Physical Unclonable Functions

Zahra Ghodsi
Motivation

• We use mobile devices to perform secure applications, threat model must include **physical access**.

• Current practice is to store a secret key in nonvolatile memory.
Motivation

- About **$600 billion** worth of fake stuff getting churned out **every year**. What’s more, it grows about 30 percent annually. Electronic components and microchips are no exception.
- Which one is fake?
Can we “fingerprint” chips?

• If only there was a way to uniquely identify electronic components...

Physical Unclonable Functions!
Process variation

• Naturally occurring variation in the attributes of transistors.

• Generally, it is not desirable since it would create mismatch between desired and measured results.
Physical Unclonable Functions

- Extract secrets from a complex physical system
- Because of random process variations, no two Integrated Circuits even with the same layouts are identical

- Delay-Based Silicon PUF concept (2002) – Generate secret keys from unique delay characteristics of each processor chip
Physical Unclonable Functions

• As the name suggests, PUF performs a functional operation

• PUF is used in two phases:
  – Enrollment: CPRs generated and stored in database
  – Verification: Challenge from CRP applied to PUF, response compared with database.
What makes a ‘good’ PUF?

- Define parameters that will **quantify the performance** of a PUF.
- Comparison method should be **independent** of the underlying PUF circuit.
PUF parameters: Uniqueness

• Uniqueness represents the ability of a PUF to uniquely distinguish a particular chip among a group of chips of the same type.

• Hamming distance of the responses $R_i, R_j$ of two chips for challenge $C$, averaged over $k$ chips (inter-chip HD):

$$
\text{Uniqueness} = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100% \n$$

• Same PUF, same challenge, different chips: ideally 50%
PUF parameters: Reliability

- PUF reliability captures how efficient a PUF is in reproducing the response bits.

- **Hamming distance** of the responses $R_i, R'_i$ of chip $i$ at different operating conditions, averaged over $m$ samples (intra-chip HD):

  \[
  HD_{\text{INTRA}} = \frac{1}{m} \sum_{t=1}^{m} \frac{HD(R_i, R'_{i,t})}{n} \times 100\%
  \]

- Same PUF, same challenge, same chip, different operating conditions: ideally 0
PUF parameters: Uniformity

• Uniformity of a PUF estimates how uniform the proportion of ‘0’s and ‘1’s is in the response bits of a PUF.

• Percentage Hamming Weight of the n-bit identifier, $r_{i,l}$, the l-th binary bit of response:

$$ (\text{Uniformity})_i = \frac{1}{n} \sum_{l=1}^{n} r_{i,l} \times 100\% $$

• For truly random PUF responses, this proportion must be 50%.
PUF instantiations: Non-electronic PUFs

- Optical PUFs
- $\mu_{\text{inter}} = 49.79\%$, $\mu_{\text{intra}} = 25.25\%$
PUF instantiations: Analog electronic PUFs

- Coating PUFs
- $\mu_{\text{inter}} \approx 50\%$, $\mu_{\text{intra}} < 5\%$
PUF instantiations: Delay-based intrinsic PUFs

• Intrinsic PUFs:
  – The PUF and measurement equipment fully integrated in the device.
  – Procedures and primitives of PUF construction naturally available for the manufacturing process of the device.
PUF instantiations: Delay-based intrinsic PUFs

- Arbiter PUFs
  - Digital race condition on two paths on a chip
- $\mu_{\text{inter}} = 23\%$, $\mu_{\text{intra}} < 5\%$
PUF instantiations: Delay-based intrinsic PUFs

• Ring Oscillator PUFs
  – Frequency of oscillator determined by the exact delay.
• $\mu_{\text{inter}} = 46.15\%$, $\mu_{\text{intra}} = 0.48\%$
PUF property description

- Challenge-response functionality of PUF $\prod: X \to Y : \prod(x) = y$

- **Evaluatable**: easy evaluation of $y = \prod(x)$.

- **Unique**: $\prod(x)$ contains some information about the identity of entity.

- **Reproducible**: $y = \prod(x)$ is reproducible (up to a small error)

- **Unclonable**: hard to construct a procedure $\Gamma \neq \prod$ such that $\forall x \in X : \Gamma(x) \approx \prod(x)$

- **Unpredictable**: given a set of CRPs, it is hard to predict $y \approx \prod(x)$ for random $x$.

- **One-way**: given only $y$ and $\prod$, it is hard to find $x$ such that $\prod(x) = y$.

- **Tamper evident**: tampering with entity $\prod \to \prod'$ such that $\exists x \in X : \prod(x) \neq \prod'(x)$
PUF applications: System Identification

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001010</td>
<td>010101</td>
</tr>
<tr>
<td>1011000</td>
<td>101101</td>
</tr>
<tr>
<td>0111001</td>
<td>000110</td>
</tr>
</tbody>
</table>

Database for Device A

Is this the authentic Device A?
PUF applications: Secret key generation

• Key is generated from intrinsic randomness (manufacturing variability), no explicit key-programming step is required

• No conventional non-volatile key memory is required (preventing probing attacks)

• Might require an intermediate step since PUF responses are noisy
Attacks against PUFs

• **Modeling attacks:** Machine learning (ML) based modeling attacks, currently the most effective attack against PUFs

• **Side channel attacks:** using information out of the power profile or the electromagnetic emanation
References