF conclusions

- Recap
  - A specific signature for each IC
  - Used for lightweight authentication and key generation
  - Use two phases: enrollment + reconstruction
  - Main Characteristics
    - Steadiness (the most critical); uniqueness, randomness
    - ISO Standard under study
  - Cost should consider postprocessing to enhance the steadiness
  - Can be attacked physically and mathematically
  - Needs of protections