



Norwegian University of
Science and Technology

SIDE-CHANNEL ATTACKS 1: INTRO

TTM4205 – Lecture 11

Tjerand Silde

22.09.2025

Contents

Black Box Crypto

Side-Channel Attacks

Password Example

SCA Protection

Contents

Black Box Crypto

Side-Channel Attacks

Password Example

SCA Protection

Black Box Crypto

We design the security of a cryptographic scheme to follow Kerckhoff's principle: if everything about the scheme, except for the key, is known, then the scheme should be secure.

Black Box Crypto

We design the security of a cryptographic scheme to follow Kerckhoff's principle: if everything about the scheme, except for the key, is known, then the scheme should be secure.

We then analyze the scheme mathematically as black-box algorithms that take some (public or secret) input and give some (public or secret) output, and prove that it is secure concerning the algorithm description and the public data with respect to the underlying hardness assumptions.

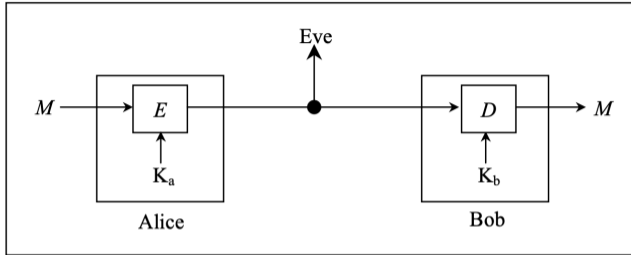


Figure 1: The traditional cryptographic model

Figure: <https://csrc.nist.gov/csrc/media/events/physical-security-testing-workshop/documents/papers/physecpaper19.pdf>

Black Box Crypto

However, security depends on your model. In practice, it matters how these algorithms are implemented and what kind of information the *physical* system leaks about the inner workings of the algorithm computing on secret data.

Black Box Crypto

However, security depends on your model. In practice, it matters how these algorithms are implemented and what kind of information the *physical* system leaks about the inner workings of the algorithm computing on secret data.

Q: What kind of information do you think might leak?

Leakage

Leakage

- ▶ The time it takes to compute

Leakage

- ▶ The time it takes to compute
- ▶ The power usage while computing

Leakage

- ▶ The time it takes to compute
- ▶ The power usage while computing
- ▶ The electromagnetic radiation...

Leakage

- ▶ The time it takes to compute
- ▶ The power usage while computing
- ▶ The electromagnetic radiation...
- ▶ The temperature increase...

Leakage

- ▶ The time it takes to compute
- ▶ The power usage while computing
- ▶ The electromagnetic radiation...
- ▶ The temperature increase...
- ▶ The memory pattern accessed...

Leakage

- ▶ The time it takes to compute
- ▶ The power usage while computing
- ▶ The electromagnetic radiation...
- ▶ The temperature increase...
- ▶ The memory pattern accessed...
- ▶ The sounds your laptop makes...

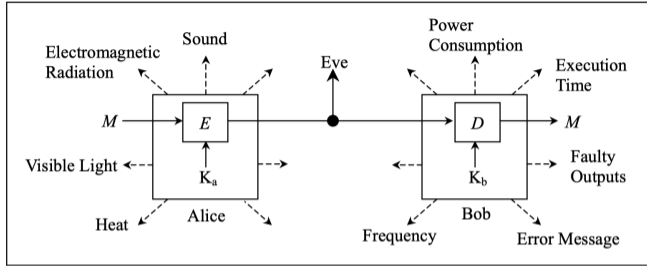


Figure 2: The cryptographic model including side-channel

Figure: <https://csrc.nist.gov/csrc/media/events/physical-security-testing-workshop/documents/papers/physecpaper19.pdf>

Researchers crack the world's toughest encryption by listening to the tiny sounds made by your computer's CPU

Security researchers have successfully broken one of the most secure encryption algorithms, 4096-bit RSA, by listening -- yes, with a microphone -- to a computer as it decrypts some encrypted data. The attack is fairly simple and can be carried out with rudimentary hardware. The repercussions for the average computer user are minimal, but if you're a secret agent, power user, or some other kind of encryption-using miscreant, you may want to reach for the Rammstein when decrypting your data.

By Sebastian Anthony December 18, 2013



Figure: <https://eprint.iacr.org/2013/857.pdf>

Examples

Q: Can you think of real-world situations where this information might leak?

Examples

Examples

- ▶ Credit cards connecting to ATMs

Examples

- ▶ Credit cards connecting to ATMs
- ▶ Applications sharing resources

Examples

- ▶ Credit cards connecting to ATMs
- ▶ Applications sharing resources
- ▶ Some publicly available crypto API

Examples

- ▶ Credit cards connecting to ATMs
- ▶ Applications sharing resources
- ▶ Some publicly available crypto API
- ▶ Malware on your phone or laptop

Examples

- ▶ Credit cards connecting to ATMs
- ▶ Applications sharing resources
- ▶ Some publicly available crypto API
- ▶ Malware on your phone or laptop
- ▶ Crypto currency hardware wallet

Examples

- ▶ Credit cards connecting to ATMs
- ▶ Applications sharing resources
- ▶ Some publicly available crypto API
- ▶ Malware on your phone or laptop
- ▶ Crypto currency hardware wallet
- ▶ Cloud key management systems

Contents

Black Box Crypto

Side-Channel Attacks

Password Example

SCA Protection

Side-Channel Attacks

Side-channel attacks (SCA) are attacks that exploits *physical leakage* in an implemented scheme to break the underlying cryptography, that is, by extracting the secret keys.

We can categorize the attacks in several different ways.

Q: Can you, based on the list of leakage, imagine how?

Side-Channel Attacks

Side-Channel Attacks

- ▶ Remote vs physical attacks

Side-Channel Attacks

- ▶ Remote vs physical attacks
- ▶ Software and hardware attacks

Side-Channel Attacks

- ▶ Remote vs physical attacks
- ▶ Software and hardware attacks
- ▶ Passive vs active attacks

Side-Channel Attacks

- ▶ Remote vs physical attacks
- ▶ Software and hardware attacks
- ▶ Passive vs active attacks
- ▶ Invasive vs non-invasive attacks

Remote vs Physical Attacks

Some side-channel attacks can be executed **remotely**, given information about how the algorithm is computed and access to timings or remotely shared sound.

For example decryption or signing queries online (remote server or WLAN) or sound through a feed (e.g. video call).

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

Paul C. Kocher

Cryptography Research, Inc.
607 Market Street, 5th Floor, San Francisco, CA 94105, USA.
E-mail: paul@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.

Figure: <https://www.rambus.com/wp-content/uploads/2015/08/TimingAttacks.pdf>



NTNU

Norwegian University of
Science and Technology

Remote Timing Attacks are Practical

David Brumley
Stanford University
dbrumley@cs.stanford.edu

Dan Boneh
Stanford University
dabo@cs.stanford.edu

Figure: <https://crypto.stanford.edu/~dabo/papers/ssl-timing.pdf>

Software vs Hardware Attacks

Some algorithms are computed in software and others in hardware, e.g., specialized circuits for computing AES or RSA.

This might impact memory allocation and SCA protection.

Passive vs Active Attacks

Some attacks are possible just by **listening** for information leakage, while other attacks requires the adversary to take a more active role, e.g., by creating **(adaptive) queries**.

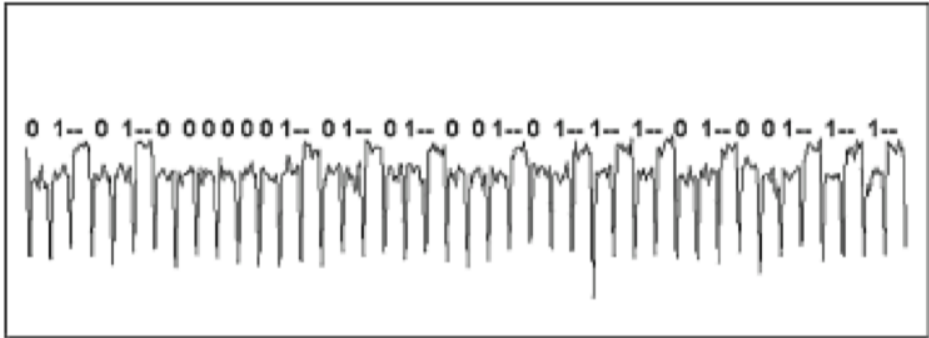


Figure: One power trace might reveal the whole key

Differential Power Analysis

Paul Kocher, Joshua Jaffe, and Benjamin Jun

Cryptography Research, Inc.
~~607 Market Street, 5th Floor~~
~~San Francisco, CA 94105, USA.~~
<http://www.cryptography.com>
~~E-mail: {paul,josh,ben}@cryptography.com.~~

Abstract. Cryptosystem designers frequently assume that secrets will be manipulated in closed, reliable computing environments. Unfortunately, actual computers and microchips leak information about the operations they process. This paper examines specific methods for analyzing power consumption measurements to find secret keys from tamper resistant devices. We also discuss approaches for building cryptosystems that can operate securely in existing hardware that leaks information.

Keywords: differential power analysis, DPA, SPA, cryptanalysis, DES

Figure: <https://paulkocher.com/doc/DifferentialPowerAnalysis.pdf>



Invasive vs Non-Invasive Attacks

An adversary that only measure time, power consumption or electromagnetic radiation is **non-invasive**.

An active adversary with physical access to the device might apply **semi-invasive** attacks using heat or lasers to interfere with the execution of programs (without destroying it).

An active adversary with physical access to the device might apply **(potentially) invasive** attacks by opening the chip to probe the circuit in the silicon itself to reveal secrets.

Contents

Black Box Crypto

Side-Channel Attacks

Password Example

SCA Protection

Checking Passwords

```
1 def isCorrectPassword(pw, user_input):
2     if len(pw) != len(user_input):
3         return False
4
5     for i in range(len(pw)):
6         if pw[i] != user_input[i]:
7             return False
8
9     return True
```

Figure: Q: What kind of information can you extract here?



Cracking Passwords

Possible vulnerabilities:

- ▶ The time depends on password length
- ▶ The time depends on correct guesses
- ▶ The attacker has unlimited trials

Protection: The time it takes to check must be independent of secrets, and we must rate-limit the number of trials.

Contents

Black Box Crypto

Side-Channel Attacks

Password Example

SCA Protection

SCA Protection

Q: Can you think of possible mitigation strategies to prevent this leakage?

SCA Protection

SCA Protection

- ▶ Constant time (secret independent) code

SCA Protection

- ▶ Constant time (secret independent) code
- ▶ Randomization of (secret) computation

SCA Protection

- ▶ Constant time (secret independent) code
- ▶ Randomization of (secret) computation
- ▶ Splitting the secret into several parts

SCA Protection

- ▶ Constant time (secret independent) code
- ▶ Randomization of (secret) computation
- ▶ Splitting the secret into several parts
- ▶ Checking that outputs are correct/valid

SCA Protection

- ▶ Constant time (secret independent) code
- ▶ Randomization of (secret) computation
- ▶ Splitting the secret into several parts
- ▶ Checking that outputs are correct/valid
- ▶ Hardware implementations of popular schemes

SCA Protection

- ▶ Constant time (secret independent) code
- ▶ Randomization of (secret) computation
- ▶ Splitting the secret into several parts
- ▶ Checking that outputs are correct/valid
- ▶ Hardware implementations of popular schemes
- ▶ Add noise to the computation (random delays)


FIPS 140-3

Security Requirements for Cryptographic Modules



Date Published: March 22, 2019

Supersedes: [FIPS 140-2 \(12/03/2002\)](#)

Planning Note (05/01/2019): 

See the [FIPS 140-3 Transition](#) project for the following information:

- [FIPS 140-3 Transition Schedule](#)
- Supporting [SP 800-140x documents](#) that modify requirements of ISO/IEC 19790:2012 and ISO/IEC 24759:2017

Author(s)

National Institute of Standards and Technology

Figure: <https://csrc.nist.gov/pubs/fips/140-3/final>

Dude, is my code constant time?

Oscar Reparaz, Josep Balasch and Ingrid Verbauwhede
KU Leuven/COSIC and imec
Leuven, Belgium

Figure: <https://eprint.iacr.org/2016/1123.pdf>

Comparative Study of ECC Libraries for Embedded Devices

Tjerand Silde

Norwegian University of Science and Technology, Trondheim, Norway
`tjerand.silde@ntnu.no`, `www.tjerandsilde.no`

Figure: <https://tjerandsilde.no/files/Comparative-Study-of-ECC-Libraries-for-Embedded-Devices.pdf>

Questions?