Side Channel Attacks and Countermeasures for Embedded Systems

Job de Haas

Black Hat USA August 2, 2007
Agenda

• **Advances in Embedded Systems Security**
  – From USB stick to game console
  – Current attacks
  – Cryptographic devices

• **Side Channels explained**
  – Principles
  – Listening to your hardware
  – Types of analysis

• **Attacks and Countermeasures**
  – Breaking a key
  – Countermeasures theory
  – Practical implementations
Security in embedded systems
Trends in embedded hardware security

• Preventing debug access
  – Fuses, Secure access control

• Protecting buses and memory components
  – Flash memories with security, DRAM bus scrambling

• Increase in code integrity
  – Boot loader ROM in CPU, Public key signature checking

• Objectives:
  – Prevent running unauthorized code
  – Prevent access to confidential information
  ➢ Effective against most “conventional” attacks
Popular ‘hardware’ attacks
Attacks on glue and BGA

- Cheap BGA reballing in phone unlocking and repair
- Glue can be removed with chemicals or hot air

(See also Joe Grand’s BH presentations on hardware attacks)
Towards cryptographic devices

- **Smart cards** represent the ultimate cryptographic device:
  - Operate in a hostile environment
  - Perform cryptographic operations on data
  - Harnessing both the cryptographic operation and the key
  - Tamper resistant

- General purpose processors are **incorporating** more and more smart card style security

- Why **not use** a smart card?
  - Also adds complexity
  - How to communicate securely with it?
  - Some do (PayTV, TPM etc)
Agenda

• Advances in Embedded Systems Security
  – From USB stick to game console
  – Current attacks
  – Cryptographic devices

• Side Channels explained
  – Principles
  – Listening to your hardware
  – Types of analysis

• Attacks and Countermeasures
  – Breaking a key
  – Countermeasures theory
  – Practical implementations
Side Channel Analysis

• What?
  – read ‘hidden’ signals

• Why?
  – retrieve secrets

• How?
  – Attack channels
  – Methods
  – Tools
Attack Channels

- Time
- Power consumption
- Electro-Magnetic radiation
- Light emission
- Sound
Passive versus active attacks

- **Passive** attacks
  - Only observing the target
  - Possibly modifying it to execute a specific behavior to observe
  - **Examples**: time, power or EM measurements

- **Active** attacks
  - Manipulating the target or its environment outside of its normal behavior
  - Uncovering cryptographic keys through ‘fault injection’
  - Changing program flow (eg. circumvent code integrity checks)
  - **Examples**: Voltage or clock glitching, laser pulse attacks
Principle of timing analysis

![Diagram showing the principle of timing analysis with processes and timing values.]
Principle of power analysis

- Semiconductors use current while **switching**
- **Shape** of power consumption profile reveals activity
- **Comparison** of profiles reveals processes and data
- Power is consumed when switching from 1→0 or 0→1

Power consumption during clock cycle
Principle of electromagnetic analysis

- Electric and Magnetic field are related to current
- Probe is a coil for magnetic field
- Generally the near field (distance \(<\lambda\)) is most suitable
- Adds dimension position compared to the one dimensional power measurement
Side channel analysis tools

- Probes
  - Power: Intercept power circuitry with small resistor
  - EM: Coil with low noise amplifier
- Digital storage oscilloscope
- High bandwidth amplifier
- Computer with analysis and control software
Test equipment

- CPU: Ti OMAP 5912 150Mhz
Listening to your hardware - demo

- Oscilloscope
- Embedded system
- Sensor
- Amplifier
- EM probe
- CPU
- I/O
- Analysis Software
- Trigger
- Analog signal
- Digitized signal
Simple Power/EM Analysis

- Recover information by inspection of **single or averaged** traces
- Can also be useful for **reverse engineering** algorithms and implementations
Differential Power/EM Analysis

- Recover information by inspection **difference** between traces with different (random) inputs
- Use **correlation** to retrieve information from noisy signals
Data/signal correlation

Correlation between samples and data bit 0

Correlation between samples and data bit 1

Correlation between samples and data bit 2

Correlation between samples and data bit 3

Black Hat USA 2007
Agenda

- **Advances in Embedded Systems Security**
  - From USB stick to game console
  - Current attacks
  - Cryptographic devices

- **Side Channels explained**
  - Principles
  - Listening to your hardware
  - Types of analysis

- **Attacks and Countermeasures**
  - Breaking a key
  - Countermeasures theory
  - Practical implementations
Breaking a key - demo

- Example breaking a DES key with a differential attack
- Starting a measurement
- Explaining DES analysis
- Showing results
DES

16 rounds

- Input and output are 64 bits
- Key K is 56 bits
- Round keys are 48 bits
- Cipher function F mixes input and round key
F- function

E permutation
32 → 48

Round key
48

S box 1
S box 2
S box 8

8 * (6 → 4)

P permutation
32 → 32
DPA on DES

- Simulate DES algorithm based on input bits and hypotheses $k$.
- Select one S-Box, and one output bit $x$. Bit $x$ depends on only 6 key bits.
- Calculate differential trace for the 64 different values of $k$.
- Incorrect guess will show noise, correct guess will show peaks.
DPA on DES results

Candidate 17 of S-box 1

Candidate 18 of S-box 1

Candidate 19 of S-box 1

Candidate 20 of S-box 1
Countermeasures

• **Decrease** leakage
  – Balance processing of values
  – Limit number of operations per key

• **Increase** noise
  – Introduce timing variations in processing
  – Use hardware means
Countermeasures concepts

- Passive Side channel attacks:
  - **Hiding:**
    Break relation between processed value and power consumption
  - **Masking / Blinding:**
    Break relation between algorithmic value and processed value

![Diagram showing the process of hiding and masking in side channel attacks]
Countermeasure examples

- **Change the crypto protocol** to use key material only for a limited amount of operations. For instance, use short lived session keys based on a hash of an initial key.

Example:

![Diagram showing key derivation]

- \( K_0 \) → Perform transaction using \( K_0 \) (transaction counter=0)
- \( K_1 = \text{SHA256}(K_0) \) 
  → Perform transaction using \( K_1 \) (transaction counter=1)
- \( K_2 = \text{SHA256}(K_1) \) 
  → Perform transaction using \( K_2 \) (transaction counter=2)
- \( K_3 = \text{SHA256}(K_2) \) 
  → Perform transaction using \( K_3 \) (transaction counter=3)
- \( K_i = \text{SHA256}(K_{i-1}) \)

Source: Kocher, P. *Design and Validation Strategies for Obtaining Assurance in Countermeasures to Power Analysis and Related Attacks*
Countermeasure examples

- Remove any execution time dependence on data and key. Do not forget cache timing and branch prediction. Also remove conditional execution that depends on the key.
- Randomly insert instructions with no effect on the algorithm. Use different instructions that are hard to recognize in a trace.

<table>
<thead>
<tr>
<th>default</th>
<th>MOV</th>
<th>XOR</th>
<th>ADD</th>
<th>INC</th>
<th>CMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>random</td>
<td>MOV</td>
<td>NOP</td>
<td>XOR</td>
<td>ADD</td>
<td>NOP</td>
</tr>
<tr>
<td>random</td>
<td>MOV</td>
<td>XOR</td>
<td>NOP</td>
<td>ADD</td>
<td>INC</td>
</tr>
</tbody>
</table>

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑
Countermeasure examples

- Shuffling: **Changing the order** of independent operations (for instance S-box calculations) per round. This reduces correlation with a factor equal to the number of shuffled operations.

- Implement a **masked** version of the cryptographic algorithm. Examples can be found in research literature for common algorithms (RSA, AES).
Countermeasure demos

- Simple analysis of unprotected trace
- Effect of randomly inserting NOP instructions
- Effect of making RSA square-multiply constant
SPA attack on RSA

Signal processing to high-light dips

Variation of interval between dips

Key bits revealed
RSA implementations

- Algorithm for $M = c^d$, with $d_i$ is exponent bits ($0 \leq i \leq t$)
  - $M := 1$
  - For $i$ from $t$ down to 0 do:
    - $M := M \times M$
    - If $d_i = 1$, then $M := M \times C$

- Algorithm for $M = c^d$, with $d_i$ group of exponent bits ($0 \leq i \leq t$)
  - Precompute multipliers $C^i$
  - $M := 1$
  - For $i$ from $t$ down to 0 do:
    - For $j = 1$ to $\text{groupSize}$: $M := M \times M$
    - $M := M \times C^i$
Example: RSA message blinding

- Normal encryption: \( M = C^d \mod n \) under condition:
  - \( n = p \cdot q \)
  - \( e \cdot d = 1 \mod \text{lcm}(p-1, q-1) \)
- Choose a random \( r \), then \( C_r = C^{re} \mod n \)
- Perform RSA: \( M_r = C_r^d \mod n = C^{dr} \mod n \)
- \( M = M_r r^{-1} \mod n \)

- During the RSA operation itself the operations with exponent \( d \) do not depend on \( C \)
Test and verification

- The best way to understand side channel leakage is to measure your own implementation.
- Side channels analysis can be performed on a device to assess its level of vulnerability to such attacks.
- Such analysis is part of certification processes in the payment industry and in Common Criteria evaluations.
- FIPS 140-3 will require side channel testing for certain levels.
DPA attacks were first published by Paul Kocher et al. from Cryptography Research, Inc. (CRI).

A large range of countermeasures are patented by CRI and other companies.

CRI licenses the use of them.

The patents give a good idea of possible countermeasures; check with CRI.
Conclusions

• With the increase of security features in embedded devices the importance of side channel attacks will also increase.

• Most of these devices with advanced security features do not yet contain hardware countermeasures against side channel attacks.

• Side channel attacks present a serious threat with wide range of possibilities and a large impact.

• Still, software developers can reduce the risks of side channel attacks by securing their implementations with software countermeasures.
Job de Haas

dehaas@riscure.com
References

2. Josh Jaffe, “Differential Power Analysis”, Summer School on Cryptographic Hardware
3. S. Mangard, E. Oswald, T. Popp, “Power Analysis Attacks - Revealing the Secrets of Smartcards”
   http://www.dpabook.org/
4. Dan J. Bernstein, "Cache-timing attacks on AES",
5. D. Brumley, D. Boneh, “Remote Timing Attacks are Practical”
7. E. Oswald, K. Schramm, “An Efficient Masking Scheme for AES Software Implementations”
8. Cryptography Research, Inc. Patents and Licensing
   http://www.cryptography.com/technology/dpa/licensing.html