Smart Card Laboratory

Introductory Lecture

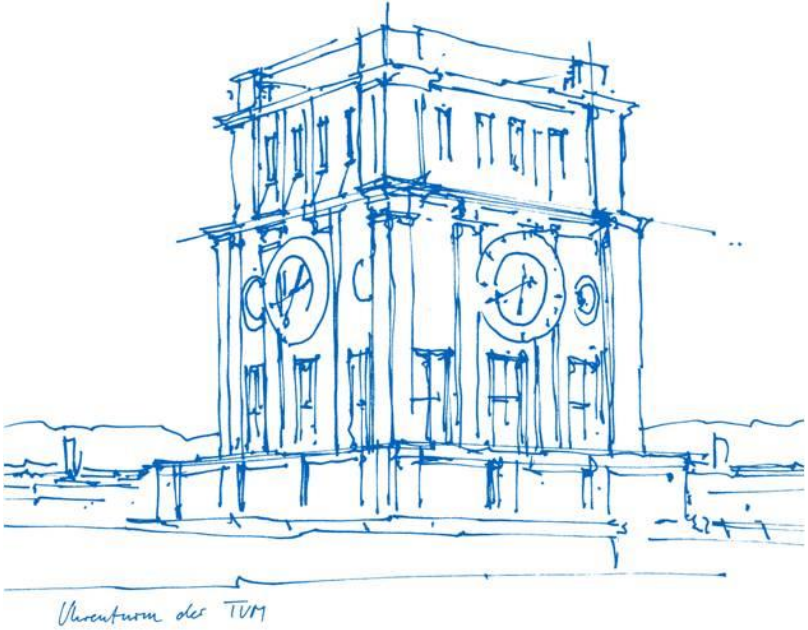
Tim Music

Chair of Security in Information Technology

TUM School of Computation, Information and Technology

Technical University of Munich

München, 23.10.2024



Contact

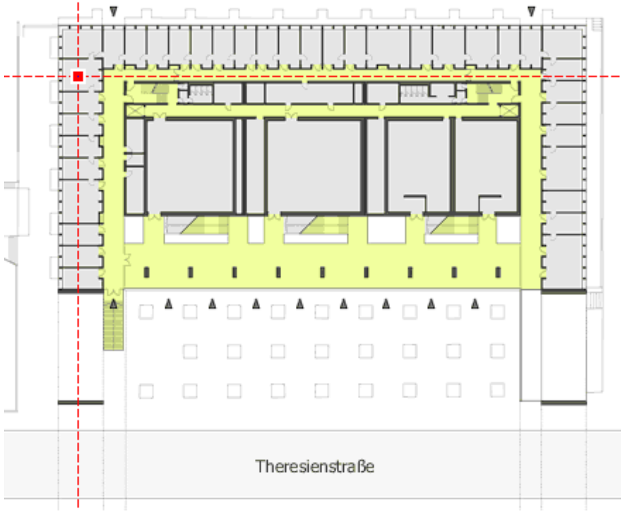
Lab Instructor

Tim Music

Research Topics:
- Physically Unclonable Functions (PUFs)
- ASIC Design
- Side Channel & Fault Attacks

Room N1010 / N1 ZG
tim.music@tum.de

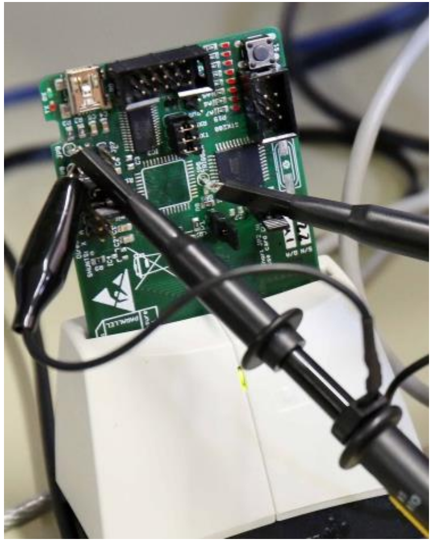
For consultation hours please schedule an appointment by mail



The Smart Card Laboratory in a Nutshell

What You will be Learning in this Course

- Focus on hardware security, specifically a **differential power analysis** (DPA) on the power side-channel
- **Design**, **implement** and **debug** a DPA hardened smart card
- Evaluate cost / benefit **tradeoffs** of **security measures**
- Plan, manage and execute a sizable project as a **team**



Outline

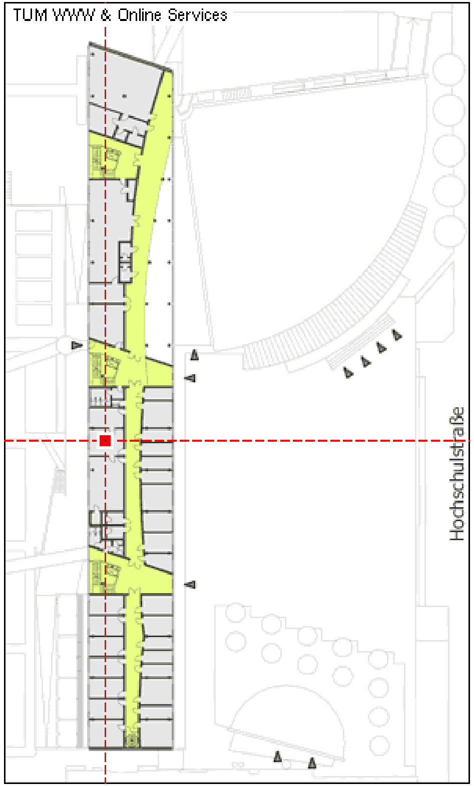
Administrative Topics

Laboratory Hours

Student room for measurements: **2947 (0509.03.947)**

The Smart Card Lab can be performed on our own schedule. The room for this is shared with SIKA (Secure Implementation of cryptographic algorithms), PUF and ASIC Lab.

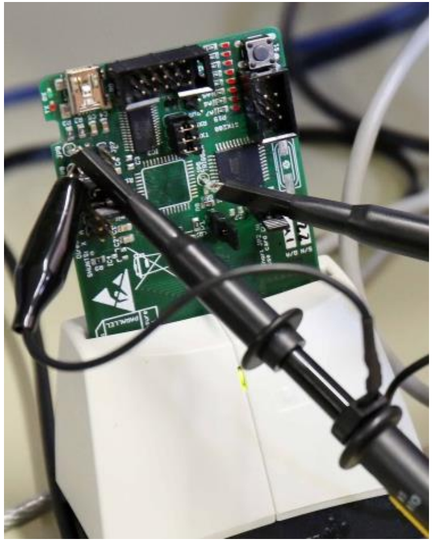
Accessible: Monday – Friday during TUM opening times.



Prerequisites

Recommended prior knowledge when taking this course

- General understanding of digital circuits and architecture of microcontrollers
- Initial experience coding in C for embedded platforms
- Initial experience with oscilloscope/logic analyzers
- **Optional theory course:** Course on Secure Implementation of Cryptographic Algorithms (SICA)



Literature

Main Literature

Power Analysis Attacks: Revealing the Secrets of Smart Cards

Stefan Mangard, Thomas Popp, Elisabeth Oswald,

ISBN-13: 978-0387308579

ISO/IEC 7816 Standard

Additional Literature

Smart Card Handbook

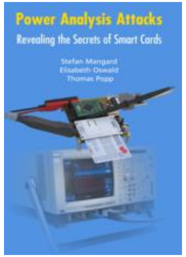
Wolfgang Rankl und Wolfgang Effing,

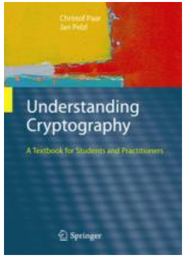
ISBN-13: 978-3-446-40402-1

Understanding Cryptography

Christof Paar and Jan Pelzl,

ISBN-13: 978-3-642-04100-6





Academic Integrity

- All work you submit for this laboratory must be your own

- If you use code/ideas from other people
- Must be clearly and explicitly noted
- Must have a proper and complete citation
- The gist of this lab course is to gather hands-on experience in implementations.

I should not have to write this, but

don‘t copy crypto libraries from the internet.

Laboratory Logistics

Date	Time	Place	Description
23.10.2024 (Wed)	10:00 – 11:30	N1005ZG	Introductory Lecture
31.10.2024 (Thu)	13:00 – 14:30	N1005ZG	Side Channel Attack Lecture
06.11.2024 (Wed)	09:30 – 11:00	2947 (Lab)	Hardware Handout
11.12.2024 (Thu)	08:30 – 12:00	N1005ZG	Intermediate Presentation
22.01.2025 (Mon)	23:59	Moodle	Report Deadline
22.01.2025 (Mon)	Final Presentation	N1005ZG	Final Presentation
Second Week February		N1011ZG	Oral Exam

Please consider that the oral exam date may be subject to change.
Also remember to register for the exam in time!

Laboratory Tools

Smart Cards
(reference & blank card)

Programmer
ST-Link V2

Logic Analyzer
Saleae Logic 8

Oscilloscope
Picoscope 5000 Series

Moodle

Using the TUM Moodle Platform for

Distributing lecture slides & lab script

Forum for general questions

Lab Reports

Team Assignment

Laboratory Version Control

Git Version Control for

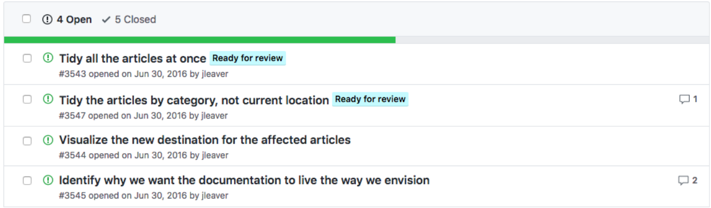
- Tracking your changes
- Synchronization between developers

Tip:

- Git issues aid centralized discussion of design choices and source code issues
- Transfer project milestones into git

We will use the LRZ Gitlab Instance:

https://gitlab.lrz.de



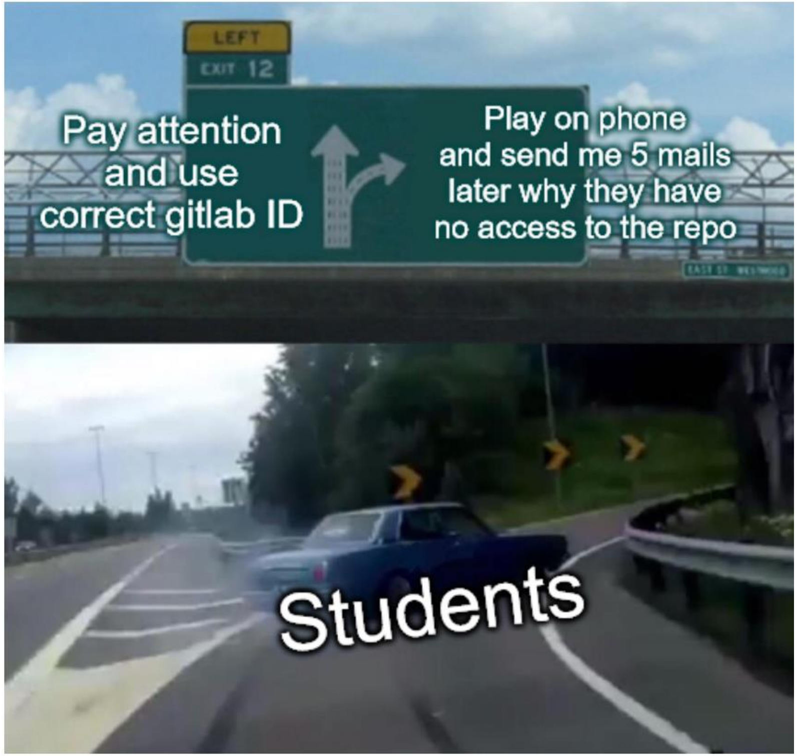
But wait…

Did you know that on average 2 out of 10 master students take a nap during the Gitlab introduction.

As a consequence, they miss how to register in Gitlab.

Instead, they prefer sending me emails.

Gitlab Registration Break



Grading

Lab Work

- Lab report influences the questions asked and to ensure fair work distribution
- We will have a look at your code and ensure it was your own work
- Lab reports are due for submission a few days before each of the presentations
- Lab report will count 50 % towards your grade

Presentations

- Take place in the student room, where you present your project progress, following some questions
- 10 minutes per team
- If desired, you are allowed to have a 20 minutes final presentation in addition to your lab report

Oral Examination

- Examination on the lecture’s theoretical background
- Counts 50 %, 10 minutes

Security & Implementations

(some history to it, and why having a secure algorithm isn’t enough)

Usage of Cryptography

Caesar Cipher (100 BC)

Rotation of the Alphabet by 13 letters

Vigenère Cipher (1467 AD)

- First cipher to use a key
- Uses interwoven Caesar ciphers
- Key addition modulo 26
- Took three centuries to break

Herbern Rotor Machine (1908)

- Electro mechanical. Typewriter-like.

Single rotor.

Key embedded in the disc.

Enigma Machine (1920)

- Multiple rotors.

Developed after WWI

Widely used during WWII

Data Encryption Standard (1977)

- IBM's "Lucifer" cipher.
- Standardized by the National Bureau of Standards

Advanced Encryption Standard (2001)

- Organized by NIST

50 submissions

Winner: Rijndael cipher

CEASAR Auth. Encryption (2012 - 2019)

15 candidates

Three winners for different use cases

NIST Post-Quantum Cryptography Competition (2016 – 2020)

Winners:

4 encryption standards

3 digital signature algorithms

NIST Lightweight Competition (2018 - 2022)

Crypto algorithms for resource constrained environments

57 submissions

10 finalists for May 2022

Military use

Commercial use

Cryptography Trends

The use of cryptography in everyday devices is becoming more prominent and important

Examples

Smartcards	Medical Monitoring / Telemetry	Cloud Computing	Internet of Things	Car-to-X	Advance Metering Infrastructure (Smart Grid)

Example: Chipcards

Telecommunication

- Telephone cards, SIM-Cards

Payment

- e-Purses, Credit cards

Access control

- Access ID, Public transportation cards

Identification

- Passport, Driving license, Medical cards

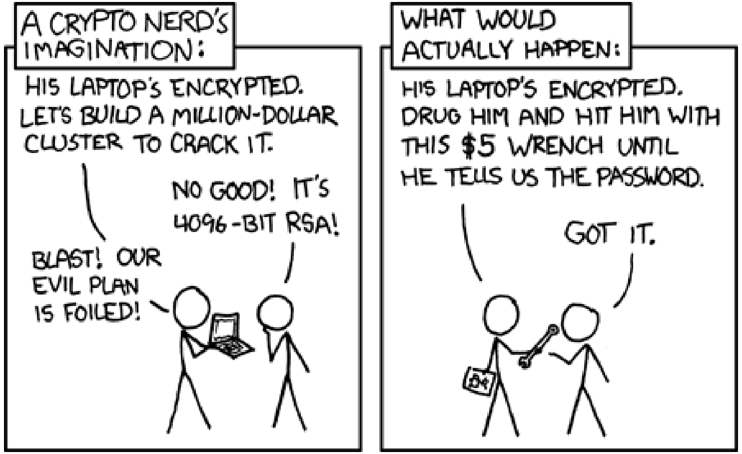
Digital Rights Management

- Pay TV

Image source: http://siliconangle.com

System Security Challenges

Compromising a system usually involves breaking the weakest link



In this lab's scenario this boils down to the secure implementations of a cipher.
Also, I would get in serious trouble if I was suggesting hitting people with wrenches.
You would not know your card's secret keys anyways.

Image source: https://imgs.xkcd.com/comics/security.png

Implementation Challenges

Every designer will somehow be involved with the topic of security when designing a system (e.g., piracy prevention)

Devices that make use of cryptography are in the hands of many users (and attackers)

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

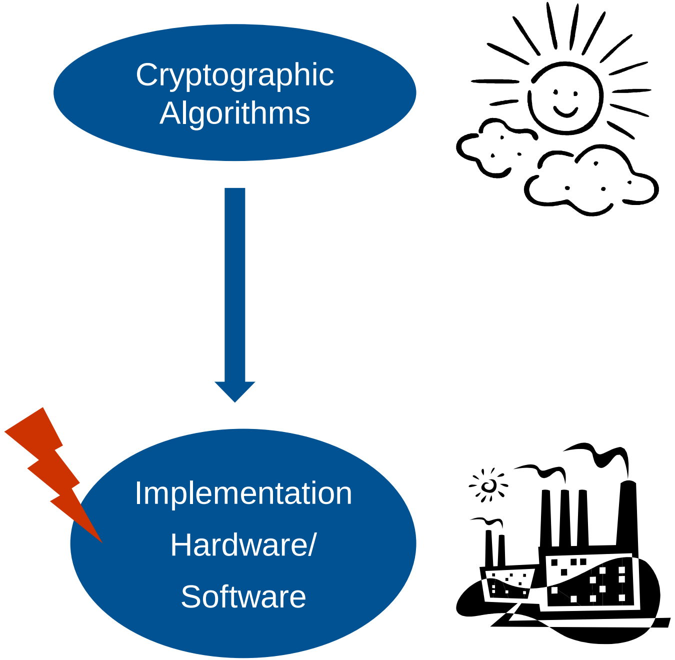
Investments into security are frequently neglected for further product features (time & money)

Implementation Challenges

The cryptographic algorithm itself can be strong
(i.e., good trapdoor, no known or hard to exploit algorithmic weaknesses)

Implementation in hardware or software can (and will) be exploited

Appropriate security measures need to be evaluated and implemented



Examples on past Hardware Security

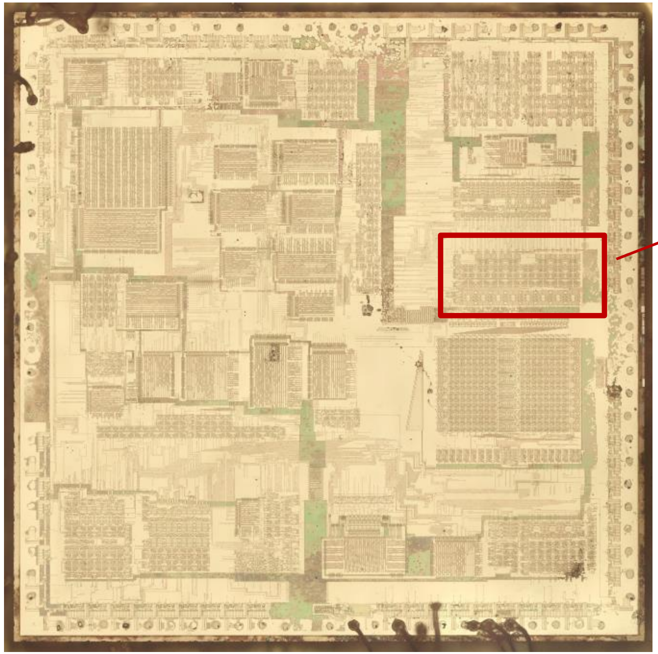
(what not to do)

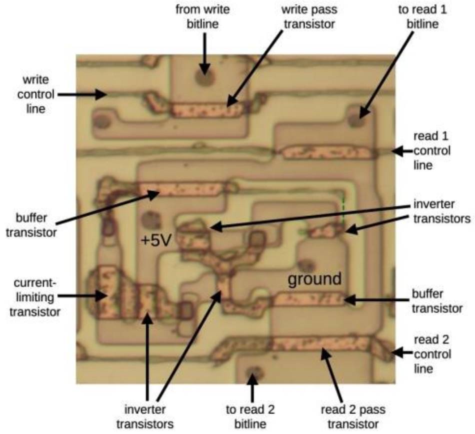
Mifare Classic

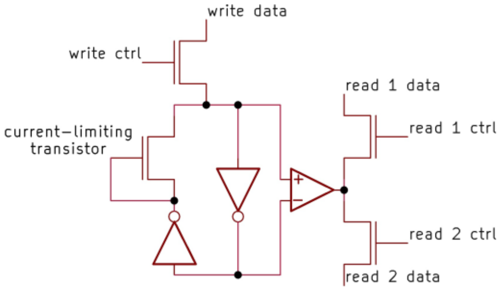
Access control and ticketing systems (e.g. Oyster Card in London)
Contactless memory card, crypto in Hardware

Image: https://en.wikipedia.org/wiki/File:Oyster_card.png

Mifare Classic







This in fact is not a smart card chip, but perfectly outlines the concept.
You can apply this to any chip in a modern digital design flow.

Images:
http://www.righto.com/2024/07/ibm-3274-keystone-chip.html
http://www.righto.com/2024/07/pentium-standard-cells.html

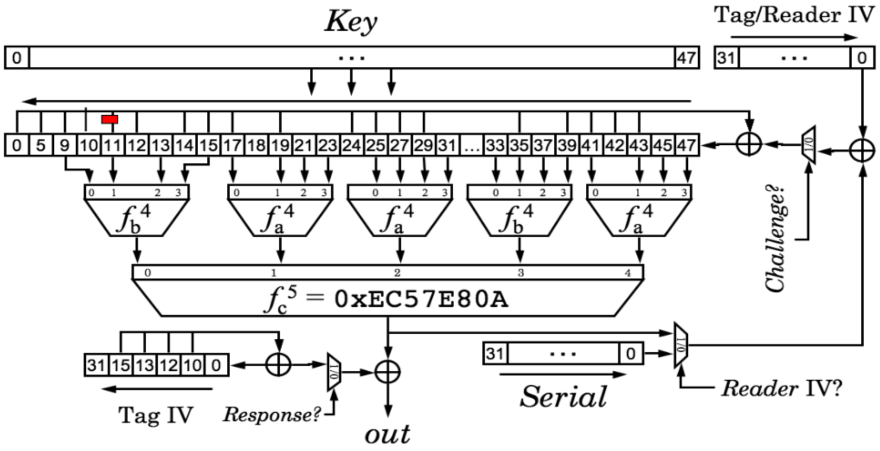
Mifare Classic

Security was broken in 2007 by researchers at the Humboldt-Universität Berlin and Radboud Universiteit Nijmegen

- Gate-level Reverse Engineering
- Protocol Analysis
- Emulators

Weaknesses

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



Details and image: https://www.cs.ru.nl/~flaviog/publications/Dismantling.Mifare.pdf

Lessons learned: Do not use proprietary crypto. Have a proper random number generator.

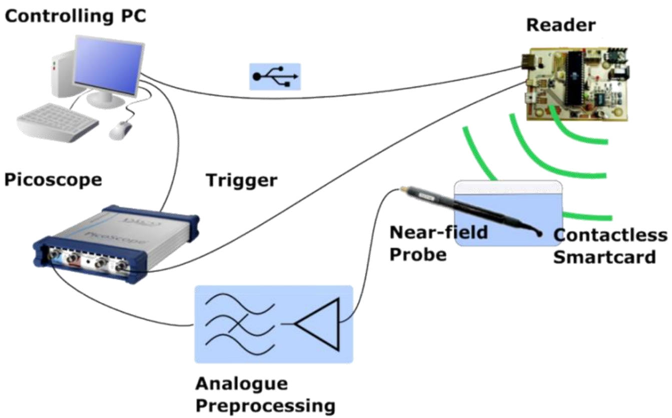
Mifare DESFire

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

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

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)

Lessons learned: Secure your algorithm against side-channel attacks

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

USIM Cards

3G UMTS / 4G LTE Cards using MILEANAGE algorithm (AES-based)

Mutual authentication protocol designed to remediate problems found in GSM (base station spoofing)

Weaknesses:

- Cheap USIM cards do not provide resistance to Side Channel Attacks
- For compatibility, phones downgrade to 2G networks when no 3G/4G/5G available

Lessons learned: Compatibility measures can downgrade your security. Again: SCA Resistance

Details: http://perso.uclouvain.be/fstandae/PUBLIS/161.pdf
https://youtu.be/x8exHMhGy1Q (Black Hat 2015)

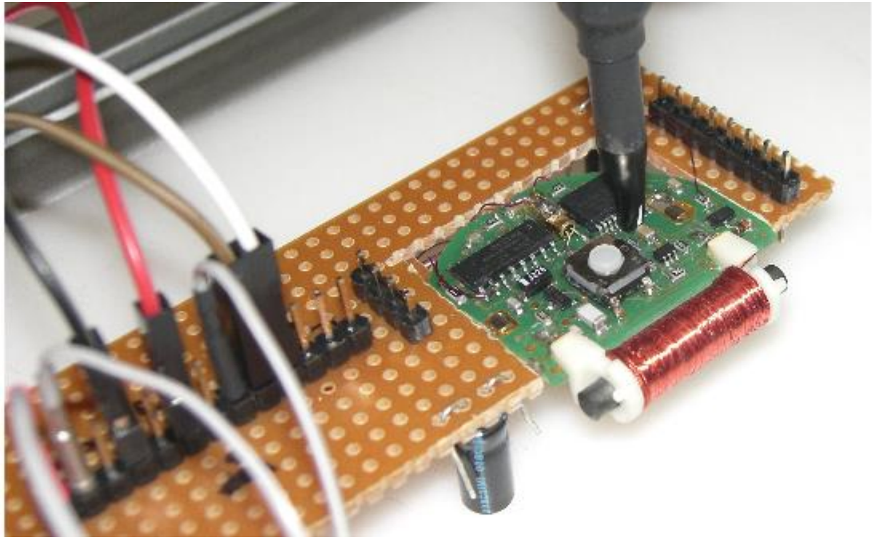
Access Control – Locking System

Transponder based facility 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



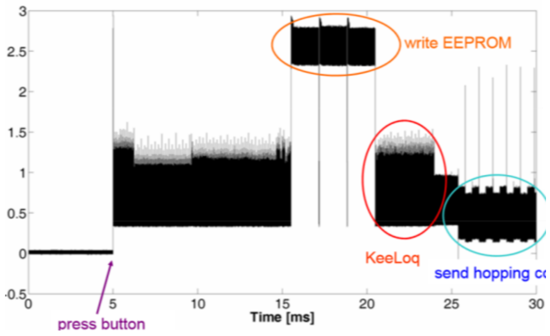
Details: https://eprint.iacr.org/2008/058.pdf

KeeLoq

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

Algorithm implemented in Hardware (NLFSR)

Widely used by Chrysler, Fiat, GM, Honda, Toyota, Volvo, VW, Jaguar, and more…



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 on the Internet)
- Susceptibility to side-channel attacks

Details: https://eprint.iacr.org/2008/058.pdf

But all of this feels kind of theoretical…

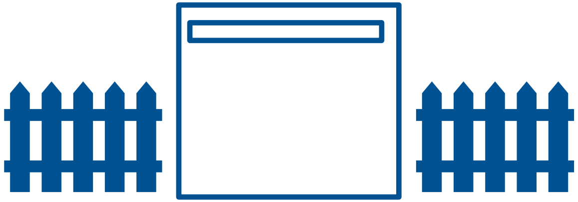
… indeed, the probability for you to be targeted as an individual is quite low.

But you just may be in the wrong place at the wrong time and belong to an unfortunate group of people.

So let us just assume you are.

Let us have a look at your house

Your House



Your Garage

Your Garage Opener

Image: https://www.magickey.com.au/garage-door-remote/Gliderol-TM-27Mhz-GOLD

Importance of a large enough key space

- Widespread remotes have a set of DIP switches on their back to conveniently program the key
- If receiver and transmitter are configured identically, the gate opens
- Most of these devices use an eight to twelve bit key

- If a key on the remote is pressed, all key bits are simply broadcast using ASK/OOK modulation (no framing)

- You are the attacker
- The goal is to *brute force* all possible combinations for an 8-bit to 12-bit fixed key garage to get access to any garage
- How many *bits* do you need to send in total to cover all key spaces?
- How much time would this take if the transmitter needs 2 ms to send one bit, followed by a 2 ms break. As there is no collision detection you need to transmit each packet 5 times.

Importance of a large enough key space II

Total amount of bits to transmit:

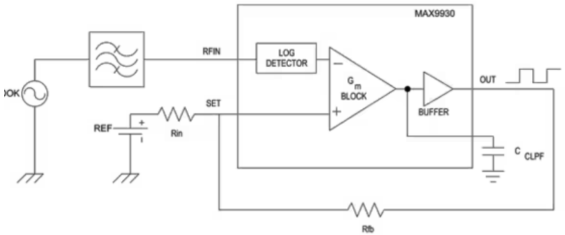
$$2^{12} \cdot 12 + 2^{11} \cdot 11 + 2^{10} \cdot 10 + 2^{9} \cdot 9 + 2^{8} \cdot 8$$

At 2 ms per bit, 2 ms delay, 5 times this resolves to ~ 30 mins.

Not too bad, not too good either. How to improve?

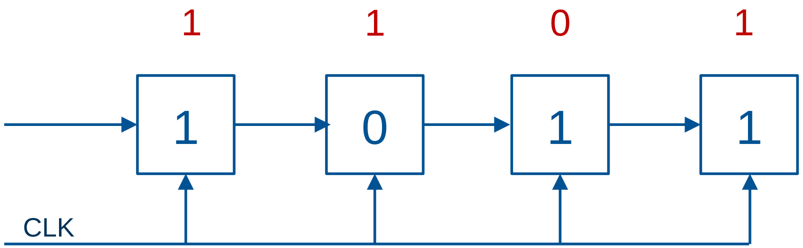
Importance of a large enough key space III

Receiver Schematic



Receiver Analog Frontend

Correct Key



Shift Register

Possible Improvements:

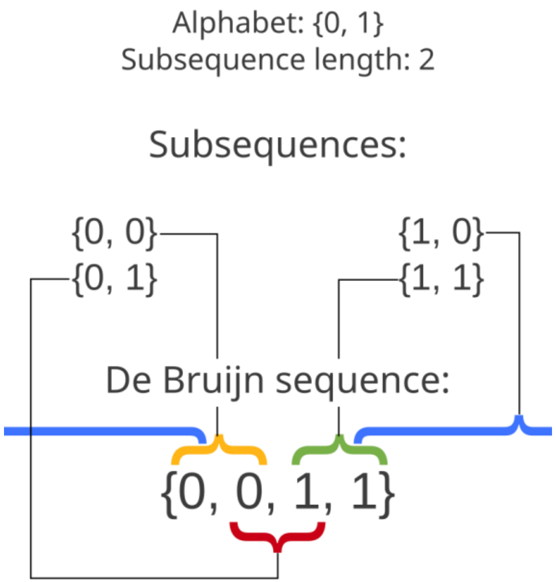
- Remove the transmit breaks
- Get a directional antenna to avoid retransmissions (log detector)
- Shift register property: All 8-bit – 11-bit codes are covered by all 12-bit codes

Details and image: https://www.digikey.de/de/articles/im-ook--youre-ook

Importance of a large enough key space IV

- Dutch mathematician Nicolaas Govert de Bruijn wrote about likenamed sequences in 1946
- A de-Bruijn sequence is a cyclic sequence in which every possible string of length n occurs exactly once as a substring
- For an binary alphabet, the length of these strings is 2^n
- As this is a cyclic sequence, we need to append the first 2^n-1 bits to the end of the sequence

- Total Time: $(2^{12} + 11) * 2\ ms \approx 8\ s$



Details and image: https://en.wikipedia.org/wiki/De_Bruijn_sequence

De Bruijn Sequence for 12-bit

000000000000100000000001100000000010100000000001110000000001001000000000101100000000011010000000001111000000001000100000000100110000000010101000000001011100000000110010
000000110110000000011101000000001111100000001000010000000100011000000010010100000001001110000000101001000000010101100000001011010000000101111000000011000100000001100110000
000110101000000011011100000001110010000000111011000000011110100000001111110000001000001000011000000100010100000010001110000001001001000000100101100000010011010000000100111
100000010100010000001010011000000101010100000010101110000001011001000000101101100000010111010000001011111000000110000100000011000110000001100101000000110011100000011010010
000011010110000001101101000000110111100000011100010000001110011000000111010100000011101110000001111001000000111101100000011111010000001111111000010000010100001000001110000
010001001000001000101100000100011010000010001111000001001000100000100100110000010010101000001001011100000100110010000010011011000001001110100000100111110000010100011000001
010010100000101001110000010101001000001010101100000101011010000010101111000001011000100000101100110000010110101000001011011100000101110010000010111011000001011110100000101
111110000011000011000010100001100011100001100100100001100101100001100110100001100111100001101000100001101001100001101010100001101011100001101100100001101101100000
011011101000011011111000011100011000011100101000011100111000011101001000011101011000011101101000011101111000011110001000011110011000011110101000011110111000011111000001
111100100001111101100001111110100001111111100010001000100110001000101010001000101110001000110010001000110110001000111010001000111110001001000110001001001010001001001011000
100100111000100101001000100101011000100101101000100101111000100110011000100110101000100110111000100111001000100111011000100111101000100111111000101000101000101000110001
1000101001001000101001011000101001101000101001111000101010011000101010101000101010111000101011001000101011011000101011101000101011111000101100011000101100101000101100101010
001011001110001011010010001011010110001011011010001011011110001011100110001011101010001011101110001011110010001011110110001011111010001011111110001100011000110001101000
110001111000110010011000110010101000110010111000110011001000110011011000110011101000110011111000110100101000110100111000110101001000110101011000110101101000110101101000011
010111100011011001100011011010100011011011100011011100100011011101100011011110100011011111100011100011100100100011100101100011100110100011100111100011101000110
001110101010001110101110001110110010001110110110001110111010001110111110001111001010001111001110001111010010001111010110001111011010001111011110001111100110001111110011000
1111101010001111101110001111110010001111110110001111111010001111111110010010010010110010010011010010010011111001001010011001001010101001001010111001001011001001001011011100
100101110100100101111100100110010100100110011100100110101100100110110100100110111100100111001100100111010100100111011100100111101100100111110100100111111100
1001010100101001011100101001101100101001110100101001111100101010011100101010101100101010110100101010111100101011001100101011010100101011011100101011101100101011101100101010
1111010010101111110010110010110011010010110011110010110100110010110101010010110101110010110110110010110111010010110111110010111001110010111010110010111011010010111011010100
1011101111100101111001100101111010100101111011100101111101100101111110100101111111100110011001101010011001101110011001110110011001111010011001111110011010011010011010011101
001111100110101010100110101011100110101101100110101110100110101111100110110011100110110101100110110110100110110111100110111010100110111011100110111101100110111110100110111
1101001101111111001110011101001110011111001110101011001110101101001110101111001110110101001110110111001110111011001110111101001110111111001111001111010101010011
1101011100111101101100111101110100111101111100111110101100111110110100111110111100111111010100111111011100111111101100111111110100111111111101010101010101110101
0101101101010101011111010101101011010101101011110101011101111010101110111101010111101101010111111110101101101011110101101101110101101110110101101111110101110101110101110111110111111
010111101111101011111011110101111110110101111111111101101101101111101101110111101101111011110111011011111111011101110111111101110111111101111101111111111111 00000000000

Importance of a large enough key space V

Lessons learned:

- Do not use a bruteforceable key space
- Require a preamble/sync word during transmission
- Utilize rolling codes

- Samy Kamkar invented this attack, named it OpenSesame and even built a fully functioning prototype out of a kid’s toy. If you are interested into such content, check his videos.

Your bike is gone now.
It is winter anyways. We should get a car.

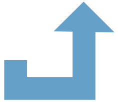
Five Step Plan towards your new Car

Only for *illustrative* purposes

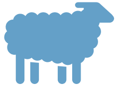
Find your desired KeeLoq protected car model and rent it

Read out the car key with physical access (10 – 30 power traces)

Read out manufacturer key (phys. access the receiver HW in the car & cloned transmitter)



Return the rental car



Clone a car key without physical access (intercept only two messages from the original key)

Come on but what’s with more high-tech cars …

… maybe like

Tesla Model S

broken in 2019

Tesla Model X

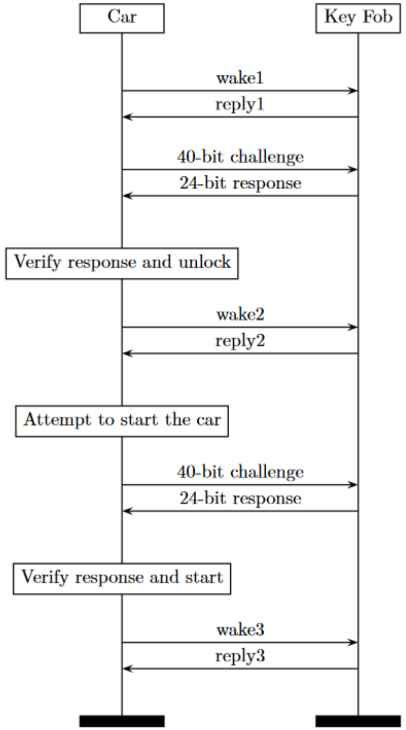
broken in 2020

Image sources: http://tesla.com
https://nieuws.kuleuven.be/en/content/2020/ku-leuven-researchers-demonstrate-serious-flaws-in-tesla-model-x-keyless-entry-system
https://nieuws.kuleuven.be/en/content/2018/security-flaws-leave-keyless-tesla-cars-vulnerable-to-theft
Model S Paper: https://tches.iacr.org/index.php/TCHES/article/view/8289/7639

Having a look at the Tesla Model S

What pitfalls did Tesla step into?

Think with your neighbors for a few minutes.



Model S Paper: https://tches.iacr.org/index.php/TCHES/article/view/8289/7639

What could Tesla have done better?

Proprietary Cipher (DST40) with small bit length (Trade Secret of Texas Instruments until 2005)

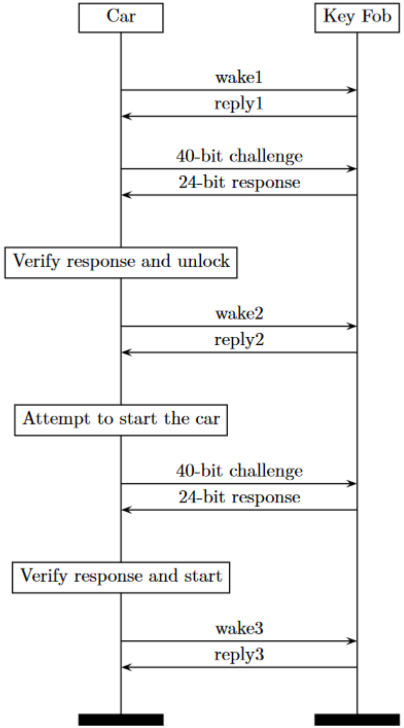
Car Identifier (two bytes) can be brute-forced

No mutual authentication of key fob and car

No DoS protection inside the car (theoretically brute-forcing the fob key would be possible)

Reverse engineering of the key fob firmware

(no security fuse)



Attack Strategy on Tesla Model S

Phase 0:

Identify the car

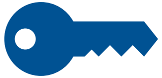
Record one wake frame and extract identifier.

Phase 1:

Impersonate the car

Send two challenges to the legitimate key fob.

Challenge values are carefully chosen.

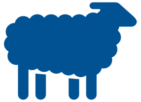


Phase 2:

Key Recovery

Grouping of all keys producing the same response to a fixed challenge.

Then brute force the subset of possible keys.



Phase 3:

Impersonate the key fob

Mimic the protocol using the victim's key.

Take Away Notes on Security

Robust security is becoming more critical in everyday-use products

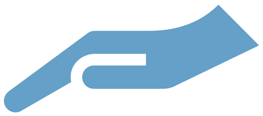
- Stark rise in use of embedded systems
- High number of people with access to these systems
- Valuable information stored within or transmitted by them

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

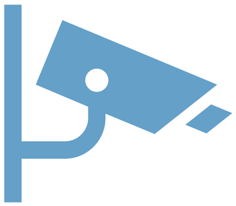
Side Note – Safety is not equal to Security

… 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)

Introduction to Smart Cards

(covering the basics)

Smart Card

Definition

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

Typical Use Cases

- Secure Data Storage
- Secure Data Processing
- Authentication
- Encryption & decryption

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

Smart Card

Hardware Components

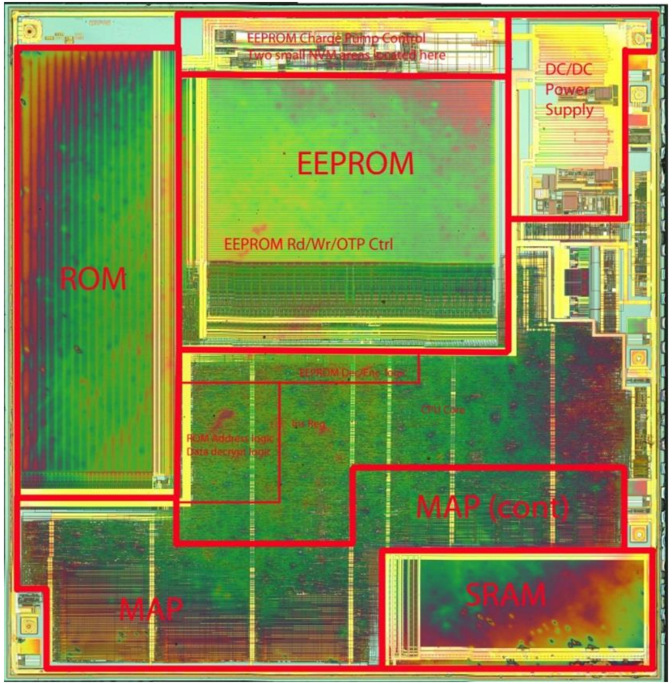


Image source: http://www.flylogic.net

Non-Volatile Memory (EEPROM, ROM)

Volatile Memory (SRAM)

Crypto Functions

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

Analog Components

- Voltage regulators
- Anti-tamper sensors

Attacks on Smart Cards

Cryptographic Algorithms

Cryptographic Protocols

Logical attacks

Hardware attacks

Attack type
- Active
- Passive

Access
- Non-invasive
- Semi-invasive
- Invasive

Smart Card Applications

Terminal Firmware

Software attacks

Attacks on Smart Cards

Cryptographic Algorithms

Cryptographic Protocols

Logical attacks

Smart Card Applications

Terminal Firmware

Software attacks

Hardware attacks

Attack type

- Active
- Passive

Access

- Non-invasive
- Semi-invasive
- Invasive

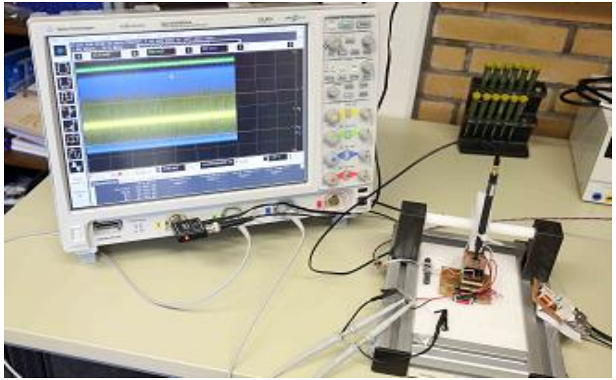
Exploiting Passive Side Channels on Smart Cards

Advantages for an attacker

- Non-invasive or Semi-invasive
- Passive is non destructive
- Relatively low-cost
- Powerful and relatively fast

Possible Side-Channels

- Timing
- *Power consumption*
- EM emission
- others…



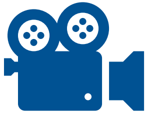
We will be covering side channel attack basics focusing on the power side channel in the next lecture.

Laboratory Objectives

(what you have to do in this lecture)

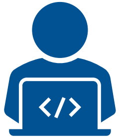
Laboratory Scenario

Scenario is a pay TV system



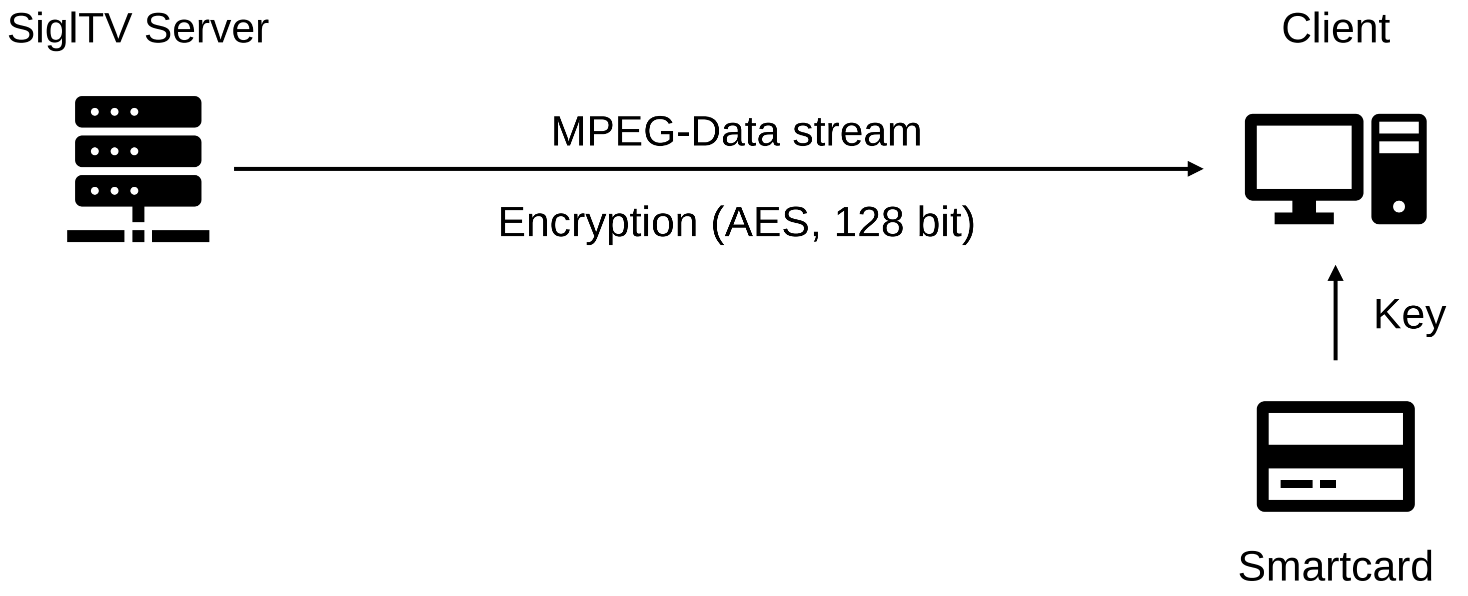
A Smart Card decrypts a video stream

You, the students, take the role of the attacker to compromise the system by cloning the Smart Card



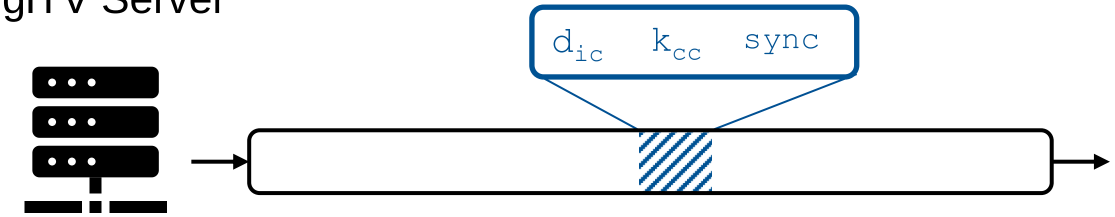
You also take the role of the developer to secure the Smart Card against the attacker

Structure of the Pay-TV System

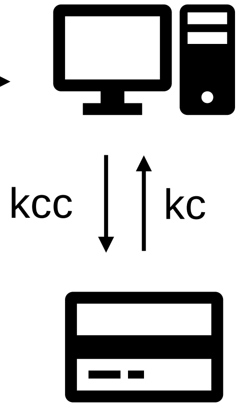


Structure of the Pay-TV System

SigITV Server



Client



$d_i = AES^{-1}(k_c, d_{ic})$

$k_c = AES^{-1}(k_m, k_{cc})$

Smartcard

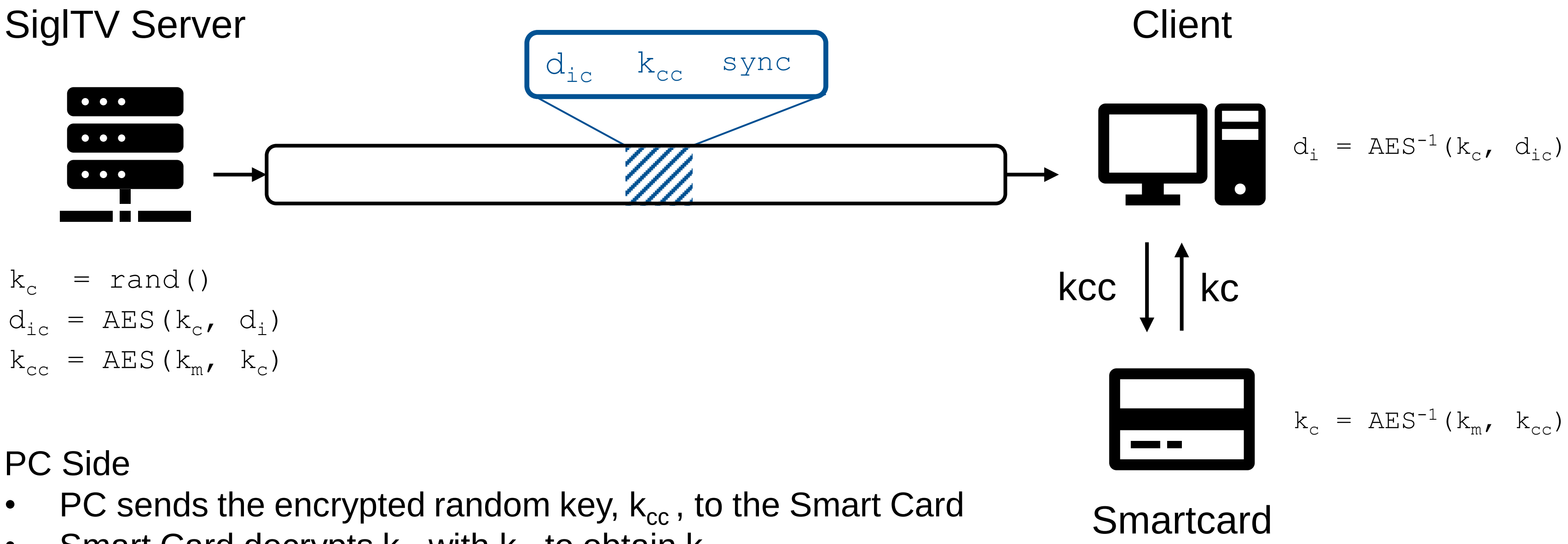
$k_c = rand()$
$d_{ic} = AES(k_c, d_i)$
$k_{cc} = AES(k_m, k_c)$

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 Pay-TV System



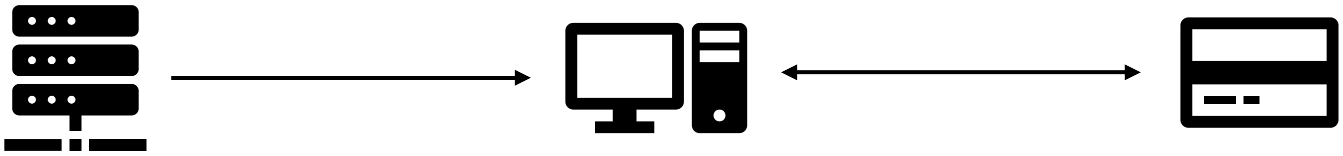
$k_c = rand()$
$d_{ic} = AES(k_c, d_i)$
$k_{cc} = AES(k_m, k_c)$

PC Side

- PC sends the encrypted random key, k_{cc} , to the Smart Card
- Smart Card decrypts k_{cc} with k_m to obtain k_c
- PC uses k_c to decrypt d_{ic}
- PC displays the (plaintext) video chunk d_i

Software Structure of the Pay-TV System

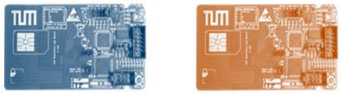
SiglTV Server | Client | Smartcard



Python Streamer

Python Client
(Decryption routines &
Smart Card interfacing logic)

Smart Card Emulator
(STM31F051)
Reference card with
preloaded master key
Blank Card



Laboratory Work Phases

Phase 1

Theoretical background

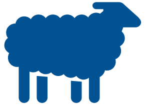
ISO7816, AES, DPA Basics

Microcontroller Target, Debugging, Logic Analyzer, HDF5, numpy

Phase 2

Attack

Implement a DPA, validate on sample traces. Perform measurements on reference card

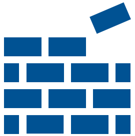


Phase 3

Clone

Implement a basic Smart Card OS. Implement and test AES decryption

Laboratory Work Phases



Phase 4

Harden your implementation

Improve DPA to attack hiding countermeasures

Implement hiding countermeasures yourself and compare with reference.

Phase 5

Harden your implementation II

Try to break masked implementations using your implemented first order DPA, when it works, note why.

Implement masking yourself

Phase 6

Write Report

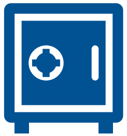
Write a four-to-five pages report about your lab learnings and results.

Takeway Laboratory Learning Objectives

The four main learning objectives

Analyze the tradeoffs between different secure implementations and their cost

Understand the communication protocol between the Smart Card and the PC



Understand the internals of the AES cipher

Understand, implement and improve your own DPA attack

Work Plan

Intermediate Presentation

What is expected for the **intermediate 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?

- Smart card clone
  - ISO UART (implementation, sampling strategy)
  - Size of the complete smart card OS and AES implementation
  - Speed of your AES implementation (compare against reference card)
  - Optimization of your card implementation either for speed, code size or memory

- Distribution of Tasks in your team

Final Presentation

What is expected for 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

Lab Report Contents

Aka what I would like to see in your lab report:

The performance of your DPA and roughly how you implemented it (custom functions, optimizations, …)

The workings of your unprotected smart card OS, state machine graph, your program size and speed of all implementations plus comparison it to the reference card.

Which aspects of the DPA you improved and the new runtime (include your PC specs). If you have special improvements to attack specific countermeasures, please elaborate. How do the new techniques compare against the simple DPA?

In regard to countermeasures, illustrate which ones and how you implemented them. What prerequisites do some countermeasures have and how does their implementation influence your smart card OS (impact in speed, size). Compare this to your base OS.

Include a gantt chart or comparable of your tasks and mark which parts who did in a small section. This will influence questions in the oral exam.

Thank you for your Attention

(any questions so far?)

Project Management

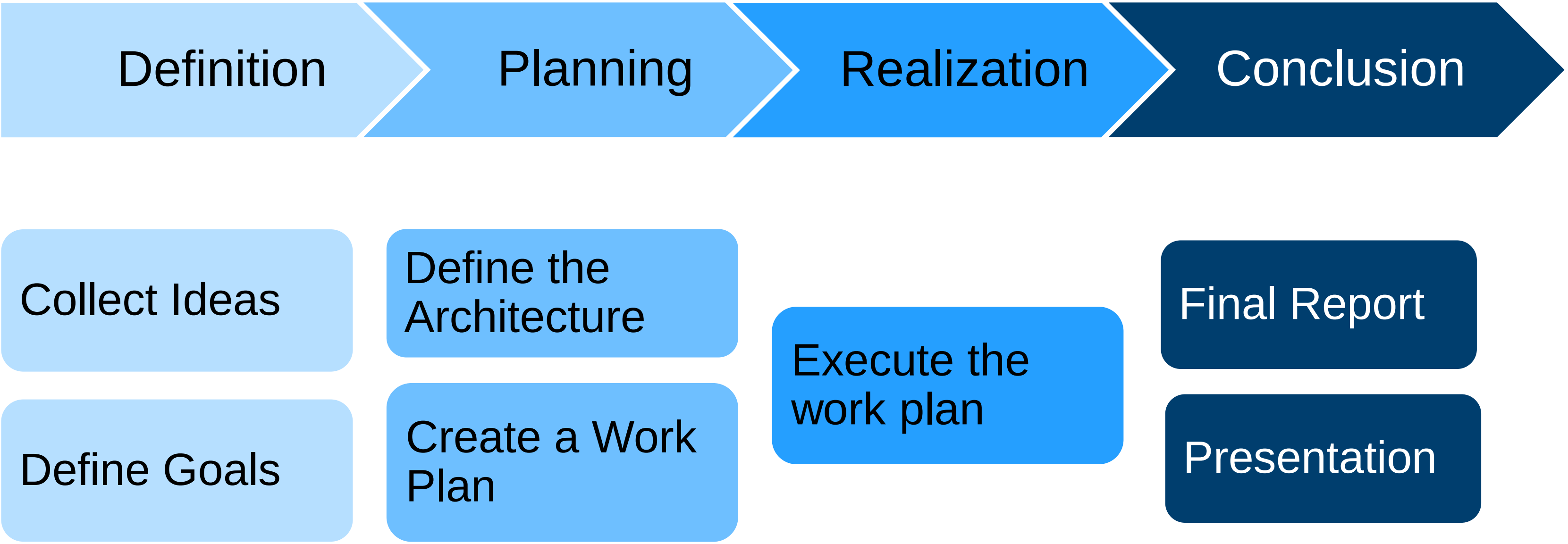
(I bet you did not see that coming)

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



Phase 1 – Definition

Definition | Planning | Realization | Conclusion

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

Definition | **Planning** | Realization | Conclusion

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

Definition › Planning › **Realization** › Conclusion

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

Definition | Planning | Realization | **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?