scademy
secure coding academy

Crypto chip-set security

CL-CHS | Onsite / Virtual classroom | 3 days

Audience: Developers, architect and testers of secure hardware components
Preparedness: Professional
Exercises: No

The biggest challenge for professionals working on design and development of crypto chip-sets is to be continuously up-to-date regarding the attack methods and their mitigation. Serving them, this course explains various physical and logical attacks on security chips, possible countermeasures and best practices.

Regarding physical attacks, the passive attacks are detailed through optical reverse engineering and various side channel analysis methods, while active attacks are discussed with special emphasis on fault injection, Focused Ion Beams and hardware Trojans. The very powerful passive and active combined attack (PACA) type is introduced through the practical example of RSA implementations. Discussion of logical attacks not only covers practical attacks against specific cryptographic algorithm implementations, but also the relevant programming bugs and mitigation techniques like buffer overflow or integer problems are introduced.

Finally, a set of guidelines is assembled to follow by engineers working in this field, and the testing methods are presented that can help to find and avoid the discussed security flaws and vulnerabilities.

Outline:

IT security and secure coding
Requirements of secure communication
Practical cryptography
Security protocols
Simple physical attacks and protections
Passive physical attacks
Active physical attacks
Passive and active combined attacks
Special security functions – Requirements and solutions
Principles of security and secure coding

Participants attending this course will:

- Understand basic concepts of security, IT security and secure coding
- Have a practical understanding of cryptography
- Understand the requirements of secure communication
- Understand essential security protocols
- Understand some recent attacks against cryptosystems

Detailed table of contents

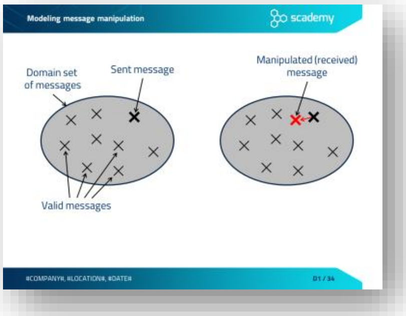
Day 1

IT security and secure coding

- Nature of security
- What is risk?
- IT security vs. secure coding
- Nature of security flaws

Requirements of secure communication

- Security levels
- Secure acknowledgment
  - Malicious message absorption
    - Feasibility of secure acknowledgment
    - The solution: Clearing Centers
  - Inadvertent message loss
- Integrity
  - Error detection - Inadvertent message distortion (noise)
    - Modeling message distortion
    - Error detection and correction codes
  - Authenticity - Malicious message manipulation
    - Modeling message manipulation..
    - Practical integrity protection (detection)
  - Non-repudiation
  - Summary
    - Detecting integrity violation
- Confidentiality
  - Model of encrypted communication
  - Encryption methods in practice
  - Strength of encryption algorithms
- Remote identification
  - Requirements of remote identification



Modeling message manipulation

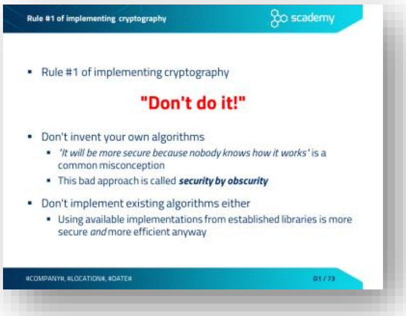
#COMPANY#, #LOCATION#, #DATE#

01 / 34

- Anonymity and traffic analysis
  - Model of anonymous communication
  - Traffic analysis
  - Theoretically strong protection against traffic analysis
  - Practical protection against traffic analysis
- Summary
  - Relationship between the requirements

Practical cryptography

- Rule #1 of implementing cryptography……………………………………………



Rule #1 of implementing cryptography

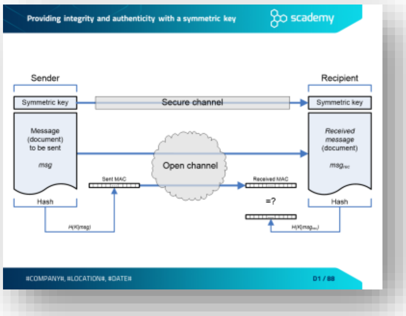
- Rule #1 of implementing cryptography

"Don't do it!"

- Don't invent your own algorithms
  - *'It will be more secure because nobody knows how it works'* is a common misconception
  - This bad approach is called ***security by obscurity***
- Don't implement existing algorithms either
  - Using available implementations from established libraries is more secure *and* more efficient anyway

#COMPANY#, #LOCATION#, #DATE# D1 / 73

- Cryptosystems
  - Elements of a cryptosystem
- Symmetric-key cryptography
  - Providing confidentiality with symmetric cryptography
  - Symmetric encryption algorithms
  - Modes of operation
- Other cryptographic algorithms
  - Hash or message digest
  - Hash algorithms
  - SHAttered
  - Message Authentication Code (MAC)
  - Providing integrity and authenticity with a symmetric key……………



Providing integrity and authenticity with a symmetric key

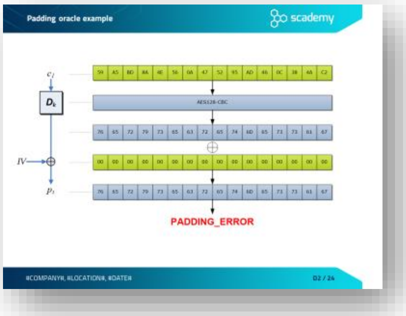
#COMPANY#, #LOCATION#, #DATE# D1 / 88

  - Random number generation
    - Random numbers and cryptography
    - Cryptographically-strong PRNGs
    - Hardware-based TRNGs
- Asymmetric (public-key) cryptography
  - Providing confidentiality with public-key encryption
  - Rule of thumb – possession of private key
  - Combining symmetric and asymmetric algorithms
- Public Key Infrastructure (PKI)
  - Man-in-the-Middle (MitM) attack
  - Digital certificates against MitM attack
  - Certificate Authorities in Public Key Infrastructure
  - X.509 digital certificate

Day 2

Security protocols

- The TLS protocol
  - SSL and TLS
  - Usage options
  - Security services of TLS
  - SSL/TLS handshake
- Protocol-level vulnerabilities
  - BEAST
  - FREAK
  - FREAK – attack against SSL/TLS
  - Logjam attack
- Padding oracle attacks
  - Adaptive chosen-ciphertext attacks
  - Padding oracle attack
  - CBC decryption
  - Padding oracle example……………………………………………………………………
  - Lucky Thirteen
  - POODLE



Padding oracle example

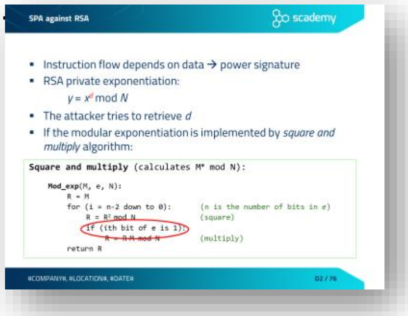
Simple physical attacks and protections

- Passive, active and PACA attacks
- Physical access to the chip
- Levels of invasiveness
- Protection principles

Passive physical attacks

- Reverse Engineering
  - Reverse engineering of integrated circuits
  - Decapsulation
  - Decapsulation in practice
  - Deprocessing
  - Removal of the passivation layer
  - Different lasers against different passivation layers
  - Optical reverse engineering
  - Requirements towards the optical microscope
  - Bonding to chip surface

- Physical protection of the chip
  - Multi-layering
  - Protective layer
  - Examples of protective shields
  - Planarization
  - Confusing attackers
  - Obfuscating the design
  - Shuffling the internal bus lines
  - Internal memory encryption
- Side channel analysis
  - Power analysis attacks
  - Process of a power analysis attack
  - Modeling power consumption
- Simple power analysis
  - Examples for simple power analysis
  - Example power consumption of DES encryption
  - SPA against RSA
  - Countermeasures against SPA



SPA against RSA

- Instruction flow depends on data → power signature
- RSA private exponentiation:

  $y = x^d \bmod N$
- The attacker tries to retrieve *d*
- If the modular exponentiation is implemented by *square and multiply* algorithm:

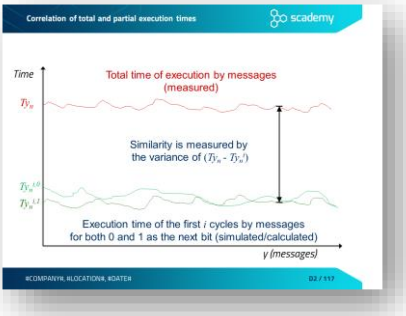
```
Square and multiply (calculates M^e mod N):

    Mod_exp(M, e, N):
        R = M
        for (i = n-2 down to 0):       (n is the number of bits in e)
            R = R^2 mod N              (square)
            if (ith bit of e is 1):
                R = R·M mod N          (multiply)
        return R
```

#COMPANY#, #LOCATION#, #DATE#

D2 / 76

- Differential power analysis
  - Process of a DPA attack
  - DPA attack – the difference function
  - DPA attack – visible correlation
  - DPA attack – spikes in differential traces
  - Influence of noise and measurement errors on DPA attack
  - DPA against DES
    - DES algorithm
    - Feistel function
    - DPA against DES
  - Improved DPA attacks
    - DPA enhancements
    - Correlation power analysis (CPA)
    - Process of CPA attack
- Protections against power analysis attacks
  - Protection measures against power analysis
  - Hiding techniques
  - Hiding: adding noise, desynchronization
  - Hiding: dual-rail precharge logic (DRP)
  - Hiding, DRP: sense amplifier based logic (SABL)

-
    - Hiding: current mode logic (CML)
    - Masking: masked dual rail precharge logic (MDPL)
    - Limitations of PA resisting logic styles
- Electromagnetic analysis
    - Electromagnetic emanation
    - Simple electromagnetic analysis (SEMA)
    - Electromagnetic analysis countermeasures
- RSA timing attack
    - Implementation of encoding/decoding in RSA
    - Fast exponentiation
    - Differences in execution times
    - RSA timing attack
    - Measurements
    - RSA timing attack – principles
    - Correlation of total and partial execution times
    - RSA timing attack – in practice
    - Example – RSA measurements
    - Example – Partial execution times are calculated for first i bits
    - Example – Candidates are ordered according to the variance
    - The RSA timing attack algorithm
    - Practical exploitation using the RSA timing attack
    - Attacking SSL/TLS servers
    - Protection against timing attacks
        - Hiding: RSA timing attack countermeasures
        - Masking: using blind signature
        - Real RSA implementations
        - Summary – lessons learnt from side channel attacks
        - Security evaluation of side-channel protections



Correlation of total and partial execution times

#COMPANY#, #LOCATION#, #DATE#

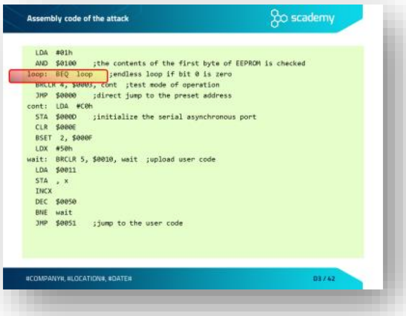
D2 / 117

Day 3

Active physical attacks

- Manipulating the circuit layout
    - Focused ion beam (FIB) workstation
    - Milling and cutting using FIB
    - Creation of test points
    - FIB accessibility

- Hardware trojan horses
  - Hardware trojan horse (HTH) risk factors
  - Taxonomy of trojans
  - Physical characteristics
  - Prevention – dead space removal
  - Design facilitated logic testing
  - Transparent mode
  - IC fingerprint / transient power-based analysis
  - IC fingerprint / timing-based analysis
  - Detection and correction using DEFENSE methodology
- Fault injection attacks
  - Fault injection attacks (FIA)
  - Fault injection models
  - Practical attacks
  - Microprobing
  - Microprobe station parts
  - Case study – reading the memory using chip surgery
  - Light, X-Ray, electromagnetic radiation
  - UV light attack – resetting the fuse
  - UV attack – locating the fuse
  - Effect of UV light on EEPROM and floating gate devices
  - Optical fault injection – white light
  - Pulsed laser
  - Tampering with temperature – data remanence
  - Tampering with temperature – cooling
  - Localized heating
  - Clock glitch attack
  - Execution of the attack
  - A sample attack against a Motorola microcontroller
  - Assembly code of the attack
  - Power glitching
  - Power glitch examples – attack on Motorola microcontroller
  - Power glitch examples – microchip PIC16F84 microcontroller
- Fault injection into RSA with CRT optimization
  - Chinese Remainder Theorem
  - RSA using CRT – mathematical background
  - RSA using CRT – the algorithm



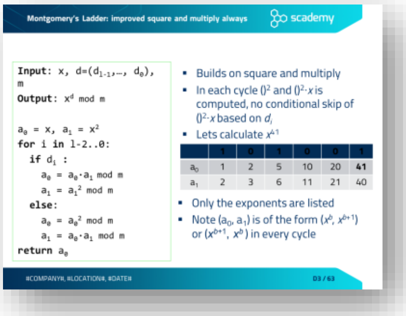
Assembly code of the attack

```
  LDA  #01h
  AND  $0100    ;the contents of the first byte of EEPROM is checked
loop:  BEQ  loop    ;endless loop if bit 0 is zero
  BRCLR 4, $0003, cont  ;test mode of operation
  JMP  $0000    ;direct jump to the preset address
cont:  LDA  #C0h
  STA  $000D    ;initialize the serial asynchronous port
  CLR  $000E
  BSET  2, $000F
  LDX  #50h
wait:  BRCLR 5, $0010, wait  ;upload user code
  LDA  $0011
  STA  , x
  INCX
  DEC  $0050
  BNE  wait
  JMP  $0051    ;jump to the user code
```

    - Bellcore attack on RSA with CRT
    - Bellcore attack algorithm
    - Lenstra attack on RSA with CRT
    - Non permanent bit-flip attack on RSA without CRT
    - RSA non permanent bit-flip attack
    - Detection of fault injection attacks

Passive and active combined attacks

- PACA against RSA with blinding and doublechecking
- SPA/DPA resistant RSA blinding
- Power traces before and after fault injection
- Fault injection – word by word multiplication
- PACA resistant RSA with CRT and Montgomery's Ladder
- Montgomery's Ladder: improved square and multiply always ....



Montgomery's Ladder: improved square and multiply always

```
Input: x, d=(d_{l-1},…, d_0),
m
Output: x^d mod m

a_0 = x, a_1 = x^2
for i in l-2..0:
  if d_i :
    a_0 = a_0·a_1 mod m
    a_1 = a_1^2 mod m
  else:
    a_0 = a_0^2 mod m
    a_1 = a_0·a_1 mod m
return a_0
```

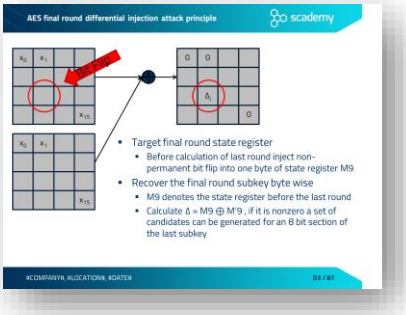
- Builds on square and multiply
- In each cycle $()^2$ and $()^2 \cdot x$ is computed, no conditional skip of $()^2 \cdot x$ based on d_i
- Lets calculate x^{41}

	1	0	1	0	0	1
a_0	1	2	5	10	20	**41**
a_1	2	3	6	11	21	40

- Only the exponents are listed
- Note (a_0, a_1) is of the form (x^b, x^{b+1}) or (x^{b+1}, x^b) in every cycle

#COMPANY#, #LOCATION#, #DATE# 03 / 63

- FA-SPA resistant modular exponentiation
- FA-SPA resistant RSA with CRT
- PACA protection – detect and derive
- DnD example – protecting the secret key
- Fault injection into ECC
    - Elliptic Curve Cryptography (ECC)
    - Elliptic curves
    - Addition and multiplication operations over elliptic curves
    - Discrete Logarithm Problem (DLP)
    - ECC fault injection attack
    - ECC attack method
- DES final round attack
    - DES algorithm
    - Feistel function
    - DES final round with permutation compensation
    - DES final round attack principle
    - DES final round attack algorithm
    - Completing the DES attack
- AES fault injection attacks
    - Advanced Encryption Standard (AES)
    - AES operation
    - AES algorithm
    - AES final round differential injection attack principle ....



AES final round differential injection attack principle

- Target final round state register
    - Before calculation of last round inject non-permanent bit flip into one byte of state register M9
- Recover the final round subkey byte wise
    - M9 denotes the state register before the last round
    - Calculate Δ = M9 ⊕ M'9 , if it is nonzero a set of candidates can be generated for an 8 bit section of the last subkey

#COMPANY#, #LOCATION#, #DATE# 03 / 87

    - Giraud's differential fault analysis attack

    - Blömer-Seifert fault attack on AES-128
    - AES pre-whitening step attack principle
    - Blömer-Seifert fault attack on AES
    - Assessment of the Blömer-Seifert fault attack on AES

Special security functions – Requirements and solutions

- Physically unclonable functions
    - Main PUF properties
    - PUF types
    - Secret key generation with PUFs
    - Further PUF applications
- JTAG attacks and protections
    - JTAG overview
    - JTAG based attacks
    - Defending against JTAG attacks
- Miscellaneous protection measures
    - Keeping the memory programming method secret
    - Unique chip ID
    - Bus scrambling
    - Sensors
    - Further design guidelines

Principles of security and secure coding

- Matt Bishop's principles of robust programming
- The security principles of Saltzer and Schroeder