TRUSTED COMPUTING

Material on TCG is based on slide material from Dries Schellekens
http://www.esat.kuleuven.be/cosic/seminars/slides/Trusted_Platforms.ppt

Special trusted computing devices

- Secure Cryptoprocessors
  - HSM: Dedicated microprocessor system with physical protection features
    - Tamper-detecting and tamper-evident containment.
    - Automatic zeroization of secrets in the event of tampering.
    - Chain of trust boot-loader which authenticates the operating system before loading it.
    - Chain of trust operating system which authenticates application software before loading it.
    - Hardware-based capability registers, implementing a one-way privilege separation model.
    - Possibly battery backup
  - Smart cards: payment cards, SIM (UICC) cards, access/ID cards
  - NFC

HSM (Hardware (or Host) Security Modules)

- Special Computers with high-grade protection with purpose to to store critical information and keys

- Some can be small – pci card/smartcard like
- Some can be large – desktop box like

- HSM in cloud environment:
  - Barbican

HSM trustworthiness

Security Certifications

- FIPS 140-2: Federal Information Processing Standard
- CC-EAL: Common Criteria Evaluation Assurance Level

HSM APIs

Frequently used

- **PKCS#11 (aka Cryptoki)**
- OpenSSL Engine
- Microsoft CAPI
- Java Cryptography Extension

SECURITY DEVICES

SMARTCARDS

RFID, NFC

SMARTCARDS

Parts of this material has been compiled from various open sources

In this lecture

- Cards of today
- Smartcard history
- Standards
- Hardware
- JavaCard
- Security issues
- Attacks on cards and crypto engines

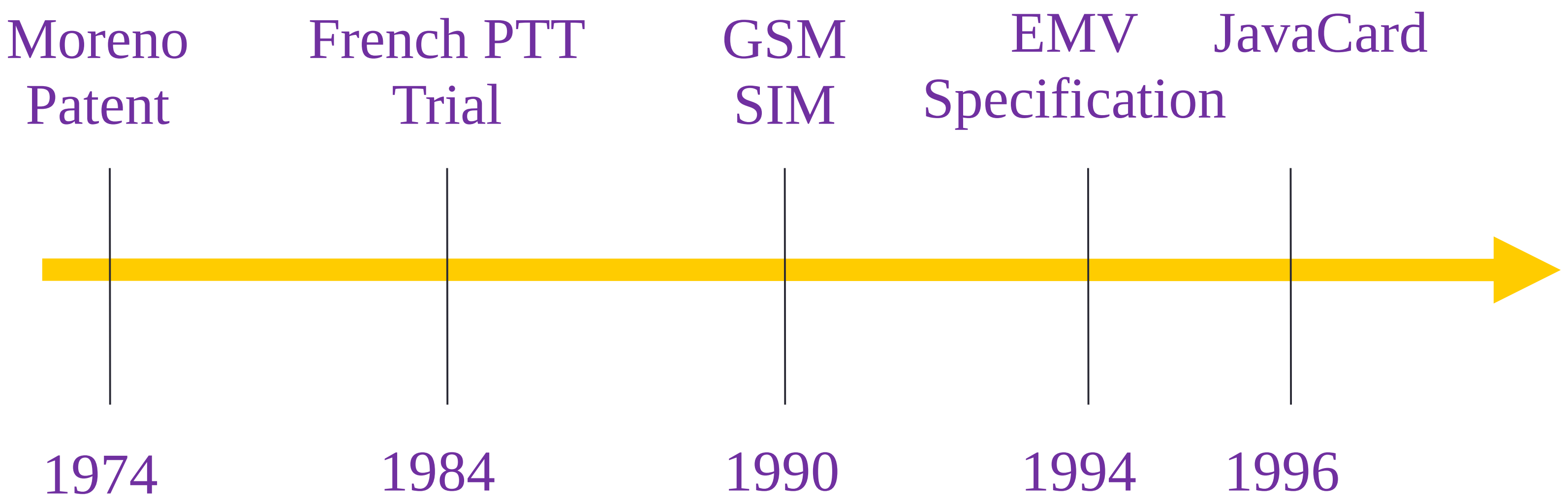
Cards of today

- Java cards
- SIM cards
- eCash cards
- Contact / Contactless Smart Cards
- Proximity cards
- Hybrid/twin cards
- Combi cards

History

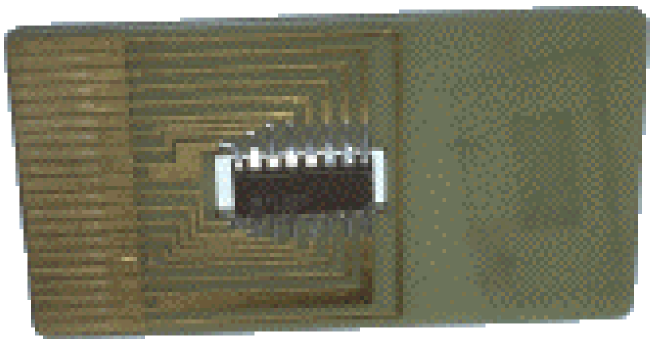
- Plastic cards :1950
- Magnetic Stripe Card
  - Very cheap to produce
  - Can store dynamic data
  - Easy to manipulate and copy (not all always!)
- Integrated Circuit Card (ICC): 1974
  - Cheap to produce (Semiconductor technology)
  - Can store dynamic data and can compute
  - Can be hardened against unauthorised manipulation

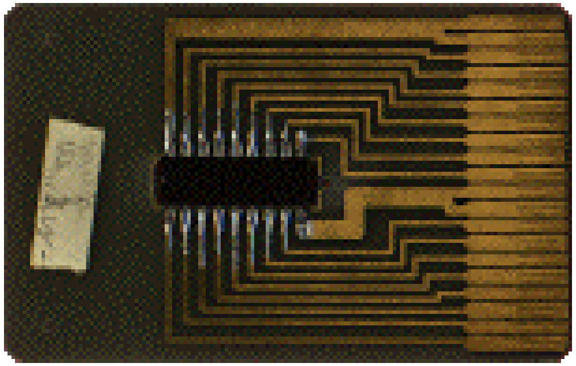
Some Milestones for ICC cards



History (1/6)

- 1974 - Roland Moreno invented a card with integrated circuit



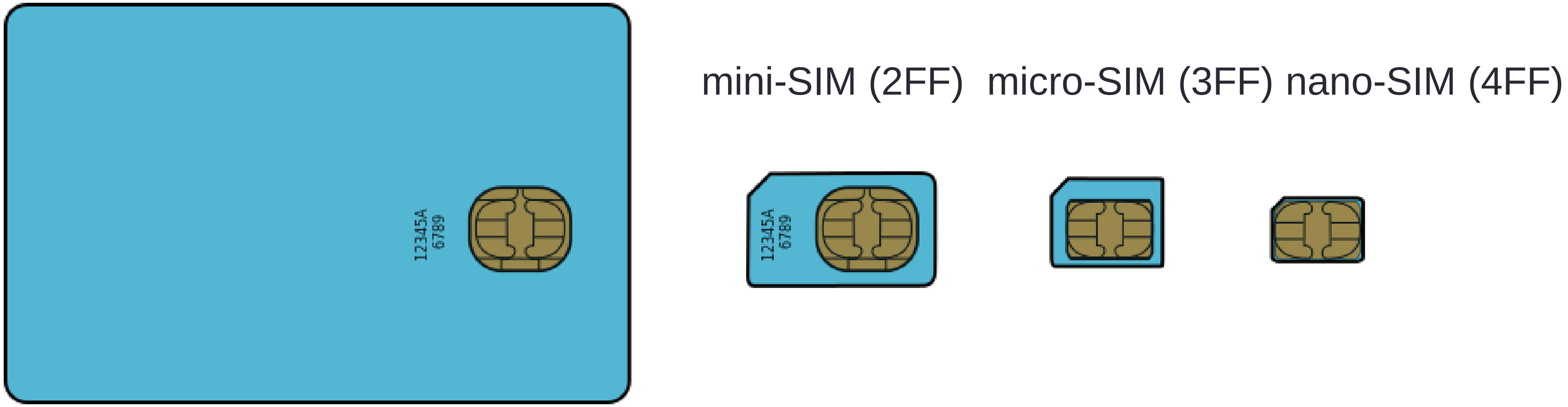


- 1979 - Release of the Bull CP8 card

History (4/6)

- 1991: The SIM card for GSM

  Original Full size of 1991 (1FF=1 Form Factor)



Picture: https://en.wikipedia.org/wiki/Subscriber_identity_module#/media/File:GSM_SIM_card_evolution.svg

- The first SIM card was developed by the German smart-card vendor Giesecke & Devrient.

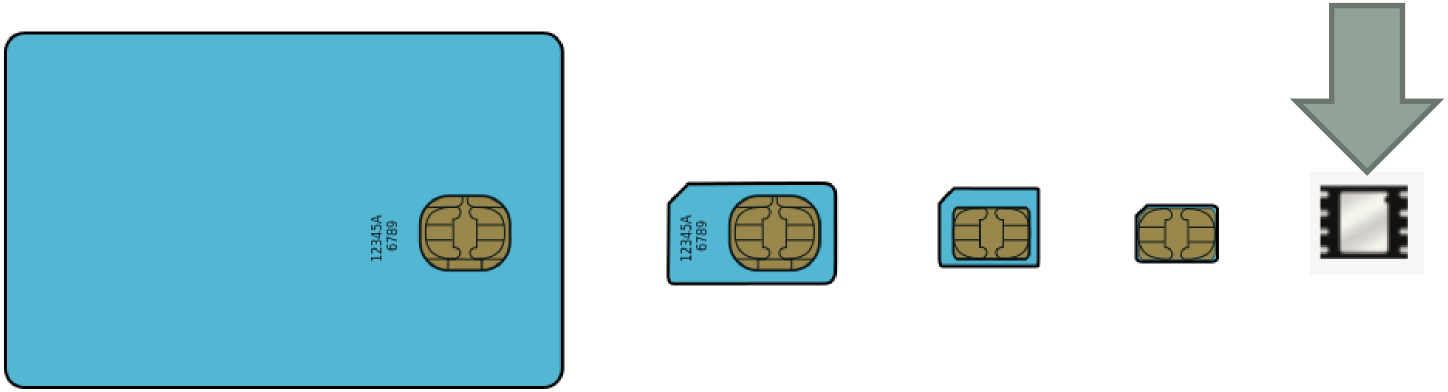
History (6/6)

- March 1999 - Version 2.1 JavaCard with :
  - The JavaCard 2.1 API Specification
  - The JavaCard 2.1 Runtime Environment Specification
  - The JavaCard 2.1 Virtual Machine Specification

Standardisation (3/5)

- ISO 7816-3 (standard)
  - Protocol for asynchronous d :half-duplex
- ETSI SCP (organisation)
  - Mainly sets the standards used for (U)SIMs

“SIM” for soldering
on device PCB



Picture: https://en.wikipedia.org/wiki/Subscriber_identity_module#/media/File:GSM_SIM_card_evolution.svg

Standardisation (4/5)

- Command format
  - Protocol: APDU Application Protocol Data Unit
  - Communication between cardreader (CAD) and Smartcard
  - Command messages

APDU for Commands								APDU for Response				Compulsory
cla	ins	P1	P2	K	data	le		data	sw1	sw2		Optional

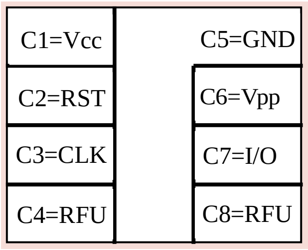
Smart Card Acceptance Devices (CAD)

- ISO 7816-4 standard
- Terminals
  - Have memory, logic, power
  - ATMS, gas pumps
- Readers
  - Connect to a computer
  - USB, serial, parallel port

Special security requirements for different use cases: access, payment, etc

I/O (Input/Output)

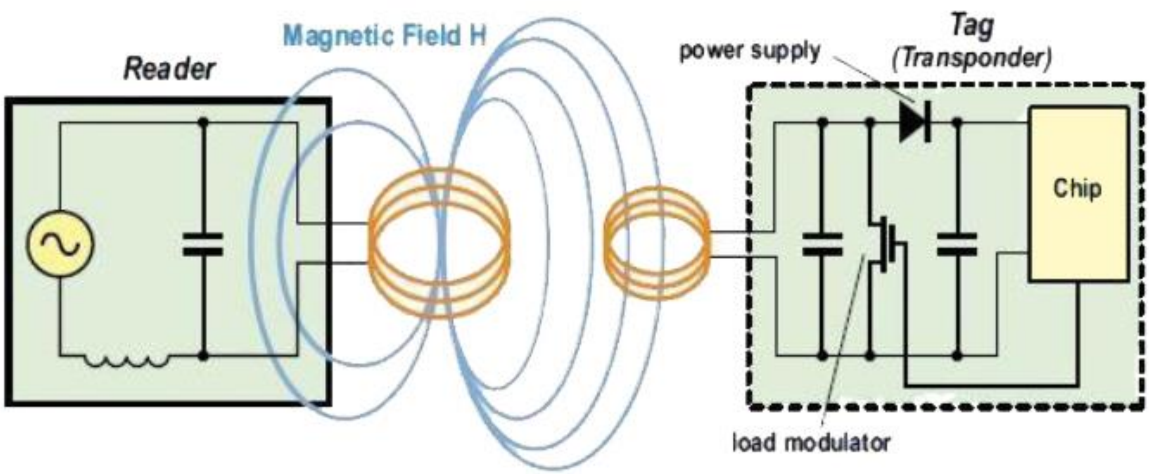
- **Contact Interface**
  - Vcc = 5 Volt (3 Volt)
  - Vpp not used anymore
  - CLK (3.5712, 4.9152, 10 MHz.)
  - UART for I/O



- **Contactless Interface** (125 kHz & 13.56 MHz)
  - *Close coupled*, a few millimeters
  - *Proximity*, less than 10 centimeter
  - *Vicinity*, more than 10 centimeter

Contactless Interface

- Power from CAD
- Modulation:
  - CAD →Card : AM, FM, PM
  - Card → CAD: AM
- Anti collision



Data Transmission T=0 protocol

- Byte oriented
- TPDU (Transmission Protocol Data Unit) ≈ APDU
  - CAD transmits CLA, INS, P1, P2, P3
  - Card transmits procedure byte ACK
  - Following communication depends on Command
  - Communications end with status bytes SW1, SW2
- Transmission errors detected via parity bit and corrected via second time transmission
- Poor separation of application and data link layer

IN → OUT

APDU for Commands								APDU for Response				Compulsory
cla	ins	P1	P2	K	data	le		data	sw1	sw2		Optional

Data Transmission T=1 protocol

- Block oriented

Prologue			Information	Epilogue
NAD	PCB	LEN	APDU	EDC
1 Byte	1 Byte	1Byte	0 - 254 Bytes	1-2 Bytes

- Block types:
  - I - application data
  - R - receive confirmation
  - S - protocol control data
- Good separation of application and data link layer which is good for multi application cards

  Yet T=0 is the one that is most often used.
- Transmission errors detected with EDC: LRC (XOR byte) or CRC ($x^{16}+x^{12}+x^5+1$), correction via S-block + PCB

Relationship between Client/Host

- Half duplex communication
- Master-Slave
- Who’s who?
    - Host – master
    - Applet – slave

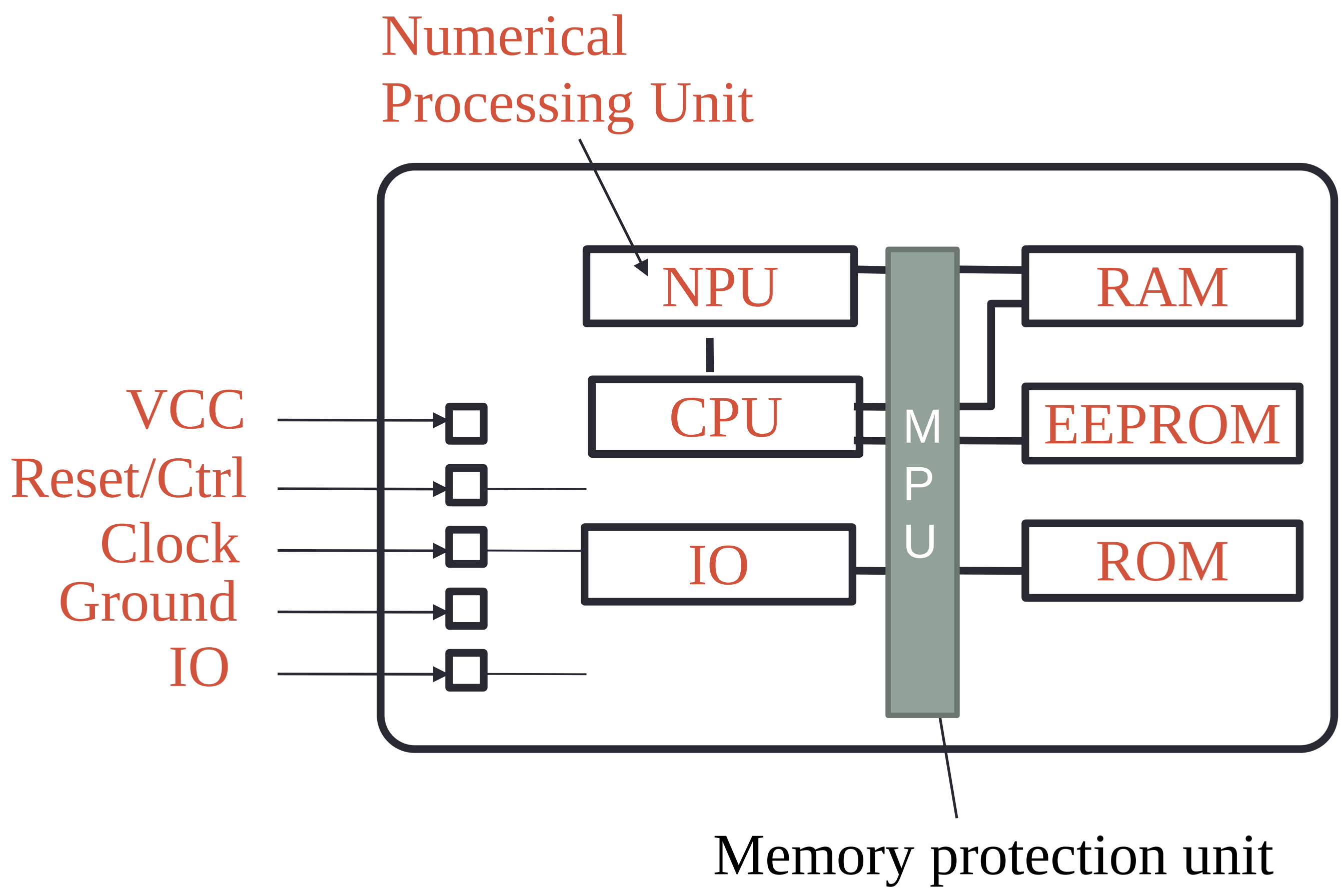
- Thus: Smartcard is passive and waits for a command. (Except at power up when it sends on its own the ATR=Answer To Reset response

Examples – PIN Verify *Command*

CLA	– 80
INS	– 20
P1	– 00
P2	– 00
Lc	– 03
L	– 010203
Le	– 00

So the command is: 80 20 00 00 03 010203 00

Architecture Smart Card

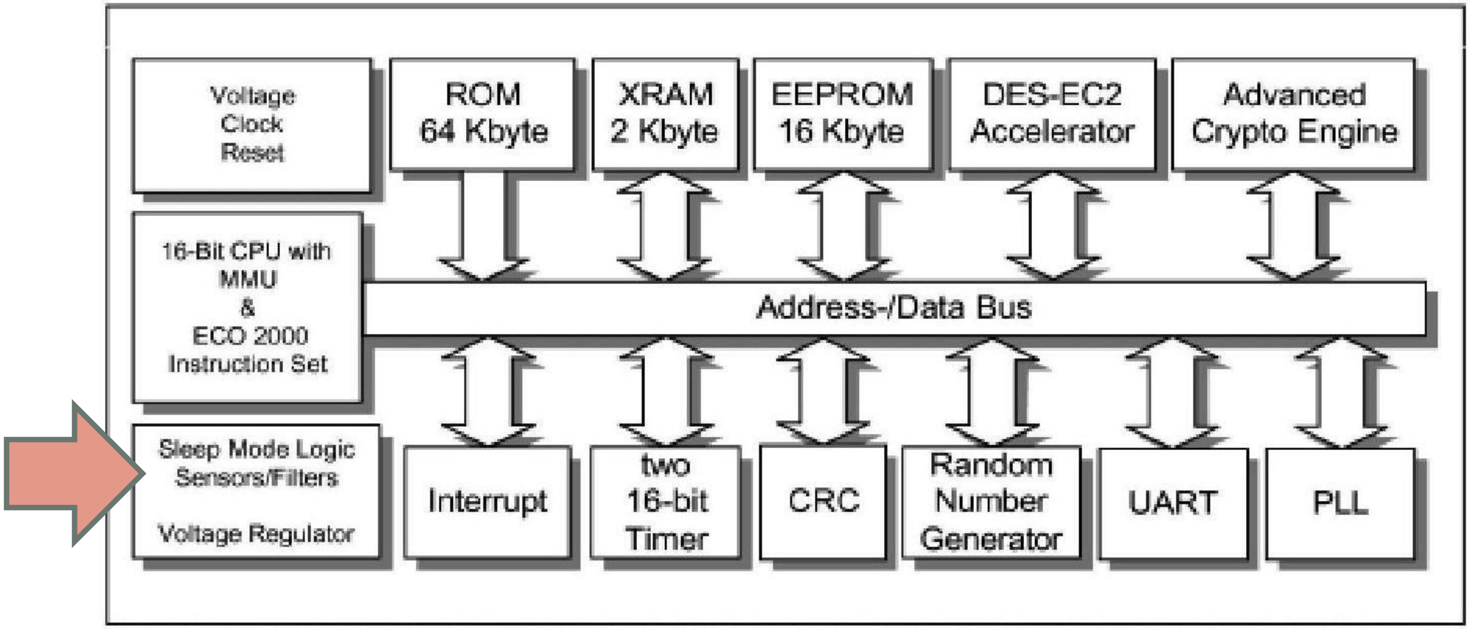


Memory

- ROM – Read only – from the birth of the card,
  - JCRE, applications, native code
- EEPROM – Persistent memory, 10 years
- RAM – Transient Memory, very expensive, “fast”

Example: Infineon SLE66

- **Smart card IC processor product with advanced security mechanisms (cryptographic engine, physical protection)**
- **Certifed according to EAL5**



Filters/Sensors

These are part of the mechanisms to protect the card against attacks

- Over/under voltage
- Too slow/too fast clock
- Penetration on ASIC

Smart card Life-cycle (1/2)

- **Production**
  - ROM : programing of code and constants

- **Initialisation**
  - EEPROM (Electrical Erasable Programmable ROM) : programming

- **Personalisation**
  - EEPROM : programming of user/application specific data

JavaCard

JavaCard

- Java Card is a stripped down version of Java for smart card
  - Familiar features including objects, inheritance, packages, dynamic object creation, virtual methods, interfaces, and exceptions.

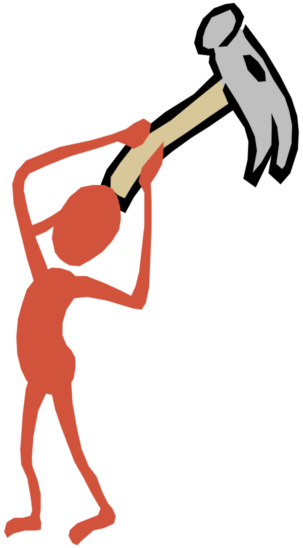
- Java Card makes multi-application cards based on a common platform possible
  - open up smart card development
  - use a real language and (re)use of standard SW development tool e.g. JBuilder

Bibliography

- JavaCard spec http://www.oracle.com/technetwork/java/embedded/javacard/downloads/releasenotes-jsp-1440109.html
- Java Card Techniques for Smart Cards, Chen, Zhiqun, Addison Wesley, 2000
- RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication, 3rd Edition
  Klaus Finkenzeller, Dorte Muller (Translated by) Wiley, June 2010
- Smart Card Handbook, 4th Edition, Wolfgang Rankl, Wolfgang Effing, Wiley, June 2010

Recommended: 1088 “European” pages

ATTACKS

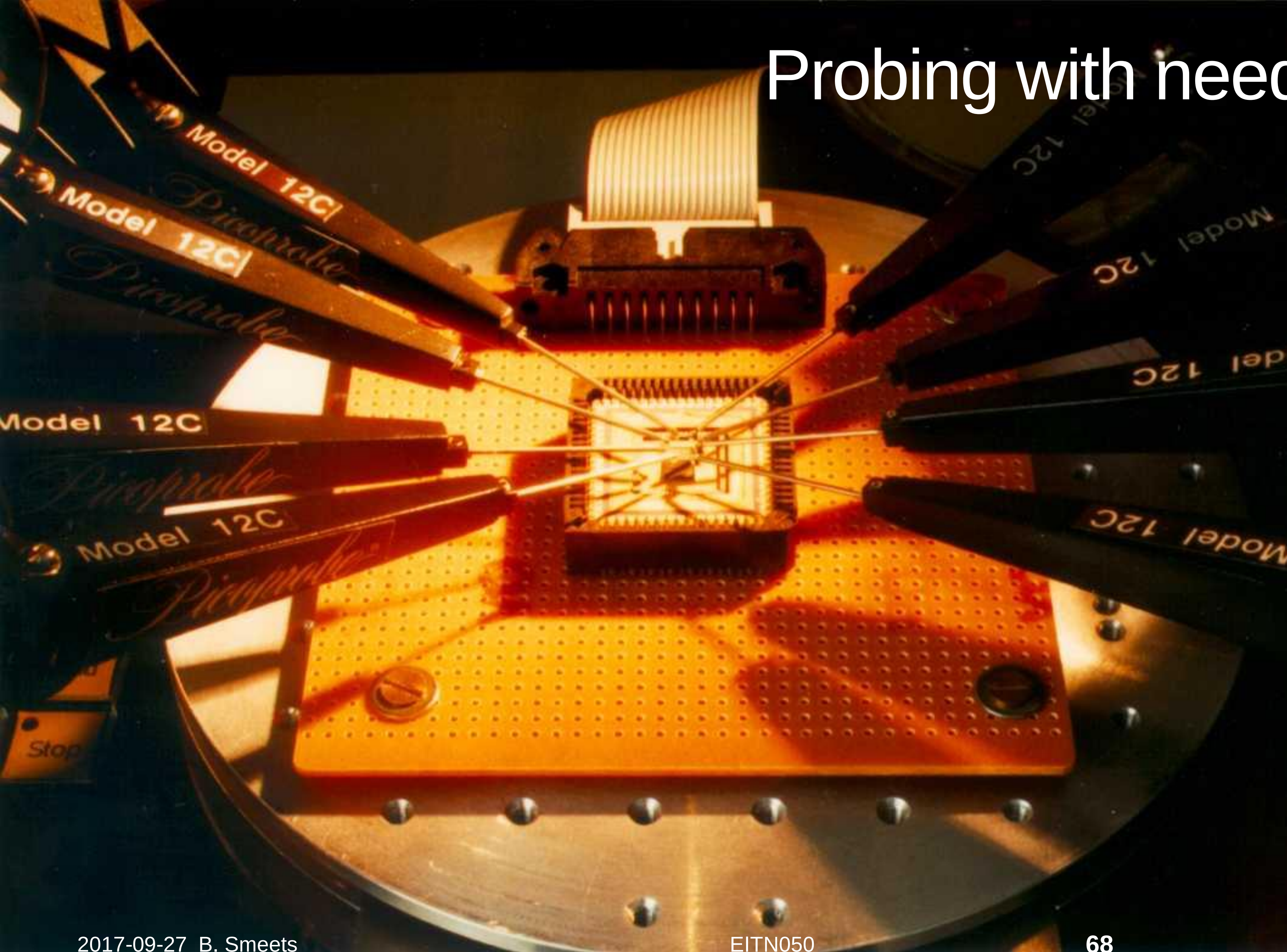


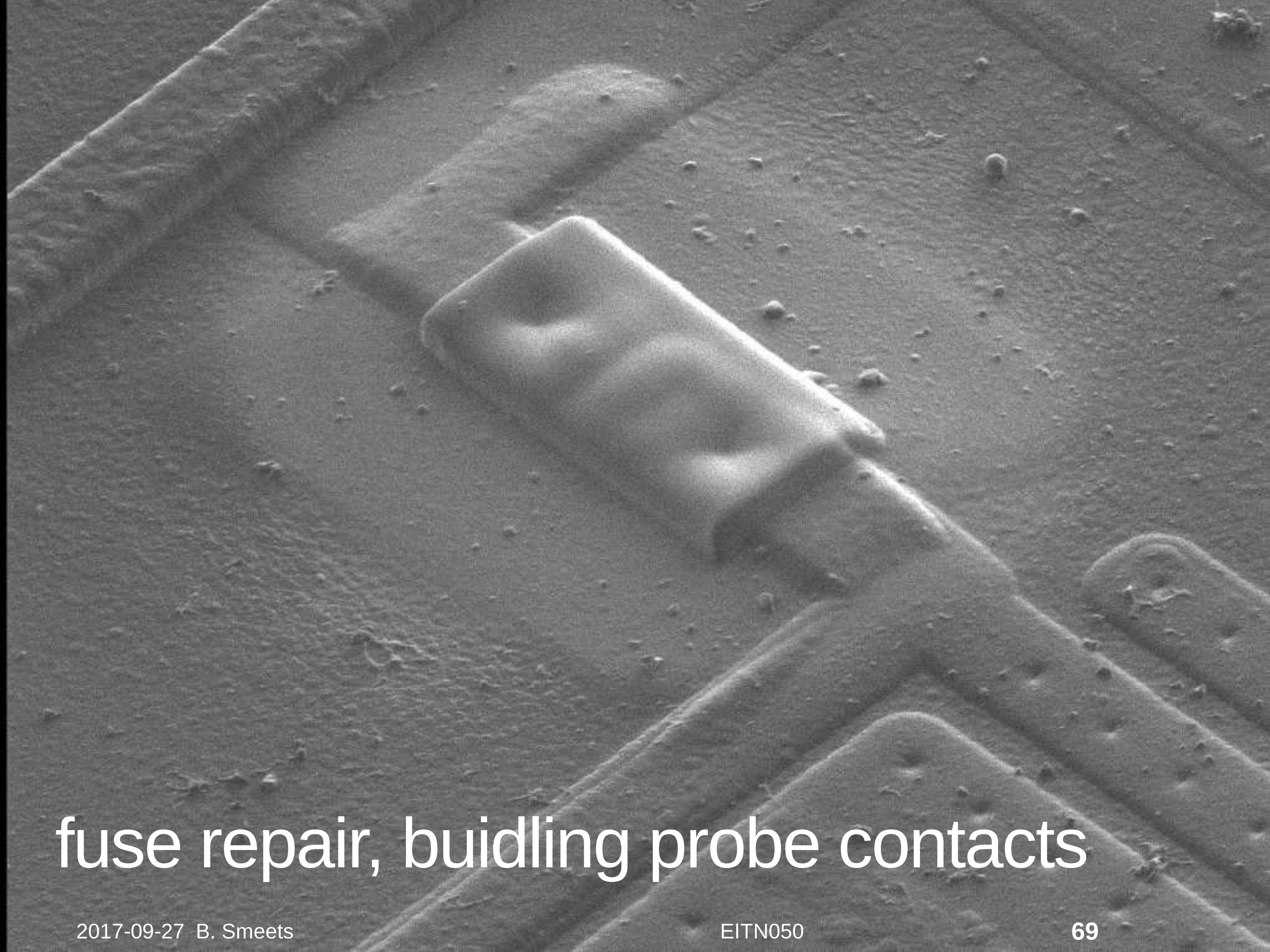
Two inroads for Attacks

- Traditional Mathematical Attacks
  - Algorithm modeled as ideal mathematical object
  - Attack would typically generalize
  - Attacks mostly theoretical rather than operational
- Implementation Attacks
  - Physical implementation is attacked
    - Reverse engineering
    - Probing
  - Vulnerabilities are difficult to control
  - Attacks are often operational—historically used to crack ciphers
  - Attack strategies are specific and do not generalize

Sub-micron probe stat

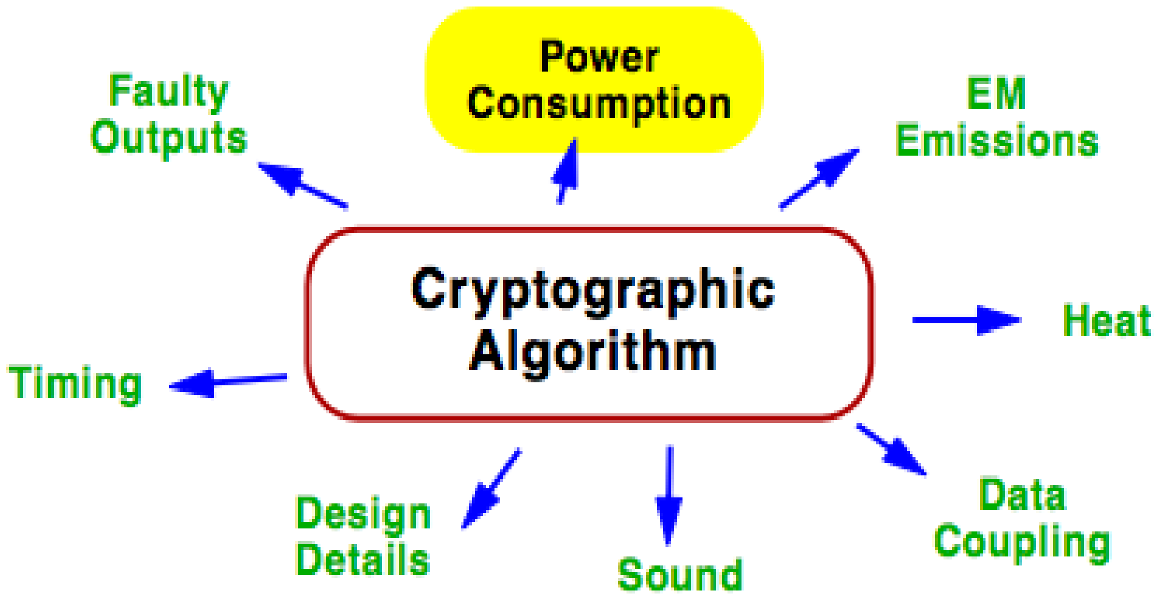
Probing with need



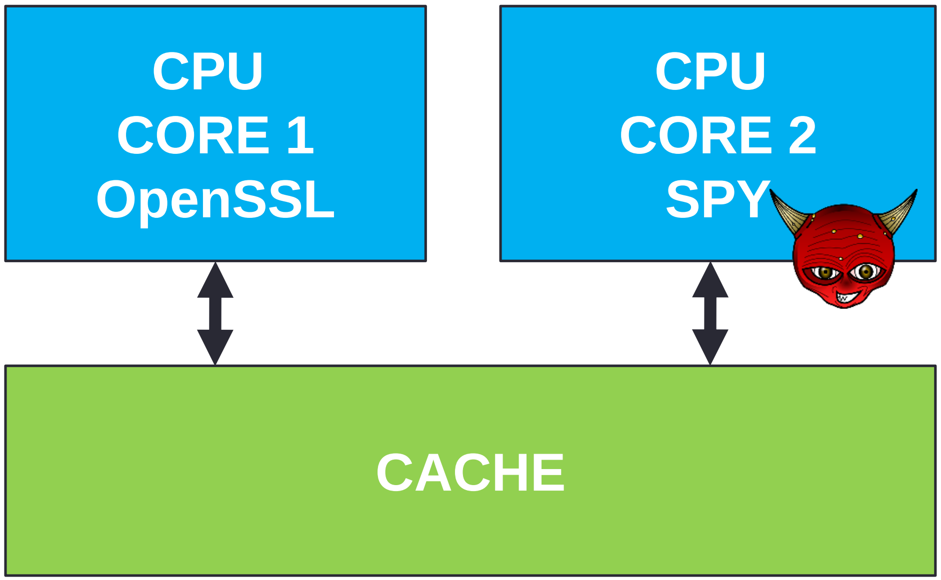


fuse repair, buidling probe contacts

Leakage Attacks



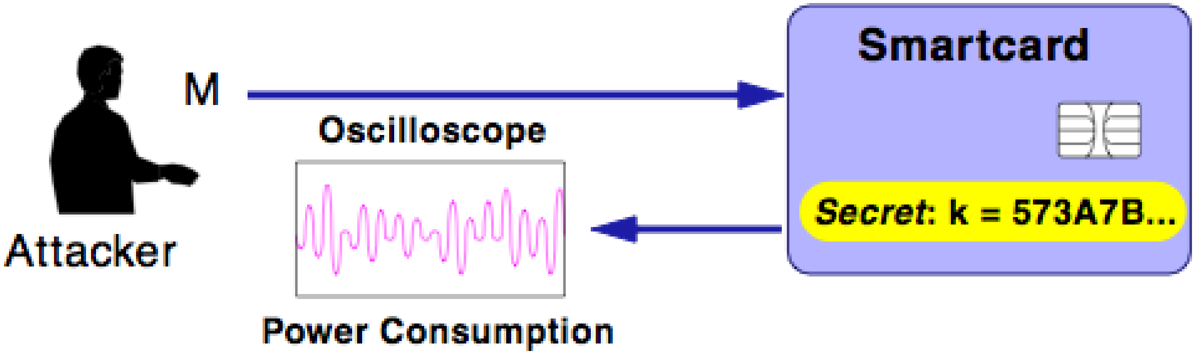
Sidechannel attacks – example in CPU



Use side information
- **Timing**
- **Cache misses**

Simple Power Analysis

- (E.g., Kocher 1998) Attacker directly uses power consumption to learn bits of secret key. Wave forms visually examined.
- Big features like rounds of DES, square vs. multiply in RSA exponentiation, and small features, like bit value.
- Relatively easy to defend against.



Attacking Modular Exponentiation

- Modular exponentiation is at heart of public-key cryptosystems
- Square-and-multiply in RSA; analogous double-and-add in Elliptic Curve
- Our Goal: Model, devise attacks, and implement attacks!

Review Square-and-Multiply Method

Compute: $M^e \bmod N$

Secret Key

```
exp1(M, e, N)
{
  R = M
  for ( i = n - 2 down to 0)
  {
    R = R^2 mod N
    if ( ith bit of e is a 1)
       R = R·M mod N
  }
  return R
}
```

Example: e = 83 → 1010011

i	e	R
-	1	M
5	0	M^2
4	1	M^5
3	0	M^{10}
2	0	M^{20}
1	1	M^{41}
0	1	M^{83}

Countermeasures for Power Analysis Attacks

- Software Countermeasures
  - Time randomization: add random delays
  - Permuted execution
  - Data Masking Techniques
- Hardware Countermeasures
  - Noise generation, power signal filtering, novel circuit designs
  - But must consume some energy to process data

Summary

- There is a potential risk of DPA to recover key from a smartcard
- Today the problem is rather well understand and countermeasures against DPA are included in the crypto and card design.
- There always lures the danger of fault insertion

But still: August 2015

Attack presented at BlackHat conference 2015

The Register®
Biting the hand that feeds IT

DATA CENTRE SOFTWARE NETWORKS SECURITY INFRASTRUCTURE BUSINESS HARDWARE SCIE

Security

Researchers look sideways to crack SIM card AES-128 encryption

Gone in ten minutes, with a little help from some exotic hardware

6 Aug 2015 at 02:32, Iain Thomson

409 76 256

RADIO FREQUENCY IDENTIFICATION (RFID) AND NEAR FIELD COMMUNICATION (NFC)

In this lecture

- Automatic Identification Techniques
- What is RFID and NFC
- Brief history of RFID
- Standards
- Applications
- Security issues

Automatic Identification Technologies

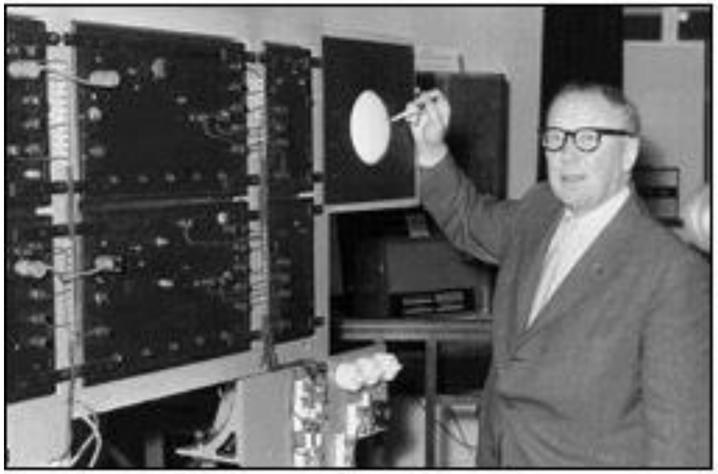
- OCR (Optical Character Recognition)
- Magnetic Stripe
- Barcode
- 2D Code
- RFID (Radio Frequency Identification)
- Biometrics
- New chipless methods (SAW, Radar ...)
- NFC (Near Field Communication)
  - Radio, capacitive, inductive, …

What is RFID ?

- Radio Frequency Identification
- It is a type of automatic identification system.
- Purpose: to enable data to be transmitted by a portable device, called a tag.

History of RFID

- January 23, 1973: the first U.S. patent for an active RFID tag with rewritable memory
- Mid-1980s: it became commercialized
  - Developed a passive RFID tag to track cows



RFID - short facts …

Advantages

- data can be modified and completed
- no ‚line of sight‘ contact required
- several transponders can be processed at the same time
- speed and high processing rate

Disadvantages, Limitations …

- Interference at metal surfaces, water, humidity
- reading distance, environment …

Chip producers

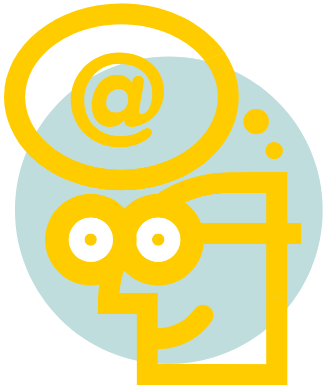
- EM, Infineon, Philips, STM TI, …

NFC - Technical Basics

- Wireless Short Range Communication Technology
  - Based on RFID technology at 13,56 MHz
  - Operating distance typical up to 10 cm
  - Compatible with today’s field proven contactless RFID technology
  - Data exchange rate today up to 424 kilobits/s

Security of NFC

- RFID and NFC industry is working hard to build *reliability* into the infrastructure
- An important next step is to build *trust*
- Much discussion so far has focused on *privacy*
- Next to come is demand for : *authentication*
- Analogy: Internet from 30 years ago to present



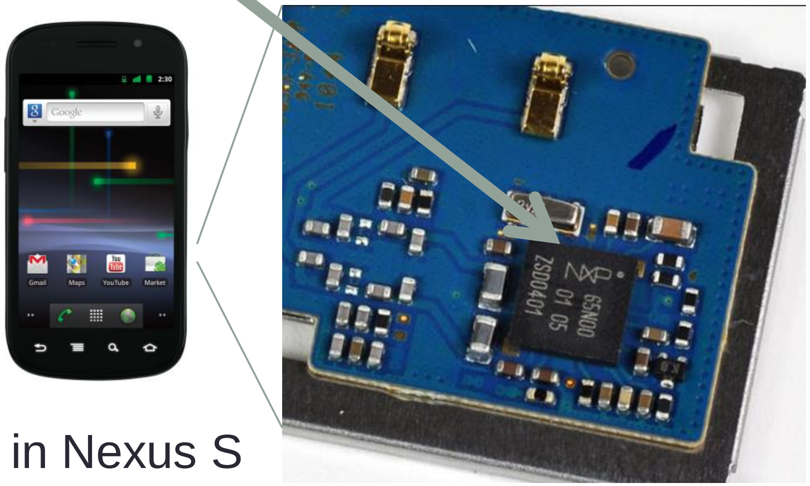
RFID vs NFC

- Simply put
- RFID is a technology that interacts like ‘Hi, here I am -- look at me!' .
- NFC and contactless is technology where the interaction is like, ‘I'm not talking to you until I know you're someone I should talk to”, which assumes some authentication.
- But this distinction can be blurred in a particular system

Mifare and FeliCa

- FeliCa is the name of Sony contact less smart card mainly used in Japan, Singapore, (US?)

- MIFARE is the name of NXP proprietary technologies based upon various levels of the ISO/IEC 14443 Type A 13.56 MHz contactless smart card standard.



NFC in Nexus S

But: What about Attacks

- **Skimming**: Reading legitimate tag data to produce fraudulent clones.
- **Swapping**: Steal RFID/NFC-tagged products then replace with counterfeit-tagged decoys.
- **Denial of Service**: Seeding a system with fake, but authentic acting tags.

Breaking NFC

- Search for open tools

- Hacking of MIFARE – film
  - Mifare (Little Security, Despite Obscurity), Karsten Nohl, Henryk Plötz
  - https://www.youtube.com/watch?v=QJyxUvMGLr0

Mifare - variants

Broken or security weaknesses reported

- ~~MIFARE Classic~~ *)
- ~~MIFARE Ultralight~~ and MIFARE Ultralight EV1
- MIFARE Ultralight C
- ~~MIFARE DESFire~~
- MIFARE DESFire EV1
- MIFARE Plus
- MIFARE SAM av2

*) Not supported in NFC standard but NXP chipsets often supports this.

http://www.ru.nl/ds/research/rfid/

Can we avoid expensive crypto?

- RSA and ECC are fine by have problem
  - Complex
  - Relatively slow on low-power hardware:
    - Not fast enough to be used in speed gates
- Alternatives exist
  - But beware when implementing.

Hopper-Blum Authentication

Shared secret

Computer($\mathbf{x}$)

Bob($\mathbf{x}$,Prob= η)

$\mathbf{a} \in \{0,1\}^k$

Challenge

$z=(\mathbf{a}\cdot\mathbf{x})\oplus v$

$z=(\mathbf{a} \cdot \mathbf{x})$?

Response

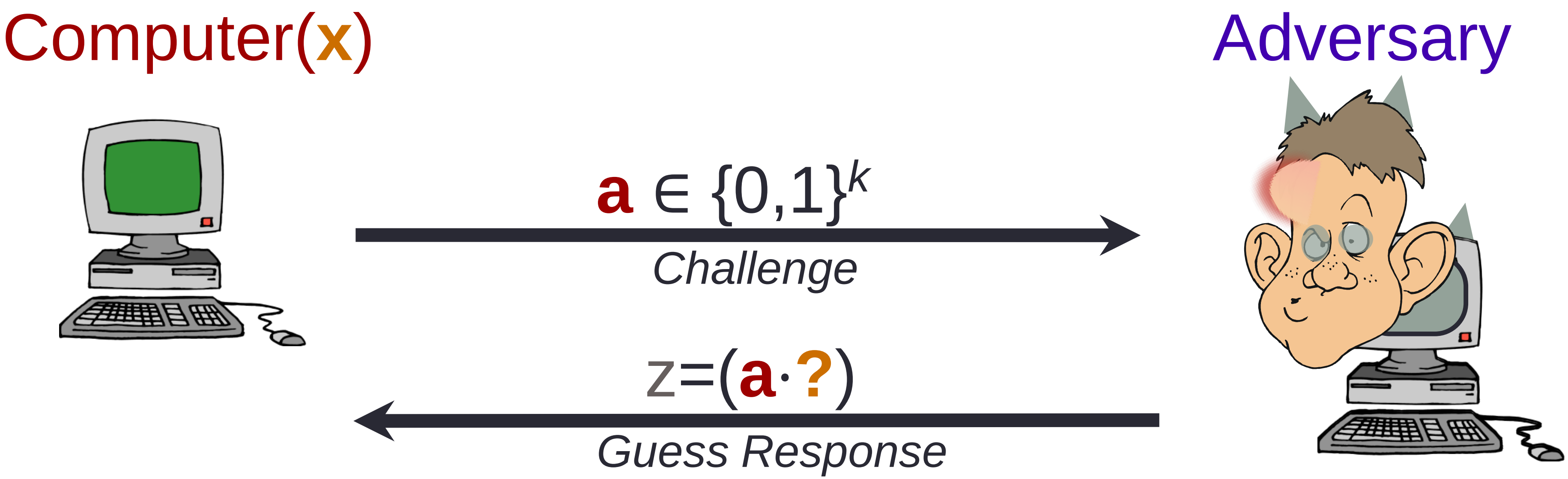
$v \in_R \{0,1\}$

Repeat for *q* rounds.

Authenticate Bob if he passes ≈ (1- η)q rounds.

Inner product ($(\mathbf{a_1}, \ldots, \mathbf{a_k}) \cdot (\mathbf{x_1},\ldots,\mathbf{x_k})$) $\in_R \{0,1\}$

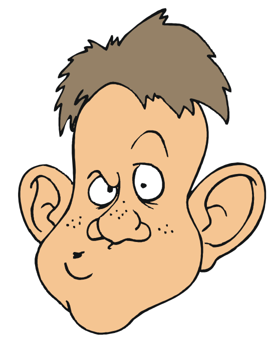
Security Against Bad Bob



Security Against Passive Eavesdroppers

Computer(**x**)

Bob(**x**, η)



$v \in_R \{0,1\}$

Eavesdropper

$(\mathbf{a_0}, z_0), (\mathbf{a_1}, z_1), \ldots, (\mathbf{a_q}, z_q)$

Find an **x’** that allows you to answer a (1- η) fraction of **a** challenges

Learning Parity with Noise (LPN)

This problem is well studied:

- Crypto and learning problems
- LPN algorithm has $O(2^{\frac{k}{\log k}})$ complexity
- Shortest Vector Problem reduction

Concrete Security

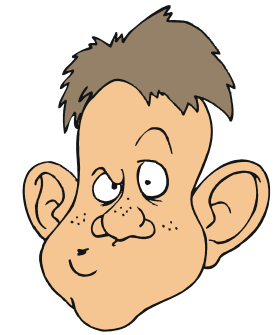
Key Size (k)	Best Attack
64	2^{35}
128	2^{56}
192	2^{72}
224	2^{80}
256	2^{88}
288	2^{96}

Estimates !

Active Attack against HB

Adversary

Bob($\mathbf{x}$, η)



$\mathbf{a'} = 000...001$

$z_0 = (\mathbf{a'} \cdot \mathbf{x}) \oplus v_0$

...

$\mathbf{a'}$

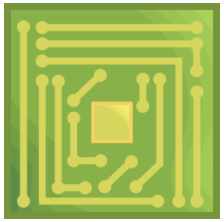
$z_n = (\mathbf{a'} \cdot \mathbf{x}) \oplus v_n$

Adversary takes majority of z_i values to get noise-free parity bit and recovers matching coordinate in $\mathbf{x}$

Extended Protocol: HB+

Reader($\mathbf{x}$, $\mathbf{y}$)

Tag($\mathbf{x}$, $\mathbf{y}$, η)



$\mathbf{b} \in \{0,1\}^k$

Blinding Factor

$\mathbf{a} \in \{0,1\}^k$

Challenge

$v \in_R \{0,1\}$

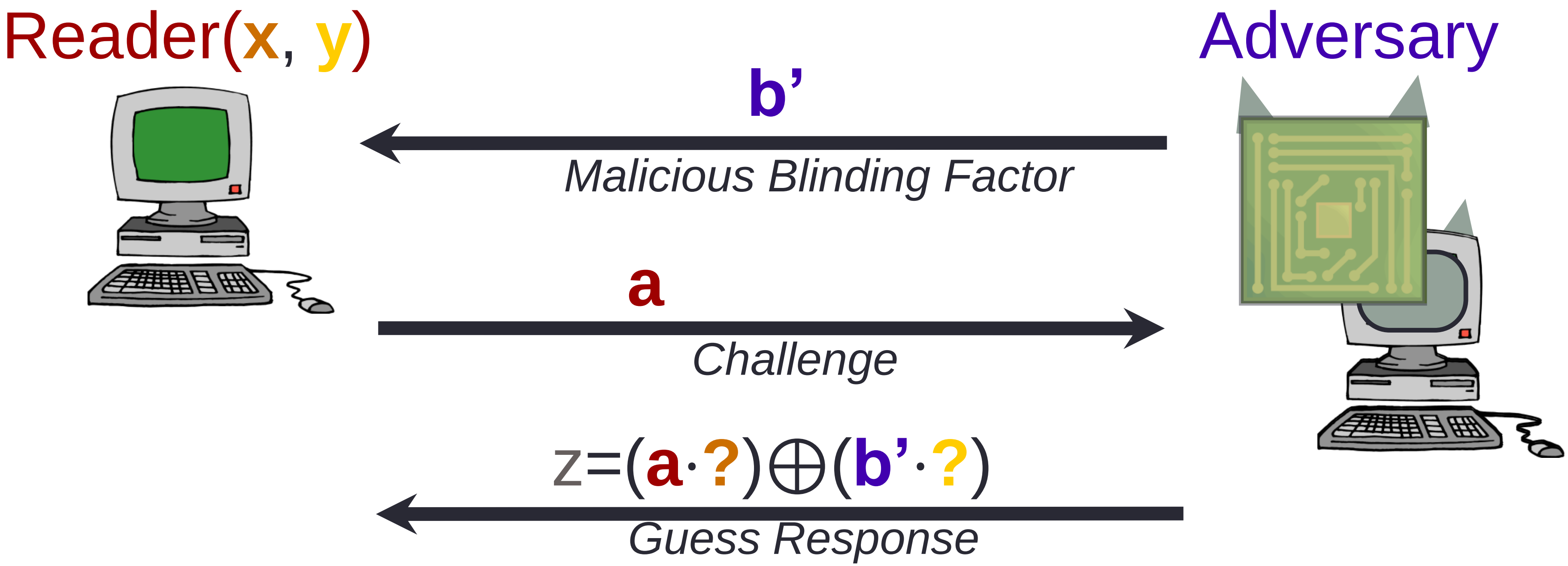
$z=(\mathbf{a}\cdot\mathbf{x})\oplus(\mathbf{b}\cdot\mathbf{y})\oplus v$

Response

check

$z=(\mathbf{a}\cdot\mathbf{x})\oplus(\mathbf{b}\cdot\mathbf{y})$?

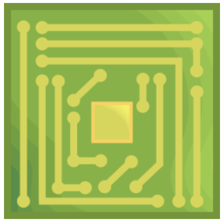
Security Against Bad Bob



Security against Active Attacks

Adversary

Tag($\mathbf{x}$, $\mathbf{y}$, η)



$\mathbf{b}$

Blinding Factor

$\mathbf{a'}$

Malicious Challenge

$v \in \{0,1\}$

$z=(\mathbf{a'}\cdot\mathbf{x})\oplus(\mathbf{b}\cdot\mathbf{y})\oplus v$

Response

Blinding sees that we get randomization in the responses and we are back to LPN