Tutorial:
Introduction to Side-channel Analysis

Patrick Schaumont
• Implementation Attacks
• Source of Side-Channel Leakage
  • Time, Power & Modeling
• Simple Side-channel Analysis
  • RSA Example
• Differential Side-channel Analysis
  • Differential Side-channel Analysis on AES SW
  • Correlation Side-channel Analysis on AES HW
• Practical Aspects
Overview

- **Implementation Attacks**
- **Source of Side-Channel Leakage**
  - Time, Power & Modeling
- **Simple Side-channel Analysis**
  - RSA Example
- **Differential Side-channel Analysis**
  - Differential Side-channel Analysis on AES SW
  - Correlation Side-channel Analysis on AES HW
- **Practical Aspects**
Logical Attacks target the crypto algorithm and assume a black-box model

E.g. Collision attack on hash function

\[ \text{messages} \rightarrow \text{hash function} \rightarrow \text{digests} \rightarrow \text{attack processing} \]

E.g. known-plaintext attack on block cipher

\[ \text{plaintext} \rightarrow \text{block cipher with key} \rightarrow \text{ciphertext} \rightarrow \text{attack processing} \]

\[ \text{ciphertext} = f(\text{plaintext, key}) \]
Physical Attacks target the crypto implementation and assume a white-box (or grey-box) model

E.g. Side-channel attack on block cipher

\[ \text{Voltage} = g(\text{plaintext, key}) \]
Consider the following timing attack [Joye]

1. Generate 256 passwords: $P_n = (n, 0, 0, ..., 0)$: $n: 0 \rightarrow 255$
2. Find $i_0 = \max(T(P_n))$
3. Generate 256 passwords: $P_n = (i_0, n, 0, ..., 0)$: $n: 0 \rightarrow 255$
4. Find $i_1 = \max(T(P_n))$
5. etc..
6. $P = (i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7)$. 

Algorithm 4 Password verification.

```
Input: $\bar{P} = (\bar{P}[0], ..., \bar{P}[7])$ (and $P = (P[0], ..., P[7])$)
Output: ‘true’ or ‘false’
1: for $j = 0$ to $7$ do
2:   if ($\bar{P}[j] \neq P[j]$) then return ‘false’
3: end for
4: return ‘true’
```
Consider the following timing attack [Joye]

Algorithm 4 Password verification.

```
Input: \( \vec{P} = (P[0], \ldots, P[7]) \) (and \( P = (P[0], \ldots, P[7]) \))
Output: ‘true’ or ‘false’
1. for \( j = 0 \) to 7 do
2. if \( \vec{P}[j] \neq P[j] \) then return ‘false’
3. end for
4. return ‘true’
```

1. Generate 256 passwords: \( P_n = (n, 0, 0, \ldots, 0) \): \( n: 0 \rightarrow 255 \)
2. Find \( i_0 = \max(T(P_n)) \)
3. Generate 256 passwords: \( P_n = (i_0, n, 0, \ldots, 0) \): \( n: 0 \rightarrow 255 \)
4. Find \( i_1 = \max(T(P_n)) \)
5. etc ..
6. \( P = (i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7) \).

This needs at most \( 256 \times 8 = 2048 \) password verifications
Brute force would need up to \( 256^8 = 2^{64} \) password verifications

The 'grey-box' aspect:
To construct a better-then-brute-force attack, we exploit
the fact that password bytes are tested sequentially
Overview

• Implementation Attacks

• **Source of Side-Channel Leakage**
  • Time, Power & Modeling

• **Simple Side-channel Analysis**
  • RSA Example

• **Differential Side-channel Analysis**
  • Differential Side-channel Analysis on AES SW
  • Correlation Side-channel Analysis on AES HW

• **Practical Aspects**
Execution Time as a Side-Channel

• **Timing:**
  • Algorithms take time to execute
  • Execution time can have a dependency on data

  E.g. loop iteration in point multiplication takes longer if a bit of $k$ is 1

```plaintext
Input:  $k = (k_{t-1}, \ldots, k_1, k_0)_2$ and $P$
Output: $Q = k \cdot P$

$Q = \infty$
for $i = t-1$ downto 0
  $Q = 2Q$
  if ($k_i == 1$)
    $Q = Q + P$
return $Q$
```
Execution Time as a Side-Channel

Timing:

- Algorithms take time to execute
- Execution time can have a dependency on data
- Execution time can be *indirectly* affected by micro-architecture state (cache, BTB)

E.g. table lookup time can change depending on cache state

```c
// cache may/may not contain a[index]

b = a[index] + 1;

// cache must contain a[index]
// execution time can be dependent on index
```
Power Dissipation as a Side-channel

- **Power:**
  - Architectures need energy to work
  - Power = Energy / Time
  - Power dissipation is manifested by electrical (E) or magnetic (H) effects and can be measured

- Power attacks make slightly stronger assumptions than timing attacks: physical proximity is required for accurate measurement!
Physical Properties of Crypto HW

- **Power:**
  - Architectures need energy to work
  - Power = Energy / Time
  - Power dissipation is manifested by electrical (E) or magnetic (H) effects and can be measured

- Mainstream technology of crypto HW is CMOS

  \[ P = \alpha \cdot C \cdot V^2 \cdot f + V \cdot I_{\text{leak}} \]

  - Dynamic
  - Static
Physical Properties of Crypto HW

- **Power:**
  - Power Dissipation on a net in a chip occurs because of signal transitions

![Diagram of a circuit with a gate and a resistor labeled as \( C_L \), with a note indicating a capacitance range of 1 fF to 1 pF.]
• **Power:**
  • Power Dissipation on a net in a chip occurs because of signal transitions

\[
\text{Voltage: } 0 \rightarrow V_{dd} \\
\text{Energy: } 0 \rightarrow C.V_{dd}^2 / 2
\]
Physical Properties of Crypto HW

• Power:
  • Power Dissipation on a net in a chip occurs because of signal transitions

\[ V_{dd} \rightarrow 0 \]
\[ C.V^2_{dd} / 2 \rightarrow 0 \]
Physical Properties of Crypto HW

- **Power:**
  - Power Dissipation on a net in a chip occurs because of signal transitions
Power:

Power Dissipation on a net in a chip occurs because of signal transitions.
Physical Properties of Crypto HW

• **Power:**
  - **Power Dissipation on a net in a chip occurs because of signal transitions**

<table>
<thead>
<tr>
<th>Net Transition</th>
<th>Vdd</th>
<th>Gnd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 → 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 → 1</td>
<td>$I_{\text{peak}}$</td>
<td>0</td>
</tr>
<tr>
<td>1 → 0</td>
<td>0</td>
<td>$I_{\text{peak}}$</td>
</tr>
<tr>
<td>1 → 1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Current flow in Vdd/Gnd connection is data dependent

• **Power Dissipation can be observed by measuring current or by observing magnetic field**
Physical Properties of Crypto HW

- **Power:**
  - Power Dissipation on a net in a chip occurs because of signal transitions

<table>
<thead>
<tr>
<th>Net Transition</th>
<th>Vdd</th>
<th>Gnd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 → 0</td>
<td>(I_{\text{leak}})</td>
<td>(I_{\text{leak}})</td>
</tr>
<tr>
<td>0 → 1</td>
<td>(I_{\text{leak}}+I_{\text{sc}}+I_{\text{peak}})</td>
<td>(I_{\text{leak}}+I_{\text{sc}})</td>
</tr>
<tr>
<td>1 → 0</td>
<td>(I_{\text{leak}}+I_{\text{sc}})</td>
<td>(I_{\text{leak}}+I_{\text{sc}}+I_{\text{peak}})</td>
</tr>
<tr>
<td>1 → 1</td>
<td>(I_{\text{leak}})</td>
<td>(I_{\text{leak}})</td>
</tr>
</tbody>
</table>

- Power Dissipation also includes **short-circuit component** and **leakage component**
Physical Properties of Crypto HW

• **Power**
  • Circuits may exhibit additional complex and data-dependent effects
  • **Glitches:**
    Spurious data-dependent transition

Glitch-free design is not possible when positive and negative gates are mixed
In SCA, power models are used in the attack to test grey-box assumptions.

The power consumption of a circuit can be estimated with a power model:

- Circuit-level (time-continuous & analog)
- Logic-level (time-discrete & 0, 1)
- Behavioral (time-discrete & abstract)
Logic-level Power Model

- **Hamming Weight**: Number of 1-bits in a given word
- **Hamming Distance**: Number of bits that are *changing* from one clock cycle to the next

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Value</th>
<th>HW</th>
<th>HD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>1110</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1011</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>0001</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Logic-level Power Model

- **HD** approximates the dynamic power dissipation
  \[ HD(v_i) = HW(v_i \oplus v_{i-1}) \]
- **HD** and **HW** are inaccurate for detailed effects
  - Uniform weight corresponds to uniform net load (unrealistic)
  - Does not capture glitches, capacitive effects, etc
• Implementation Attacks
  • Classification
• Source of Side-Channel Leakage
  • Time, Power & Modeling
• Simple Side-channel Analysis
  • RSA Example
• Differential Side-channel Analysis
  • Differential Side-channel Analysis on AES SW
  • Correlation Side-channel Analysis on AES HW
In SPA*, one exploits a direct connection between a power measurement and a subpart of the secret key

\[ P(n) = C.\text{key}(n) + \text{noise} \]

- \( P(n) \) are power measurements
- \( C \) is a constant
- \( \text{key}(n) \) is part of the key
- \( \text{noise} \) assumed to be \( \ll \) \( C.\text{key}(n) \)

* The P in SPA stands for Power; the ideas of SPA apply to any form of side-channel leakage. Similarly, the P in DPA standard for Power, although the ideas of DPA are not restricted to Power-based Side-channel Leakage.
• Consider the RSA signature again
• If the power consumption of a Modular Squaring and Modular Multiplication are different, we can directly observe the bits in the key!

\[
\text{Input: } m, N, d = (d_{k-1}, \ldots, d_0)_2, \text{ and } \mu : \{0, 1\}^* \rightarrow \mathbb{Z}/N\mathbb{Z} \\
\text{Output: } S = \mu(m)^d \pmod{N} \\
1: R_0 \leftarrow 1; R_1 \leftarrow \mu(m) \\
2: \text{for } j = k - 1 \text{ downto } 0 \text{ do} \\
3: \quad R_0 \leftarrow R_0^2 \pmod{N} \\
4: \quad \text{if } (d_j = 1) \text{ then } R_0 \leftarrow R_0 \cdot R_1 \pmod{N} \\
5: \quad \text{end for} \\
6: \text{return } R_0
\]
Input: $m, N, d = (d_{k-1}, \ldots, d_0)_2$, and $\mu : \{0, 1\}^* \rightarrow \mathbb{Z}/N\mathbb{Z}$

Output: $S = \mu(m)^d \pmod{N}$

1: $R_0 \leftarrow 1; R_1 \leftarrow \mu(m)$
2: for $j = k - 1$ downto 0 do
3: \hspace{1em} $R_0 \leftarrow R_0^2 \pmod{N}$
4: \hspace{1em} if ($d_j = 1$) then $R_0 \leftarrow R_0 \cdot R_1 \pmod{N}$
5: \hspace{1em} end for
6: return $R_0$
SPA on RSA

Which are multiplications, which are squarings?

(a)

(b)

Input: $m, N, d = (d_{k-1}, \ldots, d_0)_2$, and $\mu : \{0, 1\}^* \rightarrow \mathbb{Z}/N\mathbb{Z}$

Output: $S = \mu(m)^d \pmod{N}$

1: $R_0 \leftarrow 1; R_1 \leftarrow \mu(m)$
2: for $j = k - 1$ downto 0 do
3: \hspace{1em} $R_0 \leftarrow R_0^2 \pmod{N}$
4: \hspace{1em} if $(d_j = 1)$ then $R_0 \leftarrow R_0 \cdot R_1 \pmod{N}$
5: \hspace{1em} end for
6: return $R_0$

[Joye]
SPA on RSA

(a) squarings, (b) multiplications

What are the key bits?

Input: $m, N, d = (d_{k-1}, \ldots, d_0)_2$, and $\mu: \{0, 1\}^* \rightarrow \mathbb{Z}/N\mathbb{Z}$

Output: $S = \mu(m)^d \pmod{N}$

1: $R_0 \leftarrow 1; R_1 \leftarrow \mu(m)$
2: for $j = k - 1$ downto 0 do
3:     $R_0 \leftarrow R_0^2 \pmod{N}$
4:     if $(d_j = 1)$ then $R_0 \leftarrow R_0 \cdot R_1 \pmod{N}$
5: end for
6: return $R_0$
SPA on RSA

(a) squarings, (b) multiplications

What are the key bits?

Input: $m, N, d = (d_{k-1}, \ldots, d_0)_2$, and $\mu: \{0, 1\}^* \rightarrow \mathbb{Z}/N\mathbb{Z}$

Output: $S = \mu(m)^d \pmod{N}$

1: $R_0 \leftarrow 1; R_1 \leftarrow \mu(m)$
2: for $j = k-1$ downto 0 do
3:   $R_0 \leftarrow R_0^2 \pmod{N}$
4:   if $(d_j = 1)$ then $R_0 \leftarrow R_0 \cdot R_1 \pmod{N}$
5: end for
6: return $R_0$

[Joye]
Example of Power Attack

(a) squarings, (b) multiplications

Input: \( m, N, d = (d_{k-1}, \ldots, d_0)_2 \), and \( \mu : \{0, 1\}^* \rightarrow \mathbb{Z}/N\mathbb{Z} \)

Output: \( S = \mu(m)^d \pmod{N} \)

1: \( R_0 \leftarrow 1; R_1 \leftarrow \mu(m) \)
2: for \( j = k - 1 \) downto 0 do
3: \( R_0 \leftarrow R_0^2 \pmod{N} \)
4: if \( d_j = 1 \) then \( R_0 \leftarrow R_0 \cdot R_1 \pmod{N} \)
5: end for
6: return \( R_0 \)
Overview

• Implementation Attacks
• Source of Side-Channel Leakage
  • Time, Power & Modeling
• Simple Side-channel Analysis
  • RSA Example
• Differential Side-channel Analysis
  • Differential Power Analysis on AES SW
  • Correlation Power Analysis on AES HW
• Practical Aspects
Differential Power Analysis

- SPA has no effective means of combining traces belonging to different input (plaintext)
- SPA is sensitive to noise

SPA uses one measurement
Differential Power Analysis

- DPA combines the power measurements with a **power model that depends on observable data** (in or out)
- DPA averages noise

---

DPA uses multiple measurements
Overview

- Implementation Attacks
  - Classification
- Source of Side-Channel Leakage
  - Time, Power & Modeling
- Simple Side-channel Analysis
  - RSA Example
- Differential Side-channel Analysis
  - Differential Power Analysis on AES SW
  - Correlation Power Analysis on AES HW
- Practical Aspects
DPA on AES SW

8-bit microcontroller power trace

rk[0] → addroundkey → subbytes → shiftrows → mixcolumns → addroundkey

rk[i] → addroundkey

rk[Nr] → addroundkey

[26000, 27000, 28000, 29000, 30000, 31000]

N_r - 1

subbytes  mixcolumns
shiftrows

[Kocher]
AES Encryption on a microcontroller, first round, indicating position of the first SBOX output

\[ V = \text{SBOX}(\text{plaintext}[0..7] \oplus \text{key}[0..7]) \]
DPA on AES SW

Distribution over 4000 encryptions

[Kocher]
DPA on AES SW

Distribution over 4000 encryption, when the Sbox output LSB=1 (left) when the Sbox output LSB=0 (right)

[Kocher]
Since $V = \text{SBOX}[\text{Key xor Plaintext}]$, and since Plaintext is known, there is an observable dependency between power consumption (graph) and the secret key. Differential Power Analysis exploits this dependency.
DPA Attack:

For each plaintext \( n \) measure power \( P \)
For each key guess \( K_n = \{0..255\} \)
calculate \( v = \text{Sbox}[n \text{ xor } K_n] \)
if \( \text{lsbit}(v) = 0 \)
\( \text{acc0}[K_n] = \text{acc0}[K_n] + P \)
else
\( \text{acc1}[K_n] = \text{acc1}[K_n] + P \)

For each key guess \( K_n = \{0..255\} \)
\( D[K_n] = \text{acc1}[K_n] - \text{acc0}[K_n] \)

Key byte \( K = \text{Argmax}(D[K]) \)

Distribution over 4000 encryption, when the Sbox output LSB=1 (left) when the Sbox output LSB=0 (right)
DPA Attack:

For each plaintext \( n \) measure power \( P \)

For each key guess \( K_n = \{0..255\} \)
- calculate \( v = \text{Sbox}[n \oplus K_n] \)
  - if \( \text{lsbit}(v) = 0 \)
    - \( \text{acc}_0[K_n] = \text{acc}_0[K_n] + P \)
  - else
    - \( \text{acc}_1[K_n] = \text{acc}_1[K_n] + P \)

For each key guess \( K_n = \{0..255\} \)
\[ D[K_n] = \text{acc}_0[K_n] - \text{acc}_1[K_n] \]

\( K_n = \text{Argmax}(D[K_n]) \)
DPA on AES SW

[D trace for wrong key guess]

[D trace for correct key guess]

[Kocher]
Overview

- Implementation Attacks
  - Classification
- Source of Side-Channel Leakage
  - Time, Power & Modeling
- Simple Side-channel Analysis
  - RSA Example
- Differential Side-channel Analysis
  - Differential Power Analysis on AES SW
  - Correlation Power Analysis on AES HW
- Practical Aspects
• AES Encryption
  • 128-bit key, 10 rounds
  • 1 round per cycle
• Assume we observe output and the power dissipation
• We will attack the last round to obtain the last roundkey
Our hypothesis does not need to guess 128 bit of the round key at the time.

Instead, we can focus on a single byte.

Assume round key $rk$

$$state = \text{sbox}^{-1}(\text{shiftrows}^{-1}(\text{out} + rk))$$
CPA on AES HW

SubBytes

ShiftRows

MixColumns

AddRoundKey

Hypothesis

Estimate

Observe

(last round)
The state byte estimate can be converted into a power dissipation estimate with a Power Model

- Hamming Weight of State Register
- Hamming Distance of State Register (this requires knowing/assuming the previous value of the state)
CPA on AES HW

- We repeat this for each AES encryption
  - Take the output
  - Infer the state from a roundkey guess
  - Infer power value of the state from the power model
CPA on AES HW

- Arrange all of this in a table

<table>
<thead>
<tr>
<th>Measured Power</th>
<th>Output Value</th>
<th>State Value</th>
<th>State Power</th>
<th>State Value</th>
<th>State Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>encryption 0</td>
<td>P0</td>
<td>O0</td>
<td>S00</td>
<td>P00</td>
<td>S10</td>
</tr>
<tr>
<td>encryption 1</td>
<td>P1</td>
<td>O1</td>
<td>S01</td>
<td>P01</td>
<td>S11</td>
</tr>
<tr>
<td>encryption 2</td>
<td>P2</td>
<td>O2</td>
<td>S02</td>
<td>P02</td>
<td>S12</td>
</tr>
<tr>
<td>encryption 3</td>
<td>P3</td>
<td>O3</td>
<td>S03</td>
<td>P03</td>
<td>S13</td>
</tr>
<tr>
<td>encryption 4</td>
<td>P4</td>
<td>O4</td>
<td>S04</td>
<td>P04</td>
<td>S14</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
As a final step, we now compare the measured power value to the estimated power value.

<table>
<thead>
<tr>
<th>Measured Power</th>
<th>Output Value</th>
<th>State Value</th>
<th>State Power</th>
<th>State Value</th>
<th>State Power</th>
</tr>
</thead>
</table>
| encryption 0   | P0           | O0          | S00         | P00         | S10         | P10         | ...
| encryption 1   | P1           | O1          | S01         | P01         | S11         | P11         | ...
| encryption 2   | P2           | O2          | S02         | P02         | S12         | P12         | ...
| encryption 3   | P3           | O3          | S03         | P03         | S13         | P13         | ...
| encryption 4   | P4           | O4          | S04         | P04         | S14         | P14         | ...
|                | ...          | ...         | ...         | ...         | ...         | ...         | ...

The table shows the measured power values and their corresponding output and state values for different encryptions. The correlation between the measured power and the estimated power values is highlighted by the arrows labeled "correlation."
• **Correlation r**
  - A factor between 1 and -1 that expresses how well one data series matches another
  - $r = 1$ means: perfect matching $x_i = y_i$
  - $r = -1$ means: perfectly complementarity $x_i = -y_i$
  - $r \sim 0$ means: unrelated data series

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X) \cdot Var(Y)}}$$

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \cdot \sum_{i=1}^{n} (y_i - \bar{y})^2}}$$
• **Correlation** $r$

\[
\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{\text{Var}(X) \cdot \text{Var}(Y)}}
\]

\[
r = \frac{\sum_{i=1}^{n}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \cdot \sum_{i=1}^{n}(y_i - \bar{y})^2}}
\]
The most likely round key is that one for which the estimates are best correlated to the measurements

\[ rkey = \arg\max_i (c_i) \]

<table>
<thead>
<tr>
<th>Encryption 0</th>
<th>Encryption 1</th>
<th>Encryption 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured Power</td>
<td>Output Value</td>
<td>State Value</td>
</tr>
<tr>
<td>P0</td>
<td>O0</td>
<td>S00</td>
</tr>
<tr>
<td>P1</td>
<td>O1</td>
<td>S01</td>
</tr>
<tr>
<td>P2</td>
<td>O2</td>
<td>S02</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Correlation:

- \( c_0 \) for encryption 0
- \( c_1 \) for encryption 1

\[ rkey = 0 \]

\[ rkey = 1 \]
• Correlation is an operation that masks out noise and unrelated factors
• Even if our estimate is not 100% correct, correlation will work!
• We only need to make estimates that have a better-than-50% probability
CPA involves 5 steps

1. Find an intermediate result in the algorithm that is dependent on a key estimate
2. Measure the power dissipation and collect data outputs (or inputs)

<table>
<thead>
<tr>
<th>Measured Power</th>
<th>Output Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>O0</td>
</tr>
<tr>
<td>P1</td>
<td>O1</td>
</tr>
<tr>
<td>P2</td>
<td>O2</td>
</tr>
<tr>
<td>P3</td>
<td>O3</td>
</tr>
<tr>
<td>P4</td>
<td>O4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
3. Evaluate hypothetical intermediate results

<table>
<thead>
<tr>
<th>Measured Power</th>
<th>Output Value</th>
<th>State Value</th>
<th>State Value</th>
<th>rkey = 0</th>
<th>rkey = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>O0</td>
<td>S00</td>
<td>S10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1</td>
<td>O1</td>
<td>S01</td>
<td>S11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td>O2</td>
<td>S02</td>
<td>S12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P3</td>
<td>O3</td>
<td>S03</td>
<td>S13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P4</td>
<td>O4</td>
<td>S04</td>
<td>S14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
4. **Evaluate power dissipation of the hypothetical intermediate results**

<table>
<thead>
<tr>
<th>Measured Power</th>
<th>Output Value</th>
<th>State Value</th>
<th>State Power</th>
<th>State Value</th>
<th>State Power</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>O0</td>
<td>S00</td>
<td>P00</td>
<td>S10</td>
<td>P10</td>
<td>...</td>
</tr>
<tr>
<td>P1</td>
<td>O1</td>
<td>S01</td>
<td>P01</td>
<td>S11</td>
<td>P11</td>
<td>...</td>
</tr>
<tr>
<td>P2</td>
<td>O2</td>
<td>S02</td>
<td>P02</td>
<td>S12</td>
<td>P12</td>
<td>...</td>
</tr>
<tr>
<td>P3</td>
<td>O3</td>
<td>S03</td>
<td>P03</td>
<td>S13</td>
<td>P13</td>
<td>...</td>
</tr>
<tr>
<td>P4</td>
<td>O4</td>
<td>S04</td>
<td>P04</td>
<td>S14</td>
<td>P14</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
5. Correlate the estimated power dissipation with the measured power dissipation and select key hypotheses with highest correlation.

<table>
<thead>
<tr>
<th>Measured Power</th>
<th>Output Value</th>
<th>State Value</th>
<th>State Power</th>
<th>State Value</th>
<th>State Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>O0</td>
<td>S00</td>
<td>P00</td>
<td>S10</td>
<td>P10</td>
</tr>
<tr>
<td>P1</td>
<td>O1</td>
<td>S01</td>
<td>P01</td>
<td>S11</td>
<td>P11</td>
</tr>
<tr>
<td>P2</td>
<td>O2</td>
<td>S02</td>
<td>P02</td>
<td>S12</td>
<td>P12</td>
</tr>
<tr>
<td>P3</td>
<td>O3</td>
<td>S03</td>
<td>P03</td>
<td>S13</td>
<td>P13</td>
</tr>
<tr>
<td>P4</td>
<td>O4</td>
<td>S04</td>
<td>P04</td>
<td>S14</td>
<td>P14</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Summary

- Implementation Attacks
  - Classification
- Source of Side-Channel Leakage
  - Time, Power & Modeling
- Simple Side-channel Analysis
  - RSA Example
- Differential Side-channel Analysis
  - Differential Side-channel Analysis on AES SW
  - Correlation Side-channel Analysis on AES HW
- Practical Aspects
The success of a side-channel attack is limited by the signal to noise ratio.

$$\text{SNR}_{\text{Side-channel}} = \frac{\text{Var(Side-channel signal)}}{\text{Var(Noise)}}$$

Switching Noise
(Noise caused by other circuits)

Electrical Noise
(Noise caused by physical phenomena)
Example: Factors impacting DPA

Assume we implement a DPA on 4 SBoxes using P and the upper byte of the word Out.

\[ V[0:7] = Sbox^{-1}(Out, Key[0:7]) \]

We will use \( HW(V[0:7]) \) for DPA: correlate(HW(V[0:7], P))

Note that the entire system includes 4 SBox.

**Will this work to find Key[0:7]?**

**Will it be any different than a system with only a single SBox?**
Example: Factors impacting DPA

Yes, we can find Key[7:0] based on HW(V[0:7]) and P.

However, P contains unrelated operations from three other SBoxes. These operations degrade the SNR for the side-channel attack.

As a result, you will need more measurements to find the same DPA peak.

\[
\text{SNR}_{\text{Side-channel}} = \frac{\text{Var}(\text{Side-channel signal})}{\text{Var}(\text{Noise})}
\]
Example: Factors impacting DPA

Can you use the same set of measurements $P$ to find other key bytes? If so, which power model would you use to find the third key byte?
Example: Factors impacting DPA

Yes. P does contain side-channel information of ALL FOUR key bytes.

Thus, you can correlate the measurements with different power models, and each power model will lead to a different key byte.

To find the third key byte, you need to use estimate $V[16:23]$, and build a power model for that byte.

For example $HW(V[16:23])$ works fine.
What value should be used for the power $P$?
• OUT = 1 value per encryption
• P must be reduced to one value per encryption as well

What power value should we use?
• OUT = 1 value per encryption
• P must be reduced to one value per encryption as well

In order to use the round key of the last round, derive the power value P from this value
In a lab setup, a trigger can provide ‘perfect’ alignment.

On an actual target, identifying a trigger may be a challenge.

(De-)Synchronization can even be used as a countermeasure.
References

- Power Analysis Book