UNIVERSITÄT PASSAU

Hardware-Oriented Security

Introduction to Side-Channel Attacks

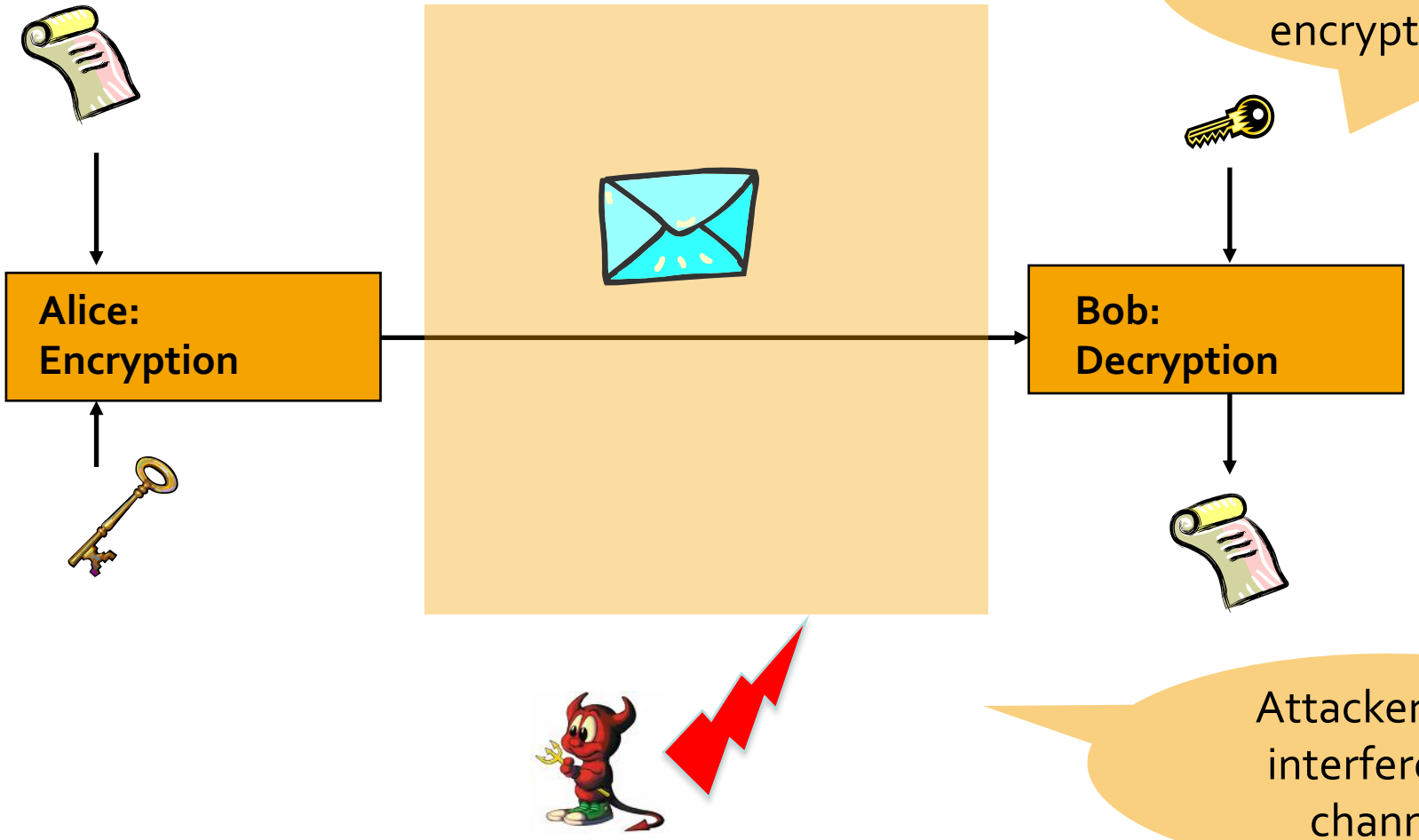
Prof. Dr. Stefan Katzenbeisser

Lehrstuhl für Technische Informatik, FIM

Outline

- **Introduction to Side-Channel Attacks against Encryption schemes**
- Timing and Power Analysis Attacks
- Countermeasures

Attacker Models in Cryptography (1)



Attacker cannot directly observe encryption/decryption

Attacker only interferes at channel

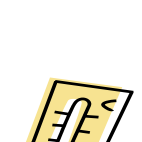
Attacker Models in Cryptography (2)

Theory

- (Interactive) Turing machines
- „private“ memory
- Attacker cannot observe computation
- Attacker can only see encrypted messages or has access to an oracle

Practice

- Physical computation has side effects:
  - Power consumption
  - Time
  - Content of caches
  - Electromagnetic emanations
  - Acoustic noise
  - …



- Abstract models do not model these side effects!

Side-Channel Attacks

Paul Kocher (1996)
Timing attacks
CRYPTO 1996

Realistic for embedded devices!

Idea:
Systematically observe side effects during the computation of cryptographic primitives

Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems

Paul C. Kocher

Cryptography Consultant
P.O. Box 8243, Stanford, CA 94309, USA.
E-mail: pck@cryptography.com.

Abstract. By carefully measuring the amount of time required to perform private key operations, attackers may be able to find fixed Diffie-Hellman exponents, factor RSA keys, and break other cryptosystems. Against a vulnerable system, the attack is computationally inexpensive and often requires only known ciphertext. Actual systems are potentially at risk, including cryptographic tokens, network-based cryptosystems, and other applications where attackers can make reasonably accurate timing measurements. Techniques for preventing the attack for RSA and Diffie-Hellman are presented. Some cryptosystems will need to be revised to protect against the attack, and new protocols and algorithms may need to incorporate measures to prevent timing attacks.

Keywords: timing attack, cryptanalysis, RSA, Diffie-Hellman, DSS.

1 Introduction

Cryptosystems often take slightly different amounts of time to process different inputs. Reasons include performance optimizations to bypass unnecessary operations, branching and conditional statements, RAM cache hits, processor instructions (such as multiplication and division) that run in non-fixed time, and

Outline

- Introduction to Side-Channel Attacks against Encryption schemes
- **Timing and Power Analysis Attacks**
- Countermeasures

Timing Attacks (1)

- Key idea: observe the timing behavior of a computation
- If timing depends on secret, this secret can be leaked!

Example: Verification of a 8-byte password

```
Input: stored password P' and new password P
Output: TRUE/FALSE
for j = 0 to 7 do
        if P'[j] != P[j] then return FALSE;
return TRUE;
```

Problem: number of loop iterations traversed depends on number of "correctly guessed" bytes of the password!

Timing Attacks (2)

```
Input: stored password P' and new password P
Output: TRUE/FALSE
for j = 0 to 7 do
        if P'[j] != P[j] then return FALSE;
return TRUE;
```

Attack requires at most 256 * 8 = 2048 calls to the verification routine, while a brute force attack would require 256^8 trials!

Attack:

- Iterate over possible values pf P[0], filling all other bytes with zeros, and measure time it takes to reject the password
- For one value, the verification time will be longer; this is the correct first byte
- Fix the correct byte and repeat process with with P[1]
- Iterate until all bits are revealed

Power Attacks

- Feasible if attacker has device with a cryptographic implementation available

- **Key idea:** record power consumption over time
- If power correlates with secrets, they can be revealed by physical measurements

- Discovery shaped field of cryptography in the 1990s!

Power Attacks against RSA

Repetition RSA:

- Select two large prime numbers p, q and set $n = pq$
- This means: $\varphi(n) = (p-1)(q-1)$
- Set of plaintexts and ciphertexts: $\mathbb{Z}_n^*$
- Select a random exponent e with $\gcd(e, \varphi(n)) = 1$
- Select an exponent d so that $ed \equiv 1 (\mathrm{mod}\, \varphi(n))$
- Public key: $pk = (n, e)$
- Private key: $sk = d$

Encryption:

$$c = E(m, pk) = m^e \bmod n$$

Decryption:

$$m = D(c, sk) = c^d \bmod n$$

Decryption operation can also be used to sign!

Efficient implementation of RSA

Efficient method to compute exponentiation: $x^y \bmod n$

Input: $x = (x_k x_{k-1} \dots x_0)_2 \quad y = (y_k y_{k-1} \dots y_0)_2$

Output: $x^y \bmod n$

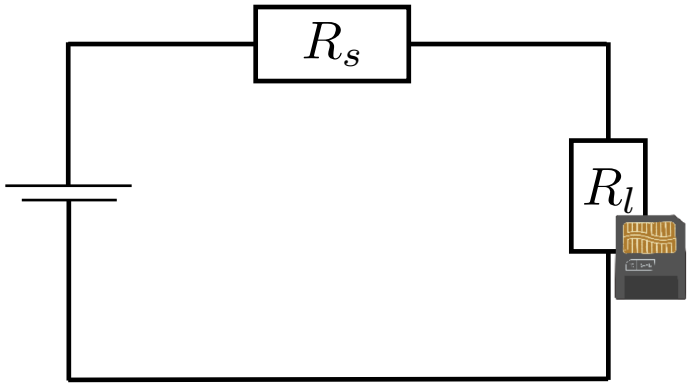
$$x^y = x^{\sum_{i=0}^{k} y_i 2^i} = \prod_{i=0}^{k} x^{y_i 2^i} = \prod_{i=0,\dots,k;y_i=1} x^{2^i}$$

```
b = 1;
for j = k downto 0 do
    b = b^2 mod n;
    if y_j == 1 then b = b * x mod n;
end for
return b
```

Number of set bits in the exponent has large influence on number of operations performed!

Side-Channel Measurement Infrastructure

$$U_s = IR_s$$



- Oscilloscope allows measurement of current over time.
- Resistor (called „shunt“) placed in series to the device to be measured.
- Computations result in a voltage drop at the resistor, which can be measured with the oscilloscope.
- Power consumption is proportional to voltage drop.

Simple Power Analysis of RSA (1)

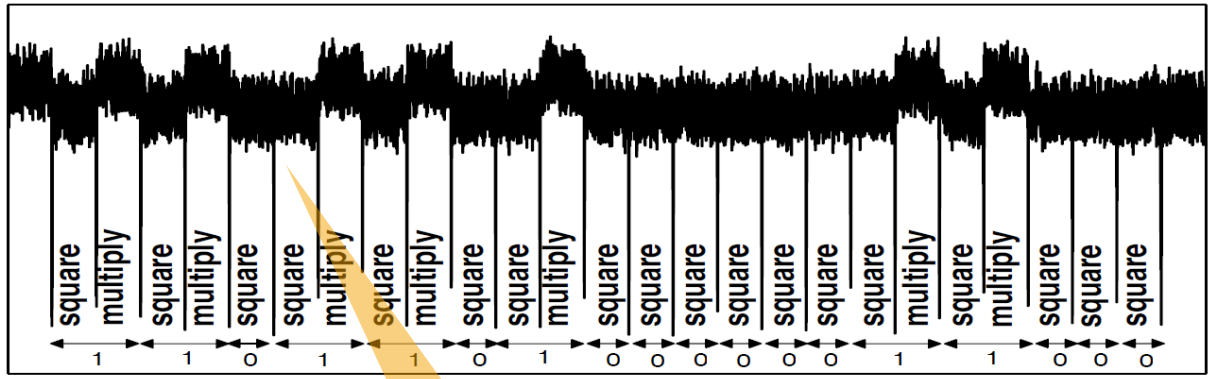
Simple Power Analysis (SPA):

- Measurement of the power consumption over time
- Different operations have different power consumption
- Key bits can be "read out" visually on the measurement

$$
\begin{algorithm}
b = 1; \\
\textbf{for } j = k \textbf{ downto } 0 \textbf{ do} \\
\quad b = b^2 \bmod n; \\
\quad \textbf{if } y_j == 1 \textbf{ then } b = b * x \bmod n; \\
\textbf{end for} \\
\textbf{return } b
\end{algorithm}
$$

Bits of the key	Operations performed by the device
1	SQ MUL
1	SQ MUL
0	SQ
1	SQ MUL
1	SQ MUL
0	SQ
…	

Simple Power Analysis of RSA (2)



Key in use is:
110110…

Bits des Schlüssels	Operationen
1	SQ MUL
1	SQ MUL
0	SQ
1	SQ MUL
1	SQ MUL
0	SQ
…	

Power Analysis: Further Applications

- Reverse Engineering:
  Identify location of „interesting“ parts during a computation

- Reverse Engineering:
  Test if a certain cryptographic algorithm is implemented in a device

- Attacks against symmetric ciphers:
  Stream ciphers, Key schedule of block ciphers, ...

Simple vs. Differential Power Analysis

UNIVERSITÄT PASSAU

Limitations / Problems of Simple Power Analysis:

- Noisy measurements make direct identification of key bits difficult
- Limits on the frequency of measurements (very fast computations)
- Visual analysis often difficult

Robust alternative: Differential Power Analysis

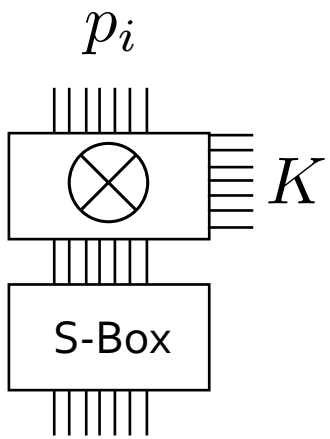
- Key idea: measurement of several power traces, which differ in *one* bit of a secret
- Use of statistical methods (hypothesis tests) to reveal bit

Differential Power Analysis (1)

Assumptions:

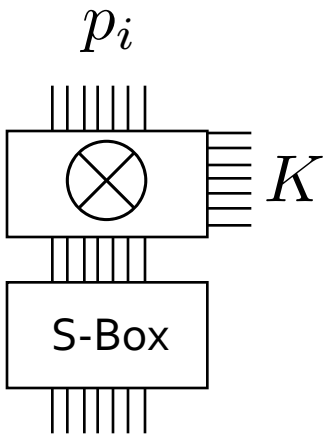
- Power consumption correlates with intermediate result of a computation
- Intermediate result depends on (correlated with) one single key bit
- Precise measurement of power trace possible

Example: Symmetric cipher (known plaintext attack), consider basic block using S-box: $S(p_i \otimes K)$



Differential Power Analysis (2)

- Assumption: least significant bit of $S(p_i \otimes K)$ creates differences within the power trace

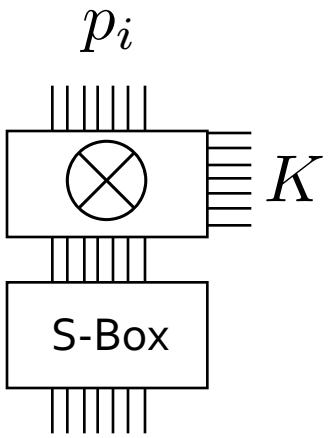


- Key byte is unknown to the attacker!
- But: he can make a guess (brute force, only 256 choices!)

- Attacker measures power trace for each known plaintext

Differential Power Analysis (3)

- Given the known plaintext, the attacker computes $S(p_i \otimes K)$ for all known plaintexts and groups power traces in two sets $\mathcal{T}_0$ and $\mathcal{T}_1$ depending on the LSB of $S(p_i \otimes K)$



- If assumptions are correct, these sets should follow different distributions!

- Compute means of all power traces in the sets $\mathcal{T}_0$ and $\mathcal{T}_1$ and consider the difference: $\langle \mathcal{T}_0 \rangle - \langle \mathcal{T}_1 \rangle$

- Difference should be large if guess of key byte was correct.

Similar to t-test in statistics!

Differential Power Analysis (4)

Complete attack:

- Divide key into bytes
- For each byte:
  - Iterate over all values of the key byte
  - Compute the difference $\langle \mathcal{T}_0 \rangle - \langle \mathcal{T}_1 \rangle$ for sets of power traces
  - Pick the byte that leads to the largest distance
- Assemble all guessed key bytes to a complete key

Linear complexity in length of key! Requires linear amount of power traces!

Outline

- Introduction to Side-Channel Attacks against Encryption schemes
- Timing and Power Analysis Attacks
- **Countermeasures**

Countermeasures against Side-Channel Attacks

UNIVERSITÄT PASSAU

Equalization:

- Remove all dependencies of control flow on secrets
- But: can be expensive, potentially “removes” optimizations

Randomization:

- Transform a particular instance of a cipher to a random instance
- Breaks the correlation between measurements and the secret

Masking:

- Mask secrets by random values and compute with „masked“ values

Side-Channel Countermeasures Equalization

- **Key idea:** remove any dependencies on the secret from leakage trace, make „computations look the same“
- Drawback: costly, often „undoes“ optimizations

```
b = 1;
for j = k downto 0 do
    b = b^2 mod n;
    if y_j == 1 then b = b * x mod n;
end for
return b
```

Example: perform “unnecessary” multiplication in case y_j=0

Side-Channel Countermeasures Randomization

- **Key idea:** randomize dependency on secret so that power trace is randomized, but still the same function is performed
- Drawback: again, some optimizations may be lost, degraded performance

Example: RSA decryption / signatures

- Use modified decryption exponent $d + k\varphi(n)$ for random *k*
- Functionally equivalent
- But: exponent becomes more complex and computations heavier

Side-Channel Countermeasures Masking

- **Key idea:** Mask intermediate values using random numbers
- For example: Boolean mask: $m \otimes r$

- Compute „with“ masked values instead of real ones
  - Mask is linear: all linear operations can be done masked
  - Problem with non-linear operation

- Requires new algorithmic design of ciphers
- Advantage: can be more efficient than other solutions

Summary

- Side channels pose a significant threat to cryptographic operations implemented in embedded devices
- Various side channels: time, power, EM emanations, etc.
- Countermeasures must be implemented in all devices that may fall into the hands of an attacker
  - Regularization of computation
  - Randomization
  - Masking