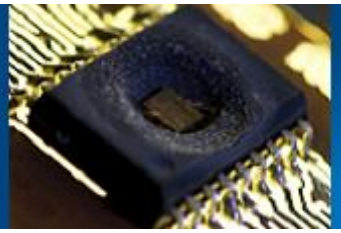


Smartcard Laboratory
Introductory Lecture

Prof. Dr.-Ing. Georg Sigl

Institute for Security in Information Technology

Lecturer: M.Sc. Oscar M. Guillen

Summer Semester 2015

Agenda

- Administrative Topics
- Why is security important?
- Introduction to Smartcards
- Objectives of the Laboratory
- Work plan
- Project Management
- Management tools for the lab

Section 1 – Organization and

ADMINISTRATIVE TOPICS

Contact

Lab Instructor

- M.Sc. Oscar M. Guillen

E-Mail

- oscar.guillen@tum.de

Consultation hours

- Feel free to contact me per email any time

Location

- N1007

Contact

Lab Tutor

- Thomas Zeschg

E-Mail

- thomas.zeschg@tum.de

Consultation hours

- Tuesday - 09:45 am -11:45 pm
- Wednesday – 1:00 pm to 3:00 pm

Location

- Laboratory Room N1003

Laboratory hours

- The Smartcard Laboratory can be performed on **your own schedule**.

<u>Dates</u>

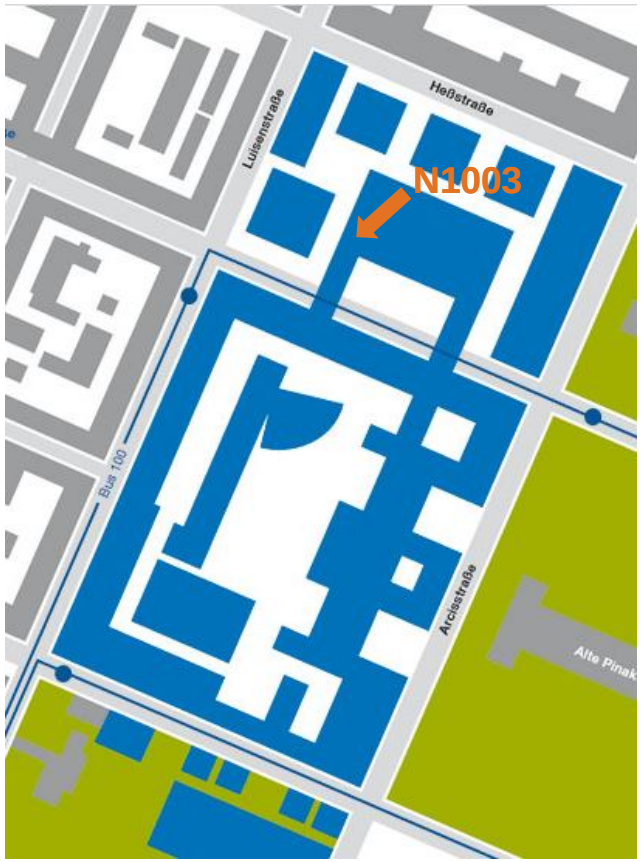
- Monday – Friday

<u>Time</u>

- 7:00 am – 9:00 pm

<u>Location</u>

- N1003 “Praktikumsraum”
- Lab-room shared with GSM-SIM



Important Dates

Lectures:

Date	Time	Place	Description
13.04.2015	11:30 am – 1:00 pm	N1005	Introductory lecture
20.04.2015	11:30 am – 1:00 pm	N1005	SCA-Introduction

Important Dates

Presentations / Exam (remember to register!):

Date	Time	Place	Description
16.06.2015	9:00 am – 12:00 pm	N1005	**Midterm** Presentation
14.07.2015	9:00 am – 12:00 pm	N1005	**Final** Presentation
24.07.2015	9:00 am – 12:00 pm	N1005	Oral **exam**

Grading

Lab work:

- Work is to be carried out in **groups**
- Each student must keep a lab protocol (template will be given)

Presentations:

- Midterm / Final Presentations

Exam:

- 10-minutes long oral examination

Teamwork

Suggestions:

- Share the work load in a fair manner
- Work together to achieve the objectives
- Contribute with ideas
- Listen to the opinions of others
- Work on your task and be open to help others
- **Keep in touch** with your team members

Section 2 – Motivation

WHY IS SECURITY IMPORTANT?

Use of Cryptography

- Historically: Military Communications
- since 1970: Industrial Data Transmission (DES)
- since 1980: Mobile Communications (GSM, Chipcards)
- since 1990: Everyday Life (WLAN, HTTPS, PGP)

Use of cryptography – Trends

The use of cryptography is becoming everyday more important

Examples:

- Smartcards
- Internet of Things
- Machine-to-Machine
- Car-to-X
- Cloud Computing
- Advance Metering Infrastructure (Smart Grid)
- Medical Monitoring / Telemetry
- Identification (Internet and in the real world)

Use of Cryptography – Everyday use

Chipcards as an example

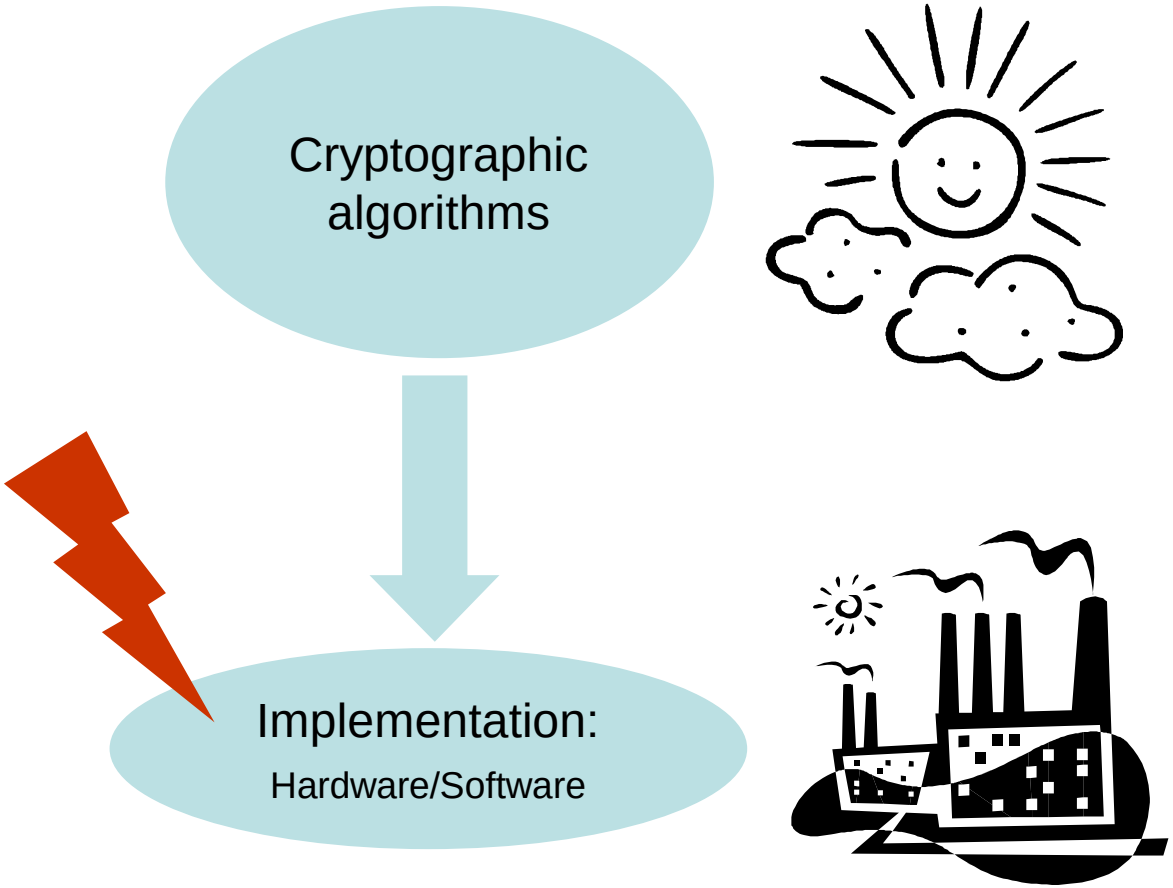
- Telecommunication
  - Telephone cards, SIM-Cards
- Payment
  - e-Purses, Credit cards
- Access control
  - Access ID, Public transportation cards
- Identification
  - Passport, Driving license, Medical cards
- Digital Rights Management
  - Pay TV

image source: http://siliconangle.com

Challenges

- Every designer will somehow be involved with the topic of security when designing a system (e.g. piracy).
- The commercial benefit for an attacker can be really high (e.g. Pay TV, product piracy), this is also true for the amount of time and money that someone can invest in order to attack a system.
- The devices that make use of cryptography are in the hands of many users (and attackers). Therefore the implementation of these algorithms must be well protected.

Cryptography in Engineering



Attacking the weakest link

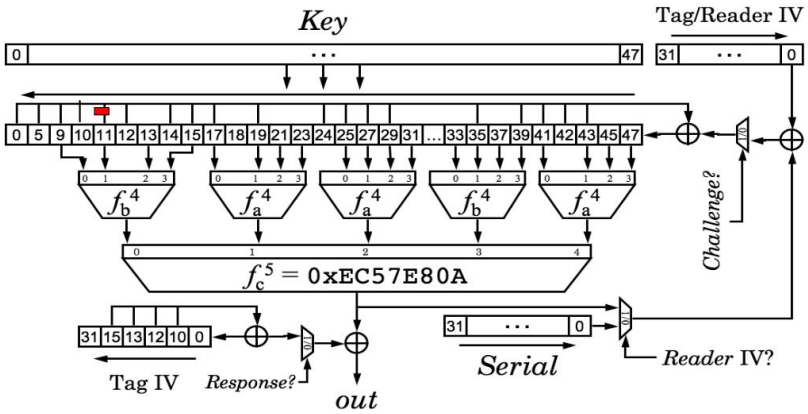
Hardware Security: Mifare Classic

Access control and ticketing systems (e.g. Oyster Card in London)

Contactless memory card, crypto in Hardware (LFSR-based)

Their security was broken in 2007 by researchers at the Humboldt-Universität Berlin and Radboud Universiteit Nijmegen by making use of:

- Gate-level Reverse Engineering
- Protocol Analysis
- Emulators



Weaknesses:

- Proprietary Cryptography (Crypto-1)
- Weak pseudo-random-numbers generator (PRNG)

Details: http://en.wikipedia.org/wiki/MIFARE

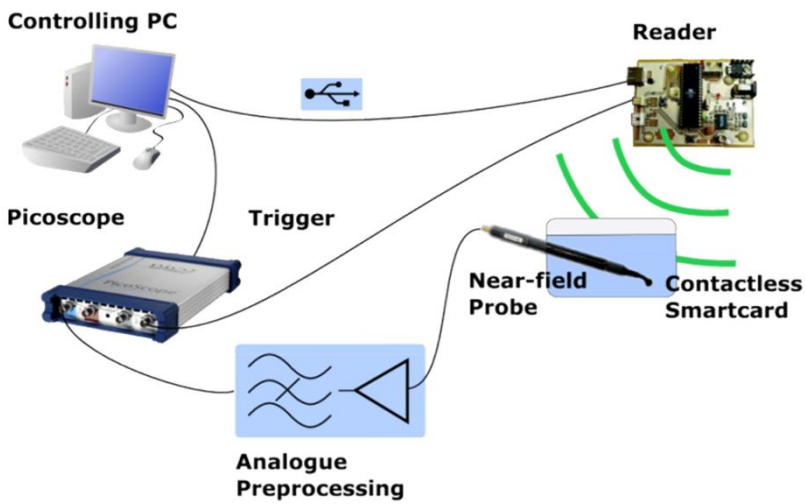
Hardware Security: Mifare DES Fire

Access control and ticketing systems (Prague, San Francisco, London,…)

Contactless memory card, (strong) crypto (3DES)

Their security was broken in 2011 by researchers at Ruhr-University Bochum:

- Home-brewed RFID reader
- Low-cost USB oscilloscope
- Near field probes



Weaknesses:

- EM Emanation (Side Channel Analysis)

Details: https://www.iacr.org/workshops/ches/ches2011/presentations/Session%205/CHES2011_Session5_1.pdf

Hardware Security: KeeLoq

“Remote Keyless Entry” Systems e.g. Car keys, Garage door openers

Algorithm implemented in Hardware (NLFSR)

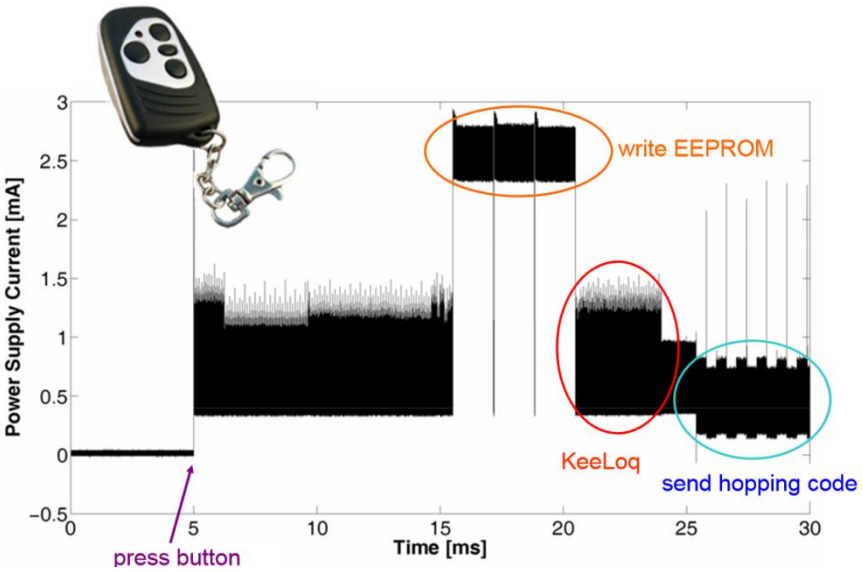
Their security was broken by researchers at the Ruhr-Universität Bochum by making use of:

- Mathematical cryptanalysis
- Side-channel attacks (DPA, SPA)

Weaknesses:

- Proprietary cryptography
  (in 2006 their algorithm was leaked in Internet)
- Susceptibility to side-channel attacks

Details: http://www.crypto.rub.de/keeloq/



Hardware Security: Locking System

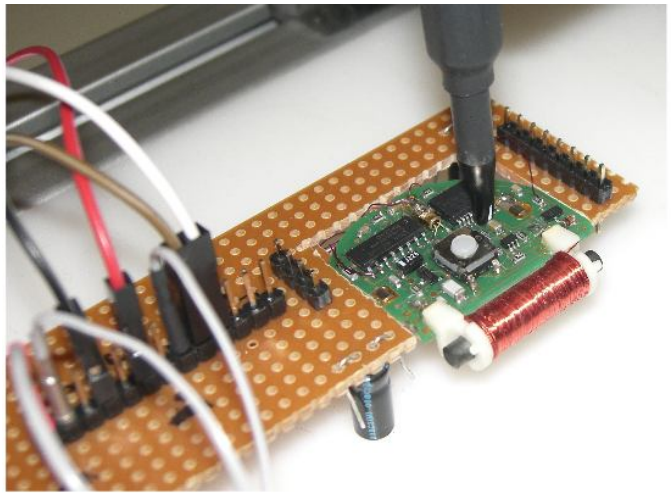
Access control

Strong Cryptography (3DES)

The security was broken by a collaboration between researchers/students from TUM, LMU, TU Darmstadt and TU Kaiserslautern

Weaknesses:

- General purpose MCU
- Weaknesses in RNG
- Susceptible to Side Channel Analysis
- Susceptible to Fault Injection attacks



Summary

Robust security is becoming more critical in everyday-use products

- Stark rise in use of embedded systems
- High number of people with access to these systems
- Valuable information stored or transmitted with them

Cryptography has evolved in the past twenty years from being a secret science practiced only by a small group of mathematicians to a fundamental discipline for engineers

Designing secure systems and **securely implementing** cryptographic algorithms are skills which engineers require more than ever

Side-note: Security is not the same as Safety
(…although in German the same word is used for both: *Sicherheit*):

- Safety: The system must not represent a hazard (to people)
- Security: The system must be resistant to attacks (to the system)

Section 3 – A brief introduction to

SMARTCARDS

Introduction to Smartcards

What is a Smartcard?

- Embedded computer (Microcontroller)
- Tamper-resistant
- Limited resources
- Embedded in a plastic card
- Low cost

Typical uses of a Smartcard

- Secure data storage
- Secure data processing
- Authentication

image source: http://crystalequityresearch.blogspot.com

Smartcard hardware components

Non-Volatile Memory
- EEPROM
- ROM

Volatile Memory
- SRAM

Crypto Functions
- Symmetric (3DES, AES,…)
- Asymmetric (e.g. RSA)

Analog components
- Voltage regulators
- Anti-tamper sensors

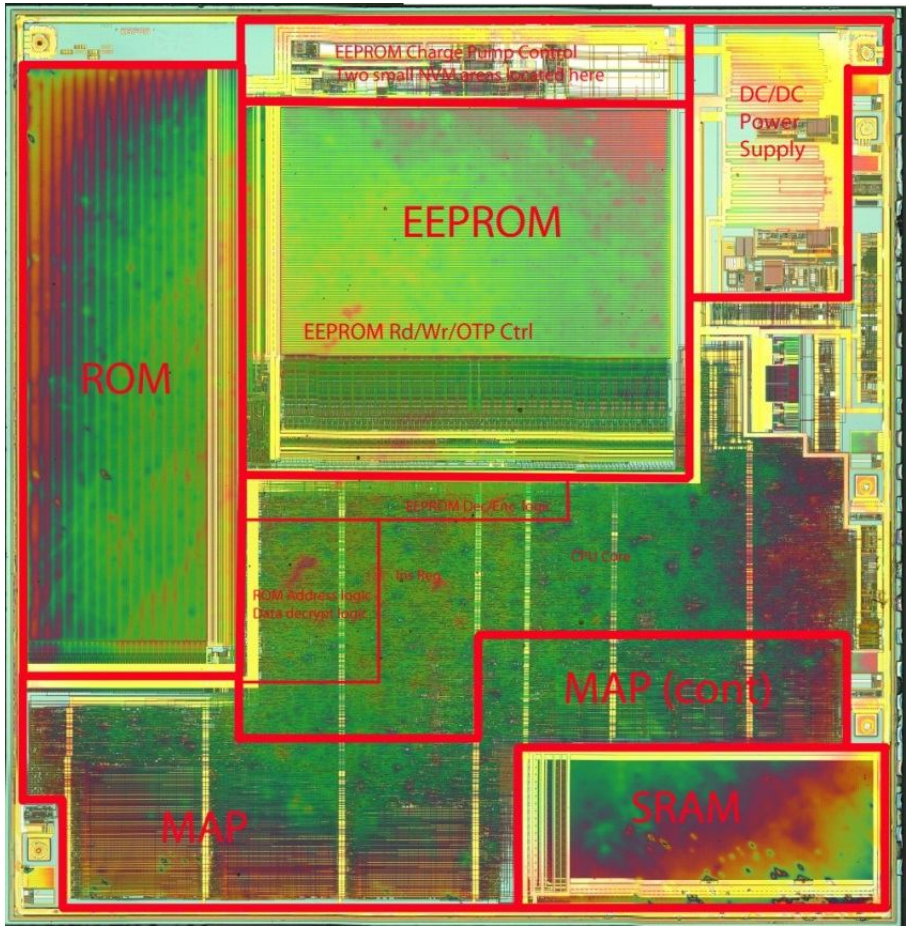
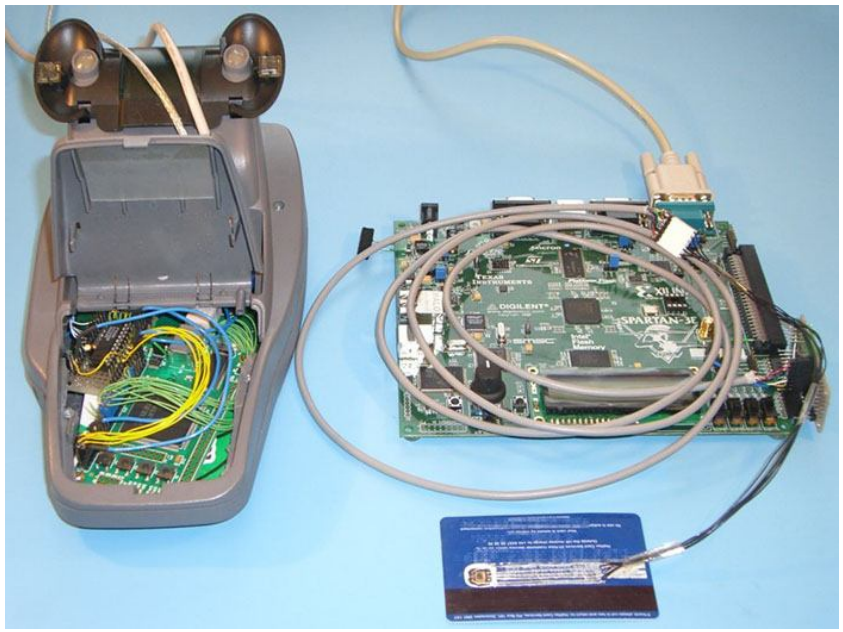


image source: http://www.flylogic.net

Attacks on Smartcards



Logical attacks

- Cryptographic Algorithms
- Cryptographic Protocols

Software attacks

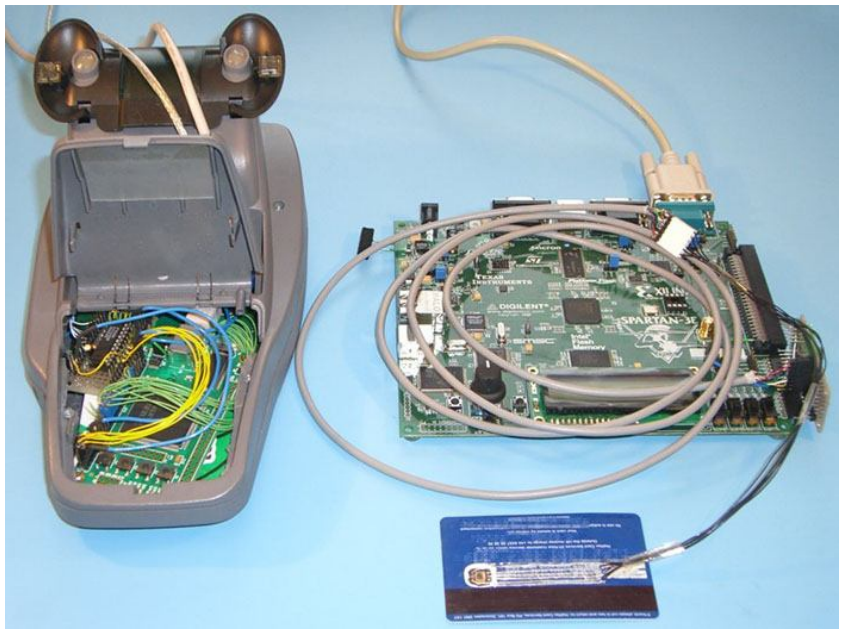
- Applications within the Smartcard
- Terminal-side applications

Hardware attacks

- Invasive
- Non-invasive
- Active
- Passive

image source: https://www.usenix.org/legacy/event/sec07/tech/drimer/drimer_html/

Attacks on Smartcards



Logical attacks

- Cryptographic Algorithms
- Cryptographic Protocols

Software attacks

- Applications within the Smartcard
- Terminal-side applications

Hardware attacks

- Invasive
- Non-invasive
- Active
- Passive

image source: https://www.usenix.org/legacy/event/sec07/tech/drimer/drimer_html/

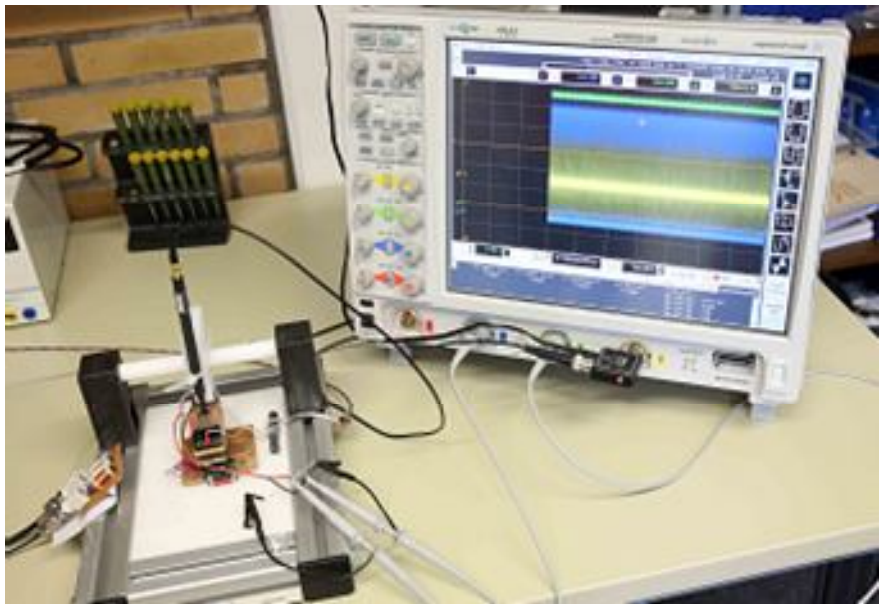
Side-channel Attacks

Advantages for an attacker

- Non-invasive
- Passive
- Relatively low-cost
- Powerful and relatively fast

Possible Side-channels

- Timing
- Power consumption
- EM emission
- others…



Section 3 – Smartcard Lab

LABORATORY OBJECTIVES

Laboratory Description

Pay TV system

- A smartcard is used to decrypt a video data stream
- The right cryptographic key needs to be present in the card for that to occur
- Students take the role of the attacker to compromise the system
- …and also the role of developer to provide a secure solution against attackers

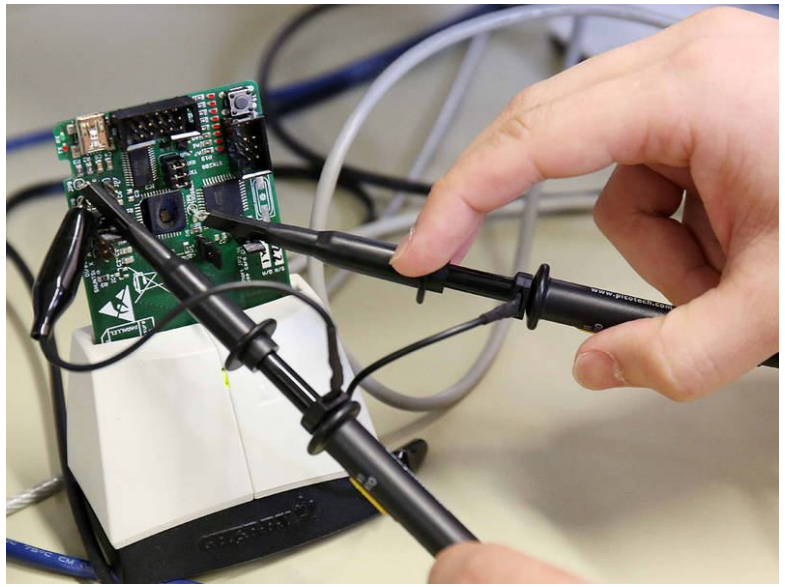
Laboratory's Objectives

Phase 1: Attack and **clone** a Smartcard

- Analyze an existing Smartcard (emulator)
- Extract the crypto key
  - Differential Power Analysis
- Create your own Smartcard OS
- Make use of the extracted key in your own card

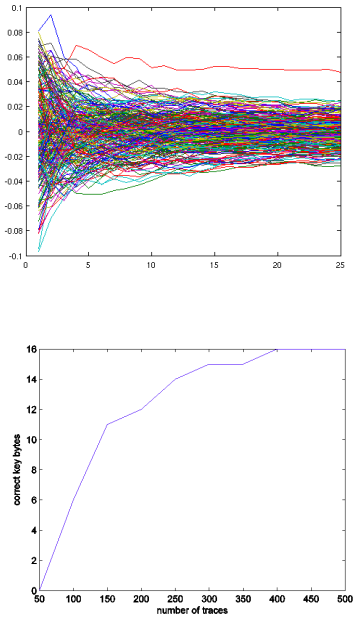
Phase 2: **Protect** your Smartcard

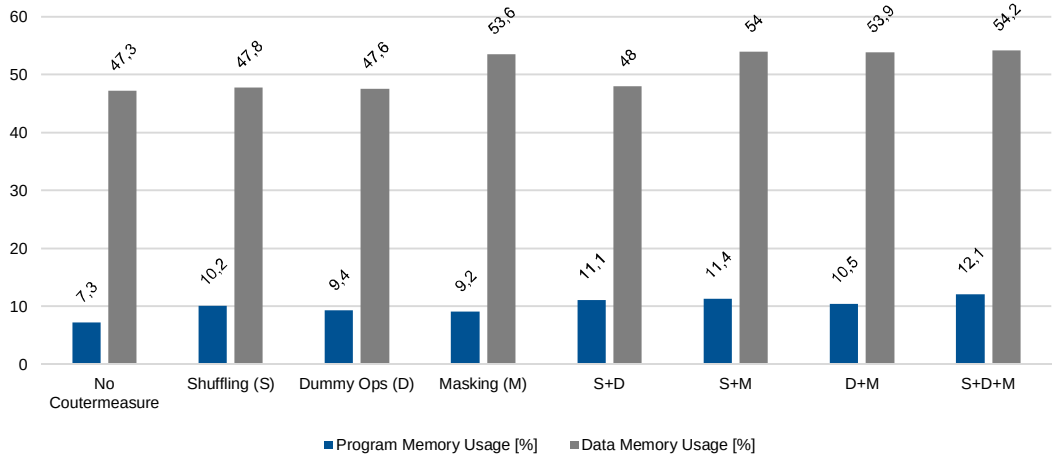
- Implement countermeasures against DPA
- Attempt to break your own countermeasures
- Evaluate the resistance of the different countermeasures



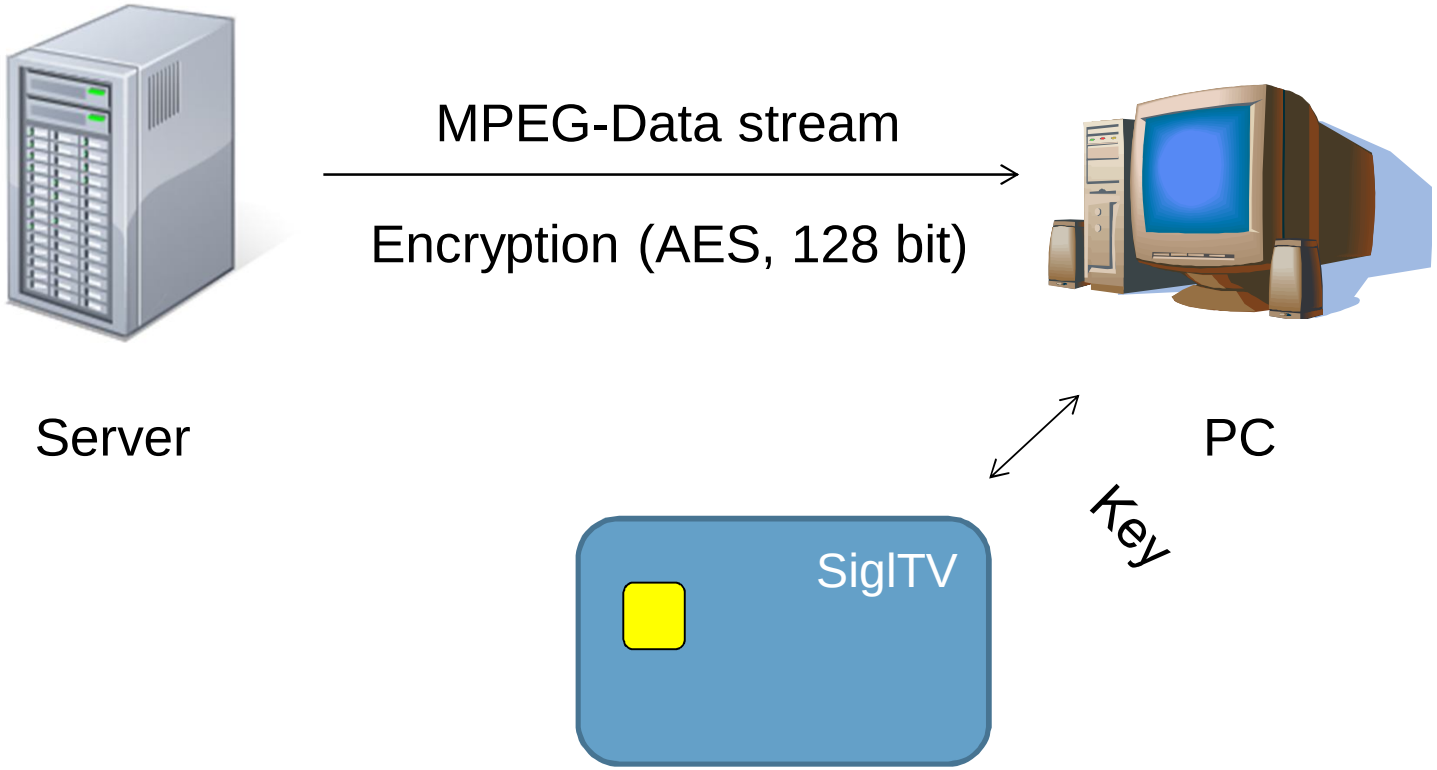
Learning Objectives

The main objective of the laboratory is to analyze the tradeoffs between different secure implementations and their cost





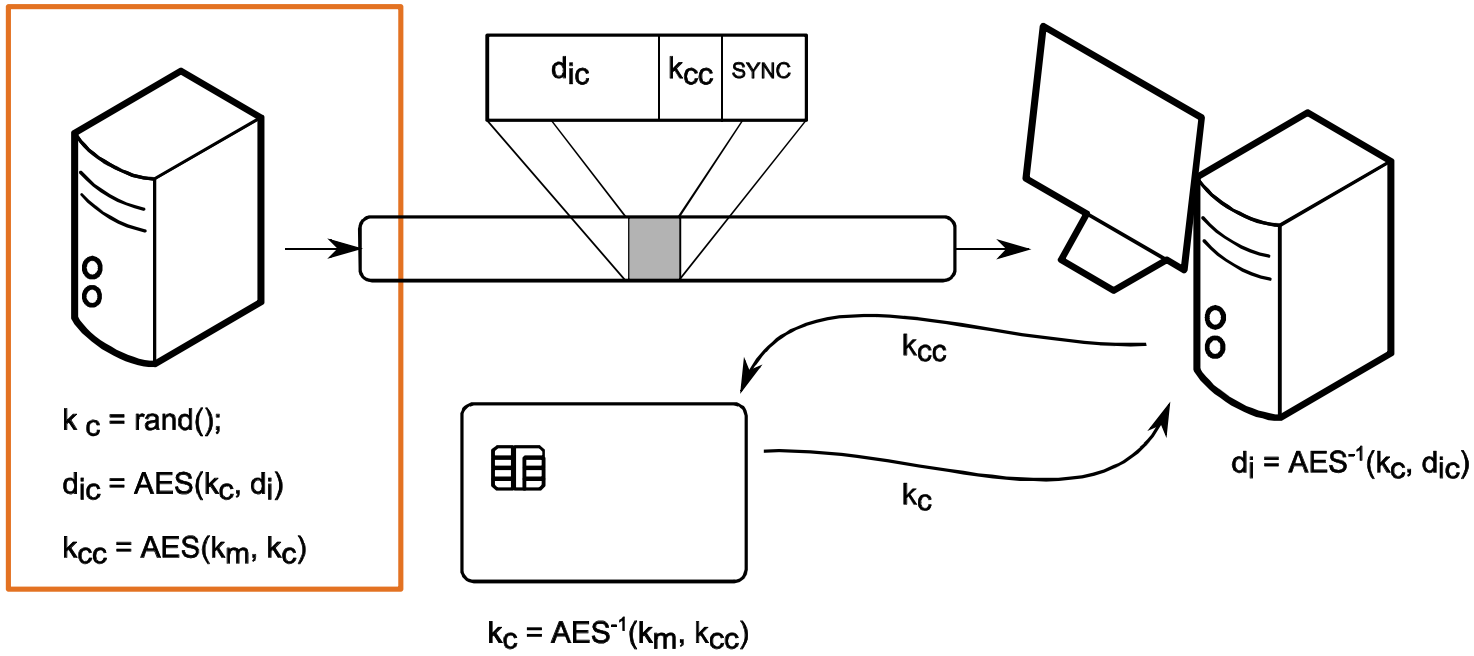
Structure of the PayTV-System



Structure of the PayTV-System

Server:

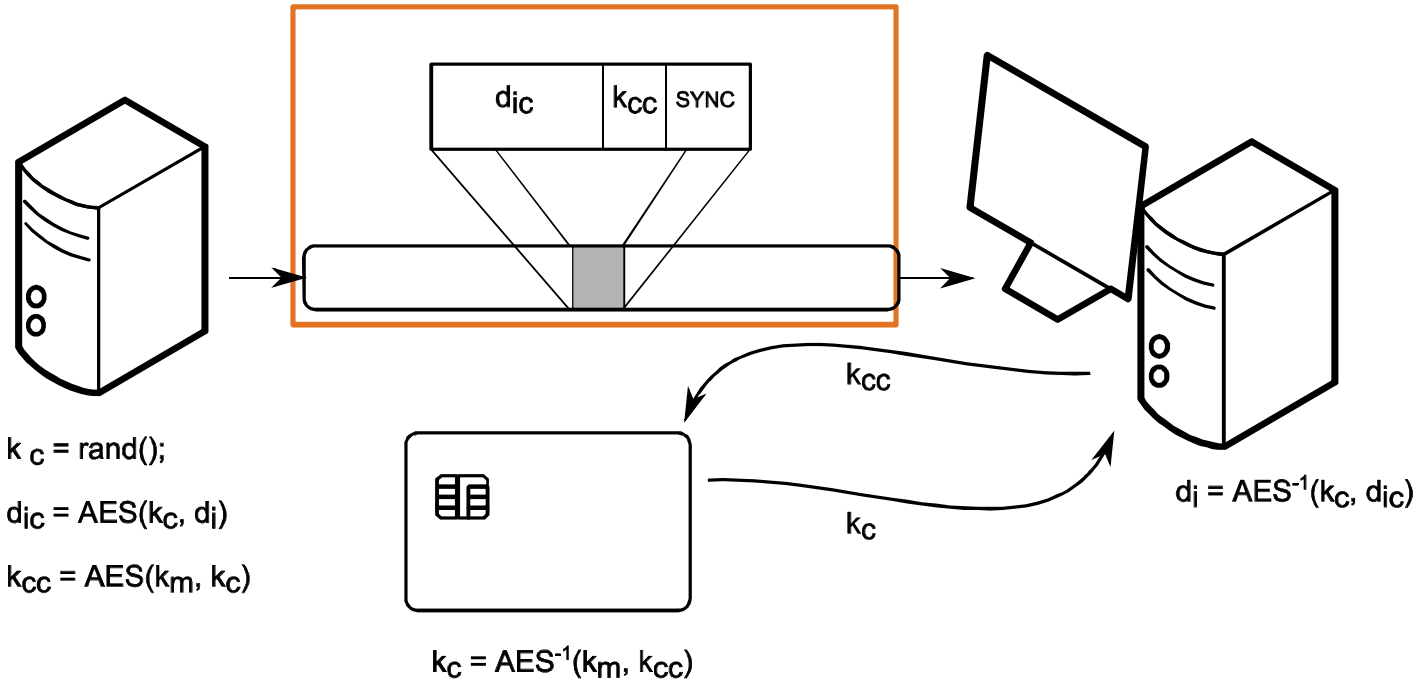
- Video data stream is divided into chunks d_i
- A random key k_c encrypts each data chunk
- The random key k_c is then encrypted with a master key k_m to generate k_{cc}
  - k_m is known only by the server and the card



Structure of the PayTV-System

Server:

- Sends packets that include
  - An encrypted data chunk, d_{ic}
  - The encrypted random key, k_{cc}
  - Synchronization data



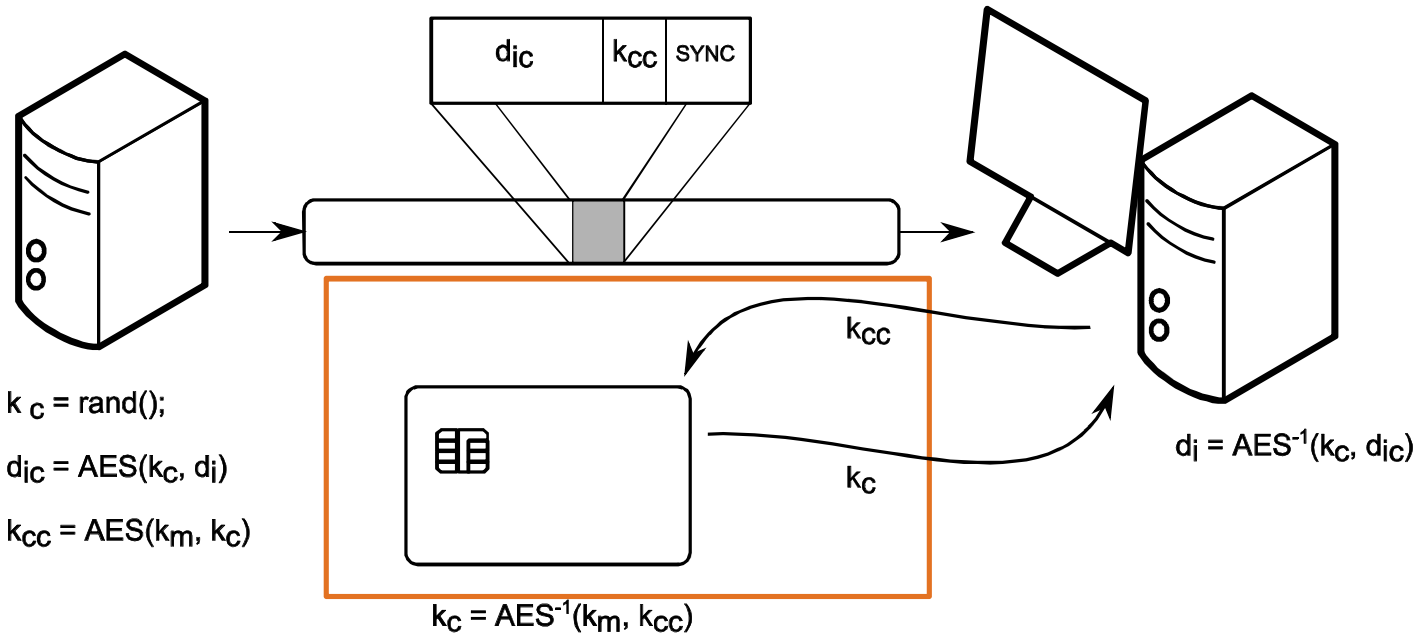
Structure of the PayTV-System

PC:

- Sends the encrypted random key, k_{cc} , to the smartcard

Smartcard:

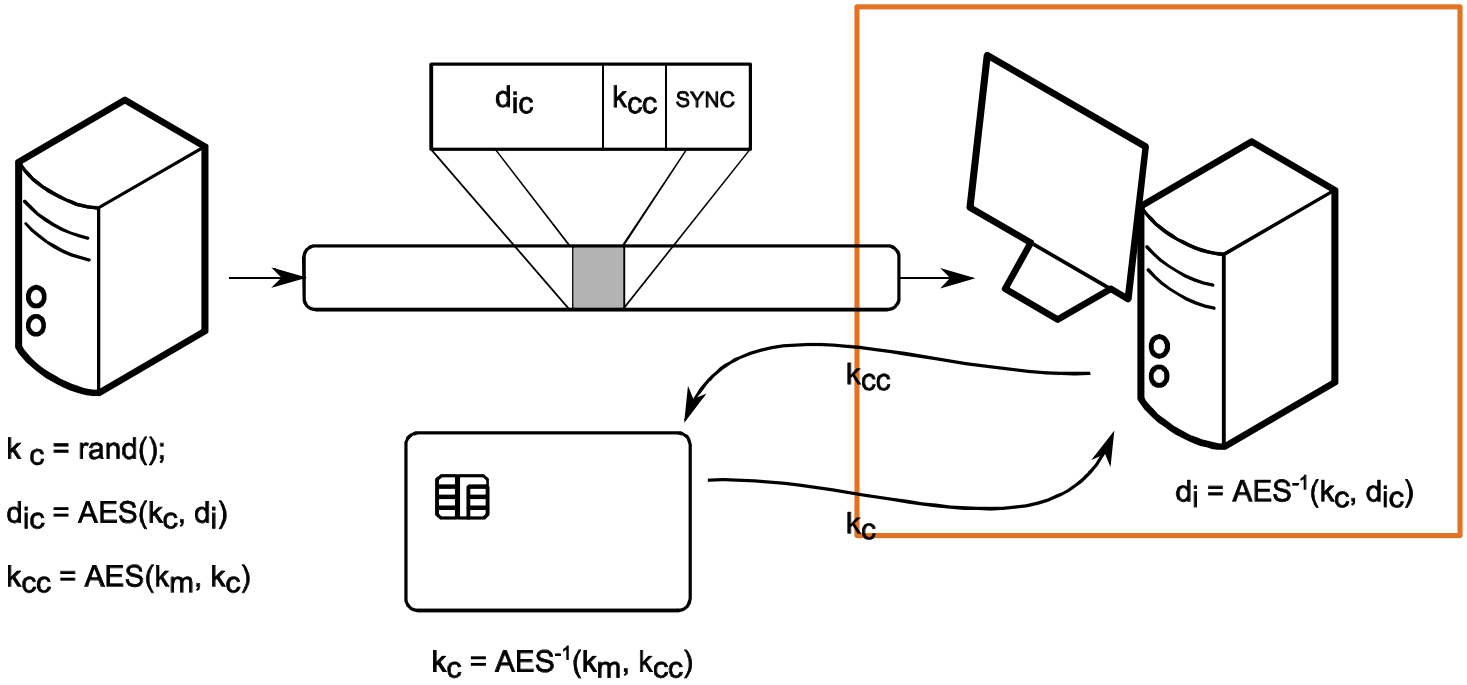
- Decrypts k_{cc} with k_m to obtain k_c



Structure of the PayTV-System

PC side:

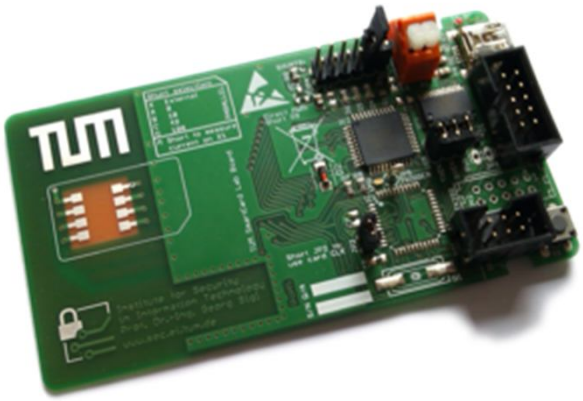
- Uses k_c to decrypt d_{ic}
- Displays the plaintext data chunk d_i



Structure of the PayTV-System

Reference implementation

- Smartcard emulator using an ATMega644
- Preloaded master key



PC-Software

- Some lines of Python code
- LAN video client

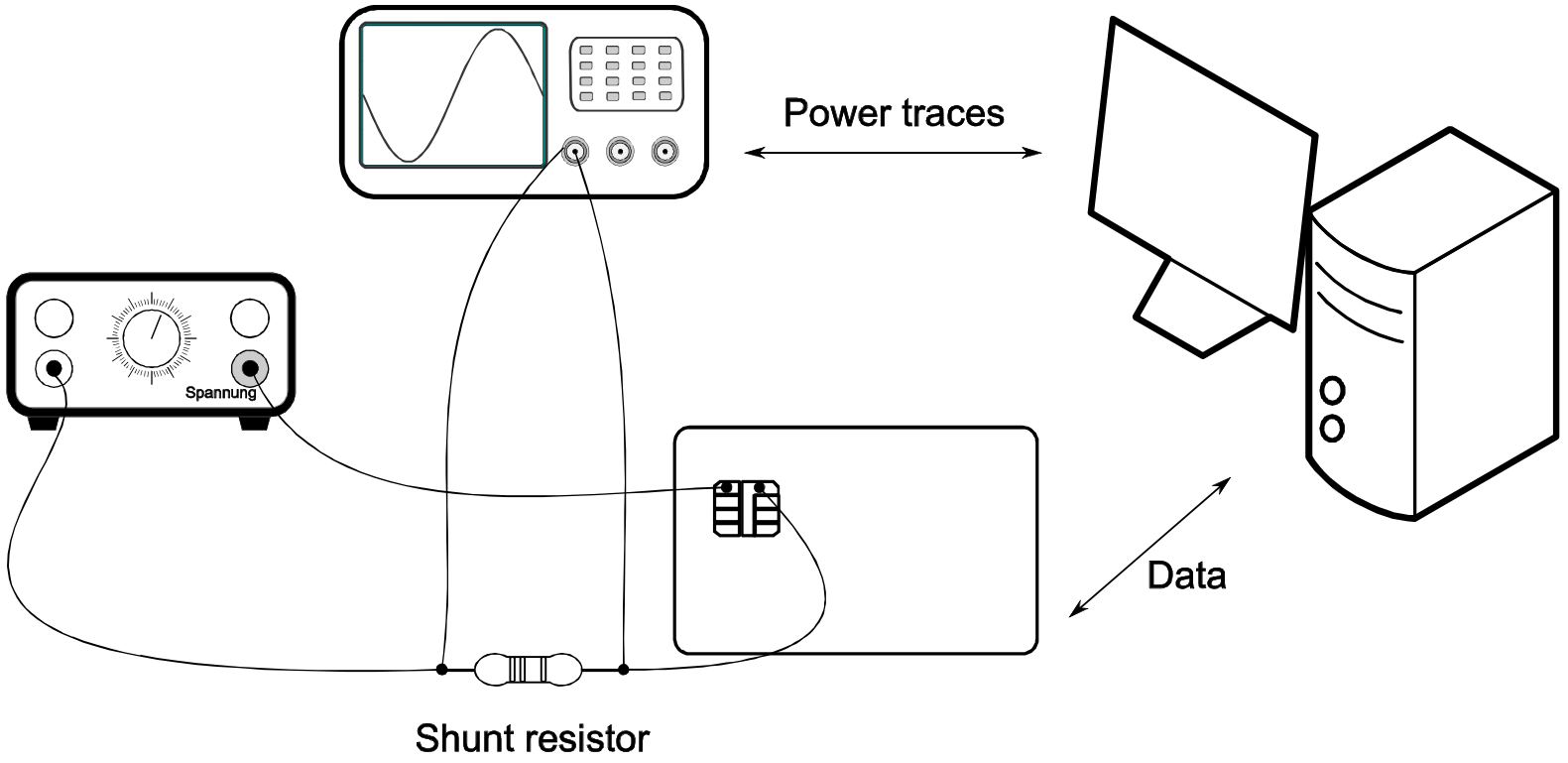
Server

Side-channel Analysis

Differential Power Analysis

- The key will be extracted using the current profile
- Measurement of the current profile:
  - Send data to the smartcard crypto function
  - Measure the current of the operation with a shunt-resistor
  - Record the power traces with an oscilloscope
  - Analyze and process the traces offline (Matlab / Python)

Implementing the DPA



Cryptographic-key Extraction

Steps

1. Assume a power model
2. Create a Key hypothesis
3. Check the correlation: Hypothesis ⇔ Measured Current Profile
4. Find out the key with the highest probability

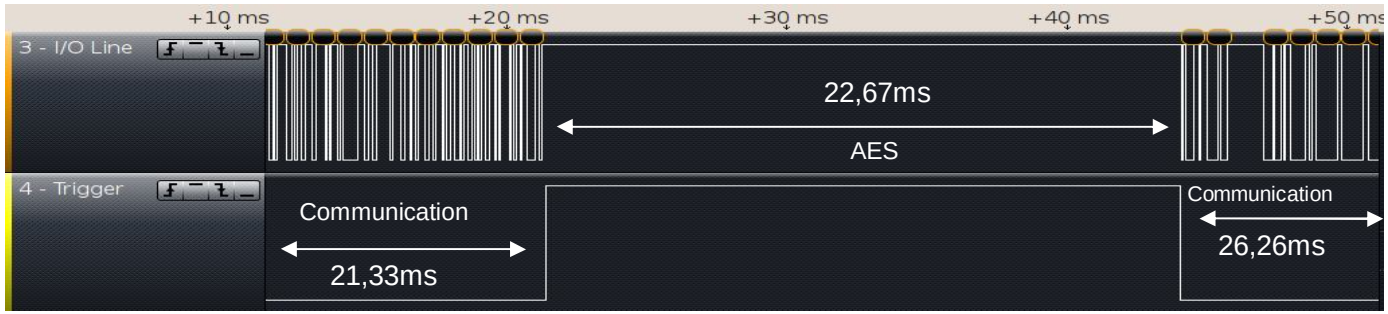
Details will be given on the second lecture

- Introduction to Differential Power Analysis

Smartcard Clone

Program the ATMega644 of your emulator card

1. Create a basic Smartcard OS following the ISO7816 standard
2. Implement AES-128 in software
3. Use the previously extracted key as master key
4. Test the functionality of the smartcard clone

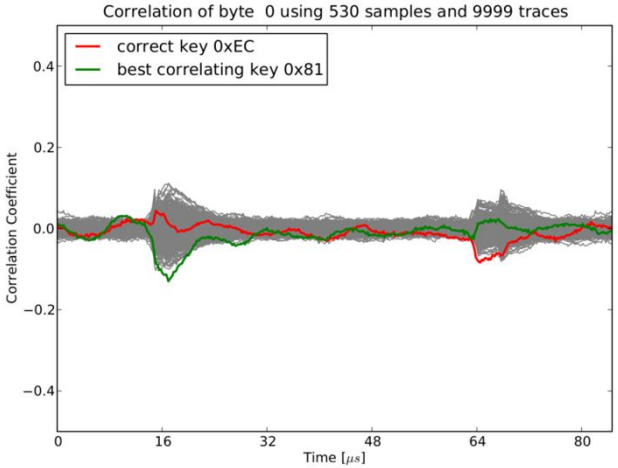


Improving the Security

Starting point: Your own implementation

Harden your code to protect it against DPA

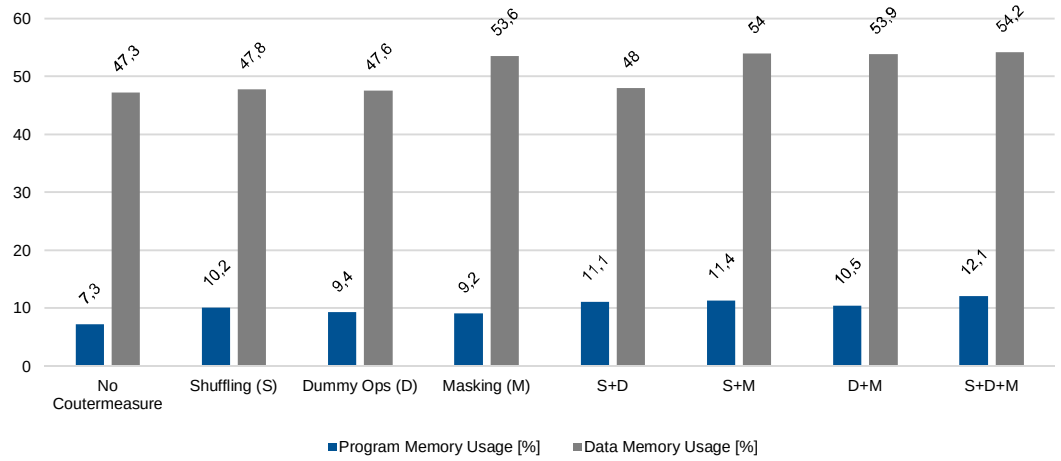
- Leakage hiding techniques
  - Random wait states
  - Shuffling operations
- Intermediate values masking



Evaluation of the Countermeasures

Attempt to break your own countermeasures

1. Measure the resistance against your attack techniques
2. Adapt your attacks
3. Improve your countermeasures
4. Document the results
5. Rinse and repeat!



Section 4 – Smartcard Lab

WORK PLAN

Work plan

Pre-Lab Assignment (on your own)

- Matlab / Python
- AVR
- Project administration and milestones planning

(Basic) Checklist for Matlab/Python:

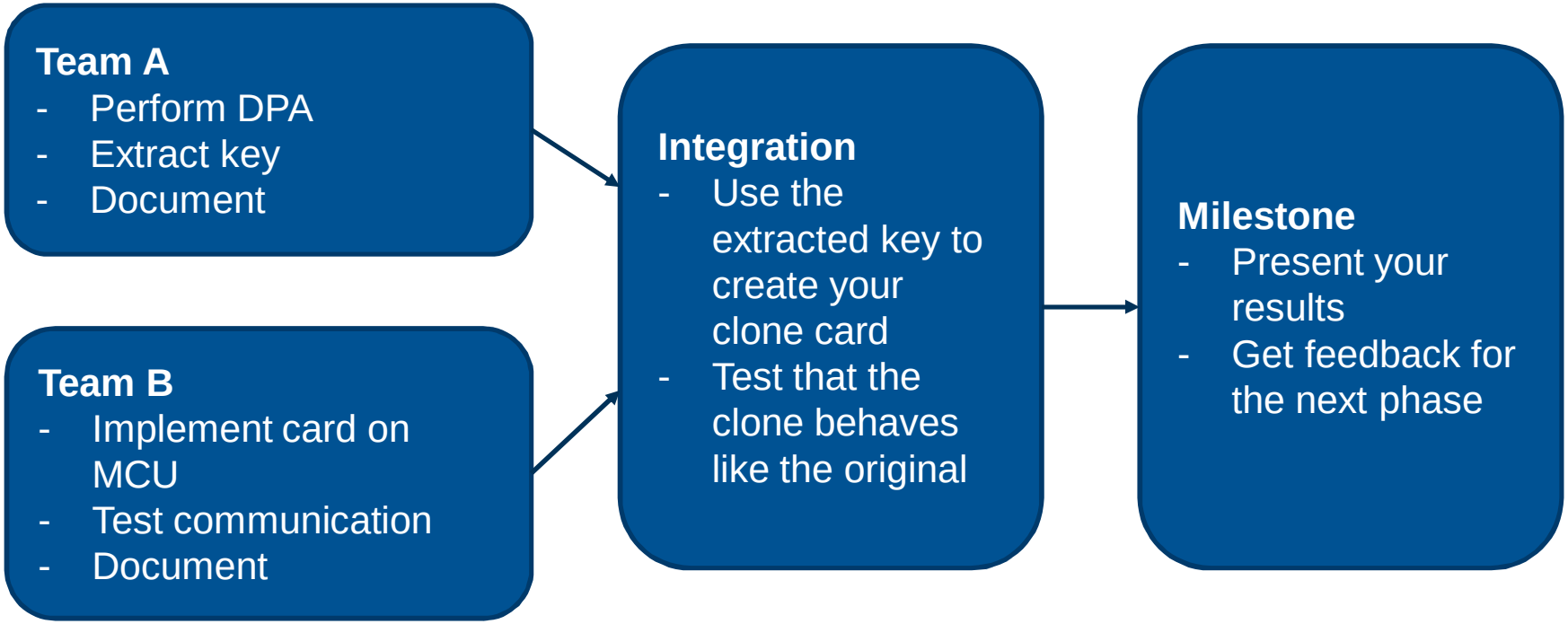
- ❑ Working with .csv files
- ❑ Working with .mat files
- ❑ Vectors manipulation
- ❑ Efficient use of memory
- ❑ Plotting

(Basic) Checklist for AVR:

- ❑ Ports
- ❑ Interrupts
  - ❑ Timers
  - ❑ Pin change
- ❑ UART (testing)
- ❑ Downloading the program

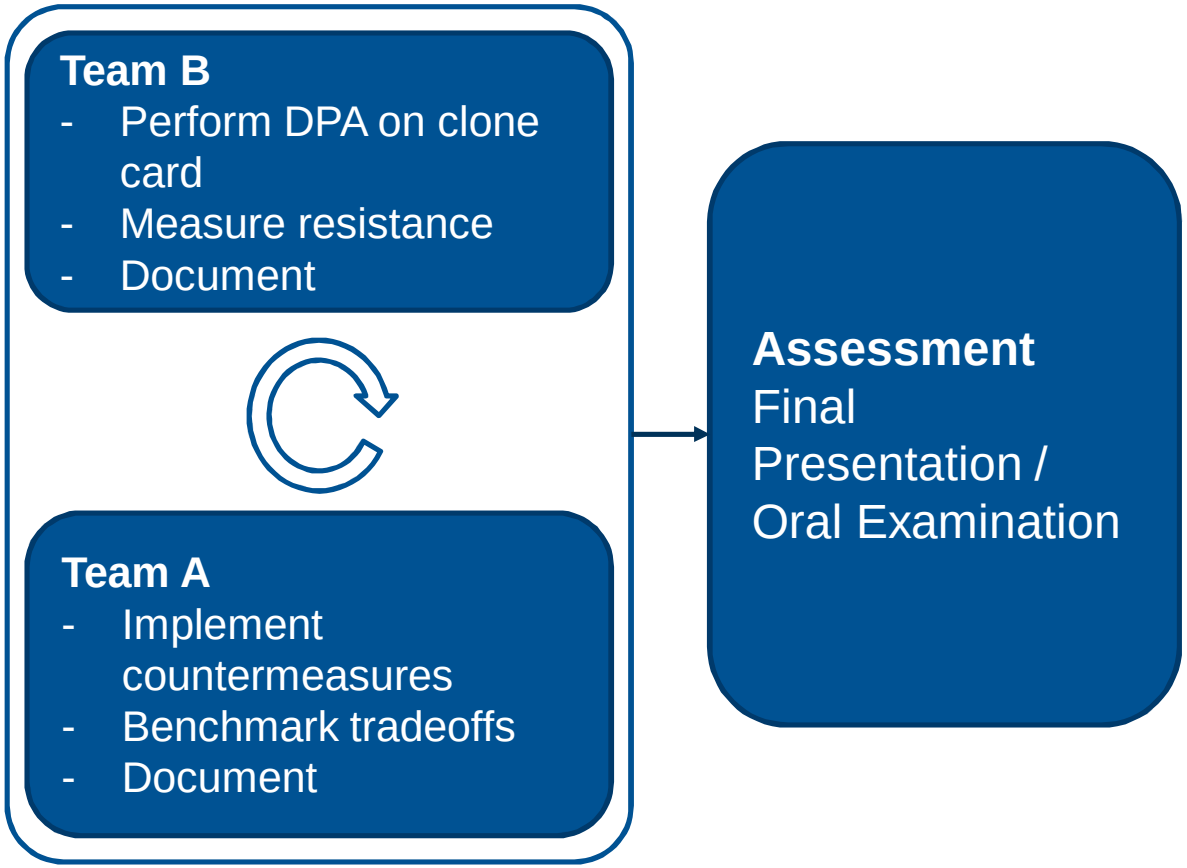
Work plan

Phase 1



Work plan

Phase 2



Milestone presentation

What is expected to see in the **milestone presentation**?

- Differential Power Analysis
  - Details of the implementation / optimizations
  - How many traces did it take to break the key?
  - How fast can you obtain all the key bytes?

- Smartcard clone
  - ISO UART (implementation, sampling strategy)
  - Size of the complete Smartcard OS
  - Size of your AES implementation
  - Speed of your AES implementation

- Project management (who is doing what and when?)
  - Plan
  - Reality

Final presentation

What is expected to see in the **final presentation**?

- Countermeasures
  - Which countermeasures where tested?
  - What is the impact in size / speed?
  - What is the resistance provided by each countermeasure?
- Attack improvements
  - Which type of improvements were made?
  - How are you attacking specific countermeasures?
  - How do the new techniques compare against the simple DPA?
- Project management
  - Plan
  - Reality

Section 5 – Smartcard Lab

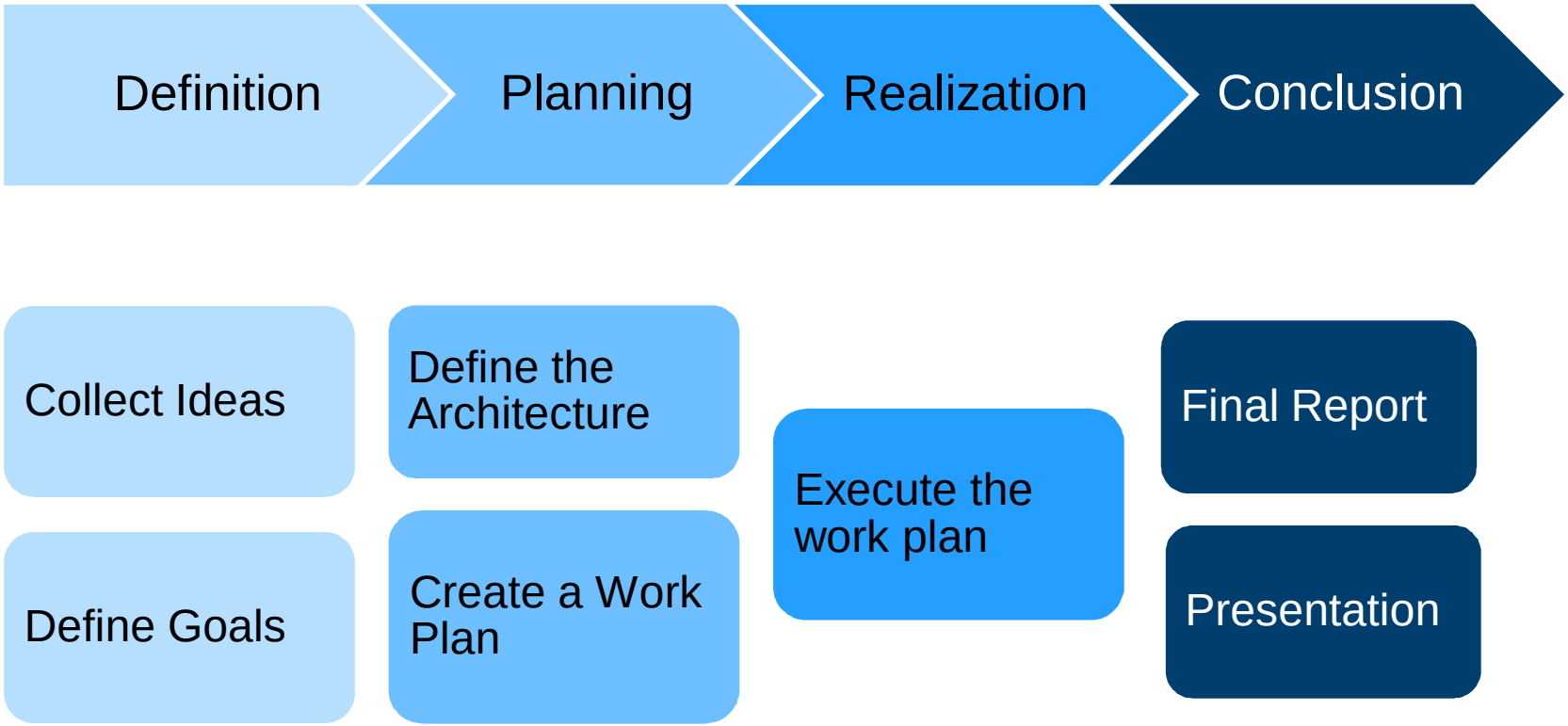
PROJECT MANAGEMENT

Which qualities must a project have?

Definition taken from DIN 69901:

“Project is an undertaking characterized essentially through the uniqueness of the conditions, for instance,
goal,
time, money, personnel, and other ***restrictions***,
the ***scope*** compared with other undertakings,
and project specific ***organization***”

Project Phases



Phase1: Definition

Requirements Specification

- Requirements of the client
- Common Problem: The client often does not know himself exactly what he wants

Feature Specification

- First draft of the plan
- Describes how the contractor will implement the requirements

Project Goals Definition

- Goals must be “SMART”

  (**S**pecific, **M**easurable, **A**ccepted, **R**ealistic, **T**imely)

Phase 2: Planning

Work-breakdown: Structure of subtasks, Tree structure

Process list:

[ID-Nr., Process description, Duration, Predecessors, Resources]

Gantt Charts

Milestones: Important events of the project

System architecture: Relationship between the different components

Interface definitions: Function calls between components

The first plan will still differ from reality,
nevertheless, planning in the first stages is very important!

Phase 3: Realization

Perform the planned operations

Documentation

- As much as needed, as little as possible
- Architecture and steps

Project-Monitoring (Cycle)

- Test
- Check if what it is, is what it is supposed to be
- Correct discrepancies
- Adapt the plan

Phase 4: Conclusion

Final Report

Final Presentation, Demonstration

Client's Approval

Evaluation - „Lessons Learned“

- What has been achieved?
- What problems where there?
- What could be improved in the future?

Section 5 – Smartcard Lab

MANAGEMENT TOOLS

Project management in the Laboratory

Subversion - SVN

- Source-Code version control system
- Tracking your changes
- Synchronization between developers

Software Project Management -Trac

- Interface to SVN
- Wiki for documentation
- Discussion forums
- Milestones
- Tickets
- Gantt Chart, Calendar

Note:

- The documentation of the lab will influence the FINAL GRADE

Crash-course Subversion

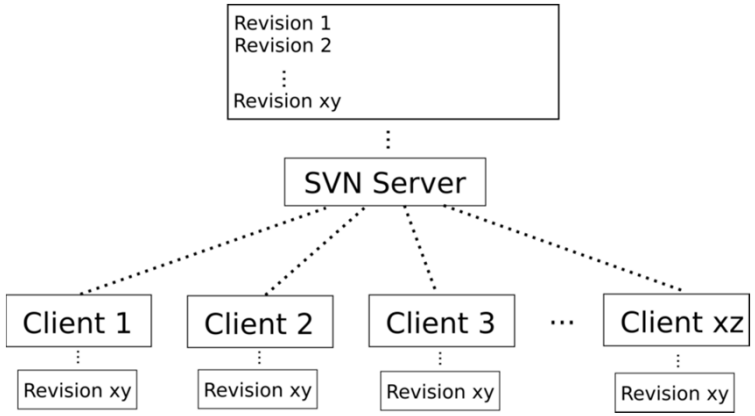
Centralized data storage on a SVN Server – **Repository**

Work with a local copy – **Sandbox**

Commands to synchronize Repository and Sandbox

Convention: Data is organized in three directories:

- `trunk/`
- `branches/`
- `tags/`



source: http://de.wikipedia.org/wiki/Apache_Subversion

Crash-course Subversion

Important commands

Checkout: `svn co https://tueisec-stusvn.sec.ei.tum.de/svn/Projektname`

Update your working copy: `svn up`

Check-in changes into the repository: `svn ci –m “Log-Eintrag“`

Print the status of working copy files and directories: `svn status`

Add files to the repository: `svn add filename.c`

Delete a file from the repository: `svn del filename.c`

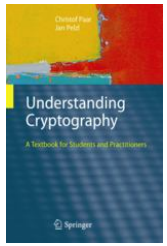
Graphical Front-Ends e.g. Tortoise: http://tortoisesvn.tigris.org

Crash-course TRAC

TRAC Live-Demonstration

https://tueisec-subversion.sec.ei.tum.de/trac/TestProject

Literature



Main Literature

- ISO/IEC 7816 Standard
- Power Analysis Attacks: Revealing the Secrets of Smart Cards
  Stefan Mangard, Thomas Popp, Elisabeth Oswald, ISBN-13: 978-0387308579

Additional Literature

- Handbuch der Chipkarten
  Wolfgang Rankl und Wolfgang Effing, ISBN-13: 978-3-446-40402-1
- Understanding Cryptography
  Christof Paar and Jan Pelzl, ISBN-13: 978-3-642-04100-6

Questions?

THANK YOU FOR YOUR ATTENTION!