This work is distributed under the CC BY-SA license, unless stated otherwise.

Hardware Security

1. Intro & Course overview & Cryptography

Ján Jančár

Summer 2026

(starting at :15)

Who am I?

</> neuromancer.sk/about

- Postdoctoral researcher in applied cryptography
- Side-channel attacks
- Elliptic curve cryptography
- Usable security
- Constant-time programming
- Very new to UPB!

My story with secure hardware



Zraniteľnosti čipových kariet DPMŽ

Ján Jančár Adam Král

18. januára 2017

Abstrakt

V tejto práci sa zaoberáme elektronickým systémom hromadnej dopravy Dopravného podniku mesta Žiliny, konkrétne ním používanými kartami typu MIFARE Classic. Tieto karty sú zastarané a zraniteľné voči niekoľkým útokom, ktoré umožňujú získať z karty jej kľúče a následne modifikovať jej celý obsah. Toto využívame získaním kľúčov z karty DPMŽ, následným reverzným inžinierstvom dát z karty a ich modifikáciou. Poukazujeme takisto na viaceré možnosti zneužitia tejto zraniteľnosti a simulujeme dobitie a používanie karty zadarmo, bez dobitia dopravným podnikom.

- 2016: First year at university
- Saw a CCC 2007 talk on breaking Mifare Classic
- DPMŽ uses Mifare Classic (public transport company)
- Together with a friend from high-school:
  1. Get a reader
  2. Run the attack tool
  3. Reverse-engineer the card data format
  4. Design attacks (free-riding, blocking cards, ...)
  5. ...
  6. Disclose to the company
- They mitigated the issues somewhat
- Then replaced the cards years later

Fast-forward 10 years and here I am giving this lecture.

Lectures and tutorials

- Lecture:
  - Slides available
  - Recordings available
- Tutorial:
  - Hands-on exercises (with hardware!)
  - Only when I am present
  - Remotely: tutorial makes little sense, Q&A on Zoom
- I will be present roughly 1/3 of lectures & tutorials

Homework assignments and exam

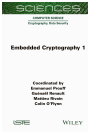
- Homework assignments
  - 3 assignments
  - Roughly 2 weeks deadline
  - Course achievement: at least 30% of points
- Exam
  - Written exam
  - Sample questions to practice: end of semester
- LLM usage policy
  - Do whatever (UPB guidelines, Wiki overview)
  - However, **you** should learn something
  - **You** will have to do the exam
  - False sense of knowledge
    - During the semester 
    - At the exam 

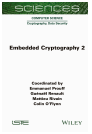
Materials

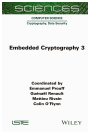












Excerpts from the following:

- The Hardware Hacking Handbook
- Power Analysis Attacks: Revealing the Secrets of Smart Cards
- Smart Card Handbook
- Embedded Cryptography 1
- Embedded Cryptography 2
- Embedded Cryptography 3

All available from the University library.

Contact & questions

- Ask during the lecture
  - Raise your hand, get my attention
- Email me: jan.jancar@upb.de

Questions

Any questions 

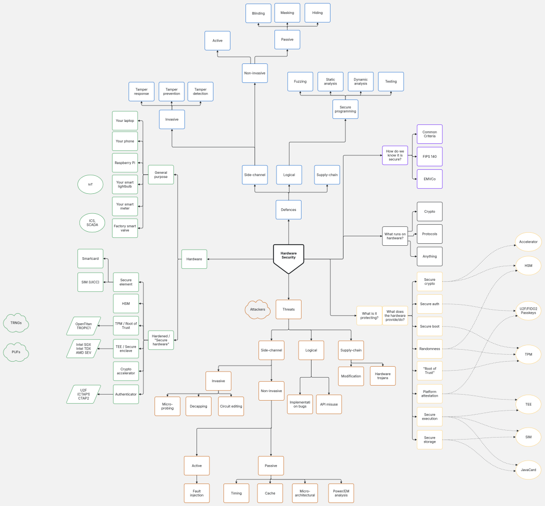
This course

- Is new!
- Your questions, interest and feedback will shape it
- Is based on material from my ex-colleagues:
  - Lukasz Chmielewski, Petr Svenda, Vashek Matyas
  - Thanks 🙏

This course (roughly)

3 parts

- Hardware
- Attacks
- Defences



This course (roughly)

1. Intro & Course overview & Cryptography
2. Embedded and secure hardware
3. Smartcards
4. Authenticators
5. Trusted Platform Modules, Trusted boot
6. Hardware Security Modules
7. Trusted Execution Environments
8. Disk encryption
9. Timing, cache and micro-architectural attacks
10. Side-channel attacks 1 (passive)
11. Side-channel attacks 2 (passive)
12. Side-channel attack defences (passive)
13. Side-channel attacks (active)
14. Side-channel attack defences (active)
15. Evaluation and Certification + Recap

Goals

- How hardware underpins security
- Why use secure hardware
- How to use secure hardware
- What hardware can and can’t do for you
- How hardware breaks
- How to protect hardware

Connections

- L.079.05638: Introduction to Cryptography
  - Focus on cryptography
- L.079.05732: Post-quantum Cryptography
  - Focus on post-quantum cryptography
- L.079.05704: Real World Crypto Engineering
  - Focus on more high-level applications of cryptography

Who are you?

- Passed courses?
  - L.079.05638: Introduction to Cryptography
  - L.079.05732: Post-quantum Cryptography
  - L.079.05704: Real World Crypto Engineering
- Programmed in C? Java? Python?
- Used cryptography libraries?
  - What an S-box is in AES?
  - What modular exponentiation is in RSA?
  - What scalar multiplication is in ECC?
- Heard of secure hardware?
  - Smartcards, Secure Elements, TPMs, HSMs, TEEs
- Played around with an embedded device?

Some notions

- Trusted vs Untrusted
  - **Trusted**: *A component whose failure may break a security property of a system.*
  - **Untrusted**: *A component on which we have no assumptions; we cannot trust it for any security properties.*
- Trustworthy
  - *A component where we have reasons to trust its security properties.*
- Trusted Computing Base (TCB)
  - *The set of all hardware, firmware, and/or software components that are critical to the security of a system.*
  - A vulnerability inside TCB may break security properties of the entire system

Some notions (cont.)

- What is the TCB of UPB employee access tokens?
  - Token HW + SW (if any)
  - Door reader HW + SW
  - Any central access database?
- Turns out it is broken
- Keep it small and simple!

Hardware

- General purpose hardware
  - Your laptop
  - Your phone
  - Raspberry Pi
  - Your smart lightbulb (IoT)
  - Your smart meter
  - Some factory's smart valve (ICS/SCADA)

- "Secure hardware" (vague category)
  - Secure element / Smartcard
  - Hardware Security Module (HSM)
  - Trusted Platform Module (TPM)
  - Trusted Execution Environment (TEE)
  - Cryptographic accelerator
  - Authenticator (U2F, FIDO2)

Why secure hardware?

- General purpose hardware
  - Complex
  - Large trusted computing base (TCB)
  - Mixes trust domains
  - Expensive
  - Easy to break with physical access
  - **Threat model**: remote attacker vs. “good” owner
- “Secure hardware”
  - Cheap, portable
  - Small TCB
  - Often single trust-domain
  - Physically separate
  - Hard to break even with physical access
  - **Threat model**: physical attacker (owner may be adversary, when ?)

Where is secure hardware used?

- Everywhere
- Let's count:
  1. Access token
  2. eID
  3. Bank cards
  4. U2F token
  5. SIM cards
  6. TEE (ARM TrustZone)
  7. TPM (Intel fTPM)
- What about you?

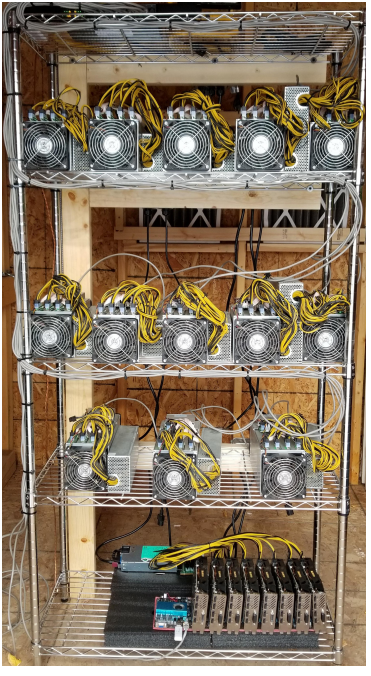
Side-channel attacks

- Hardware is physical
- There are many new threats when you account for this
- **Side-channels**
  - What is a side-channel?
  - Any unintended observable of a computation
  - Examples from real life?



https://www.spiegel.de/panorama/niederlande-polizei-haarlem-jagt-hanfplantagen-besitzer-im-schnee-a-1016867.html


copyright www.kavanileuven.nl



```
def checkPassword(password):
    correct = "hunter2"
    for i in range(len(correct)):
        if i >= len(password) or password[i] != correct[i]:
            return false
    return len(password) == len(correct)
```

Electro-magnetic leakage

- *ECDH Key-Extraction via Low-Bandwidth Electromagnetic Attacks on PCs*
- Stole a cryptographic key from electro-magnetic radiation of a laptop across a wall
- Single execution!

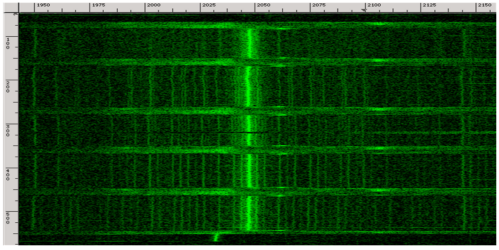


Figure 1: EM measurement (0.5 sec, 1.95-2.15 MHz) of five scalar-by-point multiplication operations using the NISTP-521 curve executed on a Lenovo 3000 N200 laptop. The scalar was overridden to be the 521-digit number obtained by repeating the pattern written to the right. In all cases, the curve point had the same random-looking x coordinate and a small y coordinate.

Sound leakage

- *RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis*
- Phone microphone next to a laptop is enough for key extraction
- Powerful microphone can extend the distance



Figure from a extract

Figure 6: Parabolic microphone (same as in Figure 5), attached to the portable measurement setup (in a padded briefcase), attacking a target laptop from a distance of 4 meters. Full key extraction is possible in this configuration and distance (see Section 5.4).

l 30 cm
ull key

Sound leakage (remotely)

- *Lend Me Your Ear: Passive Remote Physical Side Channels on PCs*
- Built-in microphone leaks information as well
- Demonstrated private-key recovery using a technique from my paper Minerva

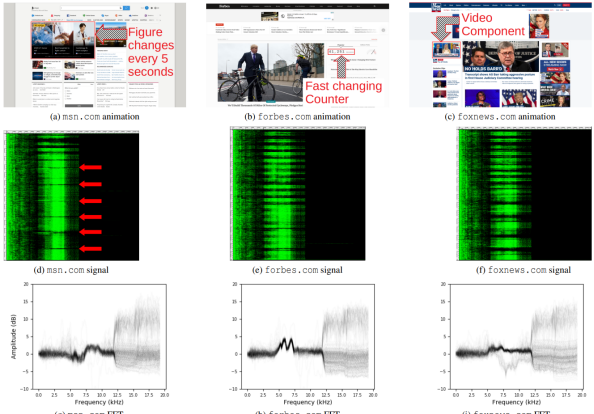
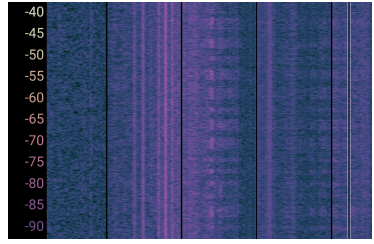


Figure 7: Attacker's signal when victim viewed `msn.com`, `forbes.com` and `foxnews.com`. For spectrograms, the vertical axis is time (40 sec), and the horizontal axis is frequency (0–24 kHz). The bottom figures show, for each website, the FFTs of all recordings in a single plot. We subtracted from the measurement the average value, over all websites, of each bin.

Side-channel demo

- You will have to be really quiet for this one 🤫
- Recently, I noticed my docking station makes a sound every time I scroll
- Investigated:
  - Mouse is Bluetooth-direct
  - Also works when scrolling on external display
  - Scrolling not necessary, flashing is enough
- **Demo time** 🤞



Supply chain security

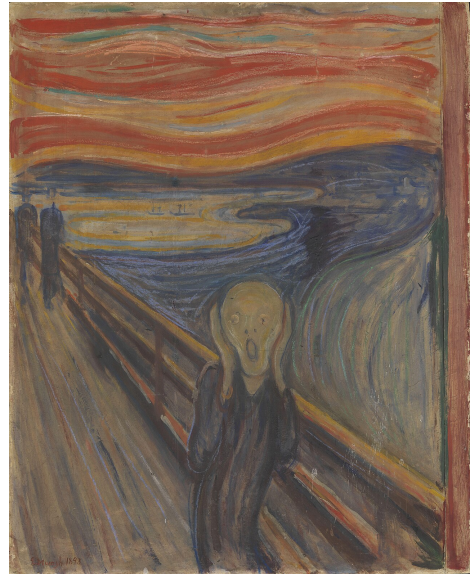
- **Tampering**
  - NSA's TAO will just intercept your shipment and backdoor it
  - Top secret
- **Threats**
  - **Interdiction**: intercepting and modifying hardware during delivery
  - **Hardware Trojan**: malicious modification at design or manufacturing stage



(TS//SI//NF) Left: Intercepted packages are opened carefully; Right: A "load station" implants a beacon

I have now probably scared you enough.

- Nothing is secure
- All of my Bitcoin is going to get stolen!1!
- My web browsing is not private while on a call?!?
- NSA will backdoor my router???



https://commons.wikimedia.org/wiki/File:Edvard_Munch,_1893,_The_Scream,_oil,_tempera_and_pastel_on_cardboard,_91_x_73_cm,_National_Gallery_of_Norway.jpg

Side-channel attack defences

How do we defend ?

- **Hardware level**
  - Add a sensor to detect tampering
  - Make it leak less (current flattening)
  - Add more noise (hiding)
- **Algorithm level**
  - Split secret and handle each piece separately (masking)
  - Add fresh randomness (blinding)
  - Constant-time programming

Secure hardware

- Various levels of capabilities
  - Fixed ID carrier: “RFID tag”
    - No secure channel
    - Easy cloning
  - Key material carrier
    - Key material loaded to PC/laptop for use
    - Not protected during use
  - Encryption/signing device
    - Keys never leave the device
    - Access must be secured (PIN unlock)
    - Limited performance

Secure hardware

- Various levels of capabilities (cont.)
  - Cryptographic accelerator
    - Keys never leave the device
    - High-speed, lots of cryptography
    - Expensive
  - Root of trust
    - Secure boot process
    - Remote attestation
  - Computational device
    - Complex logic in app on the device
    - Keys never leave the device
  - Execution environment
    - Separate execution environment (“Secure world”)
    - Remote attestation
    - Secure even against the kernel/hypervisor

Secure Element / Smartcard

- Many definitions of “secure element”
- We use a narrow one:
  - Is isolated: physically separate
  - Runs applets: separate applications
  - Has applet firewall: mutually untrusted applications
  - Offers a cryptographic API: built-in or plug-in
  - Has protections against side-channel attacks
  - Has protections against tampering: Prevention / Detection / Response

Secure Element / Smartcard (cont.)

- Several form factors:
  - Smartcards
  - SIM card (UICC)
  - Smart SD card
  - USB dongle
  - Implants
  - ...
- JavaCard platform
  - Dominant platform
  - Billions worldwide
  - Applets written in a subset of Java
- Use:
  - eID
  - Bank cards
  - Authenticators
- More in the next lecture



https://commons.wikimedia.org/wiki/File:JaCarta_collage_01.JPG

Hardware Security Module (HSM)

- “A beefy smartcard for a datacentre”
- Usually no applets but fixed cryptographic API (PKCS#11)
- High-performance
- Use:
  - Banking
  - Certification authorities
  - SSL/TLS



https://commons.wikimedia.org/wiki/File:NCipher_nShield_F3_Hardware_Security_Module.jpg

Trusted Platform Module (TPM)

- Smartcard-like
- No applets, fixed TPM API
- Root of trust:
  - For trusted boot
  - Trust has to start somewhere
- Examples:
  - Discrete TPM
  - Integrated TPM
  - Firmware TPM
- Use:
  - Trusted boot
  - Disk encryption (key storage)
  - Remote attestation



https://en.wikipedia.org/wiki/Trusted_Platform_Module#/media/File:Lenovo_N20_Chrome_-_motherboard_-_Infineon_SLB9655TT12-49890.jpg

Trusted Execution Environment (TEE)

- Separate “secure world” environment that runs on main CPU
- Runs *enclaves* or *virtual machines*
- Inaccessible from “normal world”
- Remote attestation
- Usually **do not consider physical attackers**!
- Examples:
  - ARM TrustZone
  - AMD PSP
  - AMD SEV-SNP
  - Intel SGX
  - Intel TDX

Authenticator (U2F / FIDO2)

- Only for authentication
- Originally a second factor: U2F
- Now can be passwordless: FIDO2 (passkeys)
- Examples:
  - YubiKey
  - SoloKey
- Use:
  - Authentication



https://commons.wikimedia.org/wiki/File:FIDO2_USB_token.png

Obscurity in secure hardware

- Secure hardware space
  - Is very secretive
  - Is very proprietary
  - Uses NDAs
  - Goes against the Kerckhoffs' principle
- Kerckhoffs' principle
  - A cryptosystem...
  - *... should not require secrecy, and it should not be a problem if it falls into enemy hands;*
- Researchers try to fight this
- Industry keeps doing this

Cryptography

What runs on secure hardware and what will be our target.

- *Cryptography turns a data protection problem into a key protection problem*
- Focus on **structural properties of implementations** that become attack targets

AES

Advanced Encryption Standard

- Block cipher
- Operates on 128-bit state, split into 4 × 4 bytes (grid)
- Operations on bytes

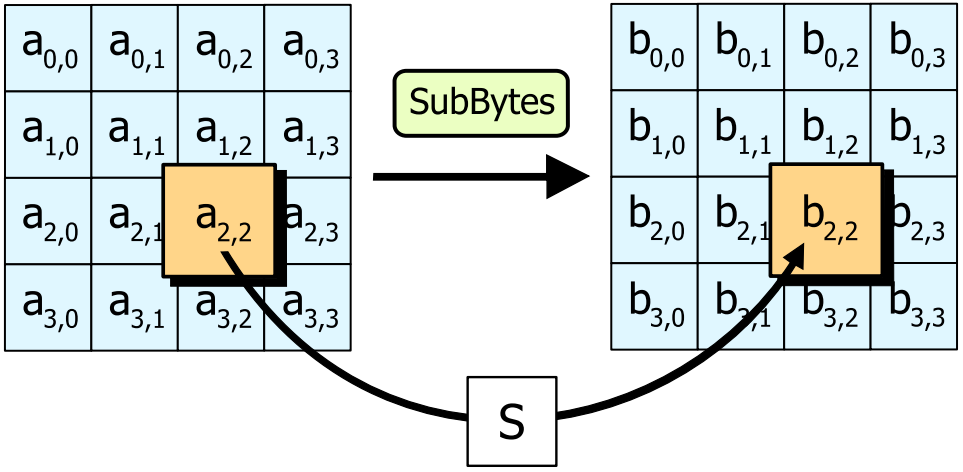
```
Procedure Encrypt(byte in[4 × 4], word w[4 × (N_r + 1)])
  byte state[4, 4] ← in
  AddRoundKey(state, w[0 : 4 – 1])

  for round ← 1 to N_r – 1 do
    SubBytes(state)
    ShiftRows(state)
    MixColumns(state)
    AddRoundKey(state, w[round × 4 : (round + 1) × 4 – 1])
  end for

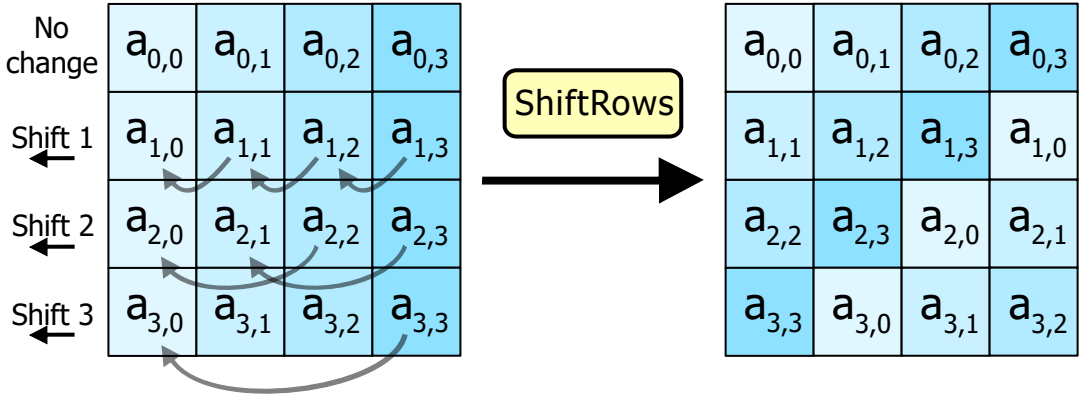
  SubBytes(state)
  ShiftRows(state)
  AddRoundKey(state, w[N_r × 4 : (N_r + 1) × 4 – 1])

  return state
End Procedure
```

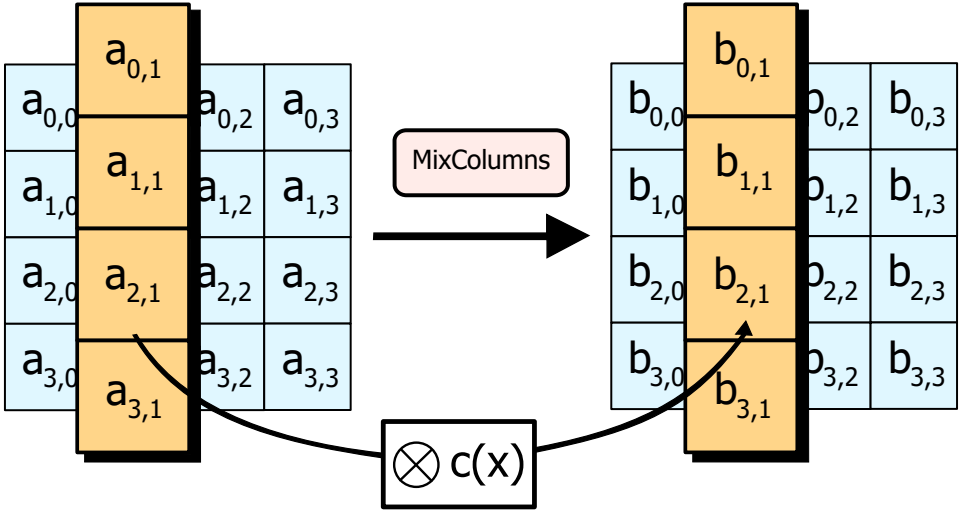
AES SubBytes



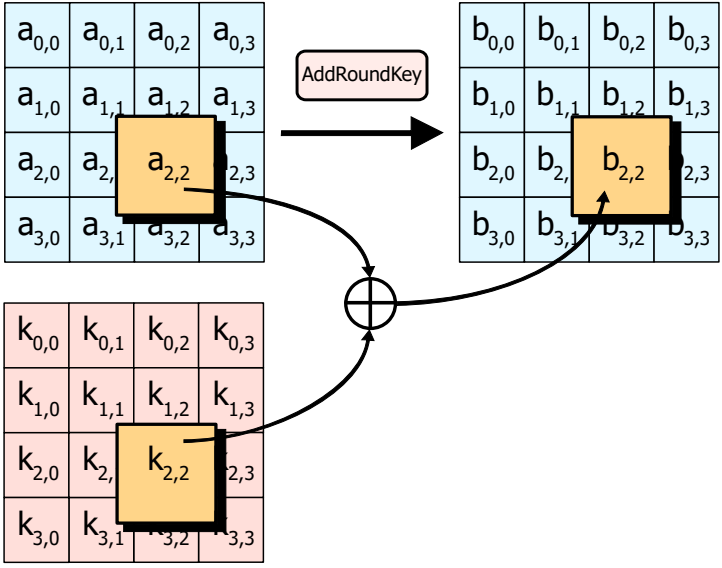
AES ShiftRows



AES MixColumns



AES AddRoundKey



AES from a hardware standpoint

- Operating on single bytes is inefficient
  - x86 likes 32 or 64 bits
- T-tables
  - Lookup the value in a large table
  - Bad: Cache attacks
- Bit-slicing
  - Pack 1 bit of 64 blocks into one value
  - Implement AES using bitwise logical operations
- Vectorization
  - Bit-sliced implementation can use vector instructions

RSA

Rivest Shamir Adleman

- Encryption and signatures
- Core operation: *modular exponentiation*
- $m^d \pmod{n}$
- $n = pq$

RSA from a hardware standpoint

- Elements are too large: 2048–4096 bits
  - x86 likes 32 or 64 bits
- Needs big integer and modular arithmetic
- RSA-CRT (Chinese Remainder Theorem)
  - Optimization to split exponentiation $\pmod{p}$ and $\pmod{q}$
  - Then recombine using CRT
- Modular exponentiation
  - Square-and-multiply (and many similar algorithms)
  - Directly handles secret key

ECC

Elliptic Curve Cryptography

- ECDH, ECDSA, …
- Core operation: *scalar multiplication*
- $[k]P = \sum_k P$
- $P = (x, y)$ is a point on an elliptic curve
- $x, y \in \mathbb{F}_p$

ECC from a hardware standpoint

- Large implementation “pyramid”
  - Big integers
  - Finite field (modular arithmetic)
  - Coordinate system + addition formulas
  - Scalar multiplication
- Scalar multiplication
  - Double-and-add (and many similar algorithms)
  - Directly handles the secret key

Recommended reading

- For **this** lecture
  - Read *Hardware Hacking Handbook, Chapter 1*
    - Intro to Hardware Security
- For the **next** lecture
  - Skim *Smart Card Handbook, Chapter 1*
    - Intro to Smartcards
  - Skim *Smart Card Handbook, Section 13.16.2*
    - JavaCard

Takeaways

- What this course will be about
- General idea of what secure hardware is
- Interest in hardware security
- Insights into implementations of AES, RSA, ECC

References

- David Oswald, Daehyun Strobel, Falk Schellenberg, Timo Kasper, and Christof Paar: *When Reverse-Engineering Meets Side-Channel Analysis – Digital Lockpicking in Practice*
- Daniel Genkin, Adi Shamir, and Eran Tromer: *RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis*
- Daniel Genkin, Lev Pachmanov, Itamar Pipman, and Eran Tromer: *ECDH Key-Extraction via Low-Bandwidth Electromagnetic Attacks on PCs*
- Daniel Genkin, Noam Nissan, Roei Schuster, and Eran Tromer: *Lend Me Your Ear: Passive Remote Physical Side Channels on PCs*
- Jan Jančár, Vladimír Sedláček, Petr Švenda, and Marek Sýs: *Minerva: The curse of ECDSA nonces*