Attacking smartcards

Erik Poll

Digital Security

Radboud University Nijmegen

Smartcard security

- Smartcards are not 100% secure
  - ongoing arms race of attacks & countermeasures
  - Ten year old cards may be easily broken today
- Crucial question: is the risk acceptable?
  - Is the effort for the attacker greater than what he can gain?
- Threats depend on application

Classification of attacks

- cost
  - time
  - equipment
  - know-how
- tamper-evidence
  - ie can the card, card holder, or card issuer see a card is being or has been messed with?
- impact for the organisation
  - and business case for the attacker

The attacker's business case

ie. the motivation for professional attacker!

The hobbyist is after fame or publicity,

the professional is after money!

Which smartcard most interesting to "hack"?

SIM, bank- or creditcard, pay TV

Here by "hack" we mean access private keys on the chip to clone cards

Most interesting: PayTV?

Least interesting: SIM card?

Cloning bank card only interesting if you can do it quickly without owner noticing, not if you have to do it in the laboratory.

Classification of attacks

An attacker can target

1. organisation: eg. issuance & usage process

2. cryptographic *algorithms*
3. *protocols, or software implementing it* on smartcard or terminal-side

} logical attacks

4. the physical smartcard itself

   side-channel attacks or invasive attacks

Smartcard attacks: cost

- Logical attacks

  Only 50$ of equipment, but possibly lots of brain power!

  Analysis may take weeks, but final attack can be in real time
- Side channel attacks (DPA)

  5K$ of equipment

  Again, lots of time to prepare, but final attack can be quick
- Physical attacks

  100K$

  Several weeks to attack a single card

(1) Attacking the crypto

Attacking the crypto

Difficult for standard algorithms, eg

DES, 3DES, AES, RSA, ECC, …

Homemade, *proprietary* cryptographic algorithms are routinely broken, eg

- Crypto-1 used in MIFARE Classic
- COMP128 and A5/1 used in GSM
- Keeloq used for car keys
- SecureMemory, CryptoMemory, CryptoRF
- iClass, iClass Elite
- HiTag2

movies: MIFARE and Hitag2

https://www.youtube.com/watch?v=NW3RGbQTLhE

https://www.youtube.com/watch?v=S8z9mgIkqBA

https://www.youtube.com/watch?v=dZfxdctzX6Q

Attacking the key management

You can easily check that people use proper cryptographic `algorithms, but not that people use it properly…

Common problems: using same key in all cards, or default keys

For example

- the iClass tags use one master key that is in all readers!
- 75% of MIFARE applications was found to use default keys or keys used in examples in documentation
  - A0A1A2A3A4A5 is an initial transport key of MIFARE tags. Googling for A0A1A2A3A4A5 produces links to documentation with other example keys to try!

(2) Attacking the protocols
(or the software implementing them)

Attacking the protocols

- Replay attack

  record communication between card & terminal, and replay it

  Shouldn't work for well-designed protocol!

- Man-in-the-Middle attack

  intercept *and modify* the communication

  Shouldn't work for well-designed protocol!

- Relay attack

  intercept communication and relay it to a different terminal

  Eg from hacked PIN terminal in mafia-operated shop to an ATM

  Very hard to prevent, if relay is done fast enough!

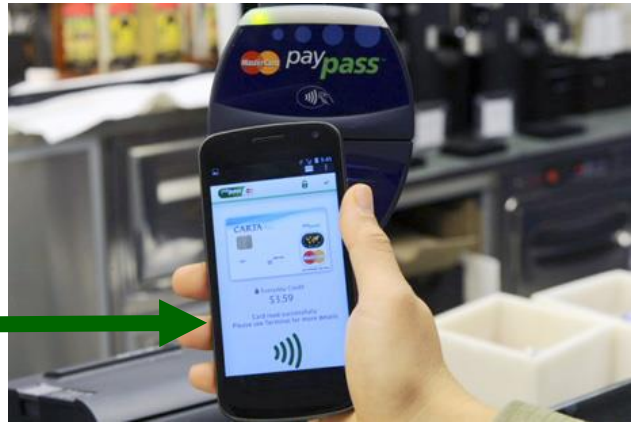
Relay attack

Using two NFC mobile phones, you can carry out a relay attack

- one phone in active mode, to read tag of the vicim
- another phone in card emulation mode, to forward the tag's communication to a real terminal

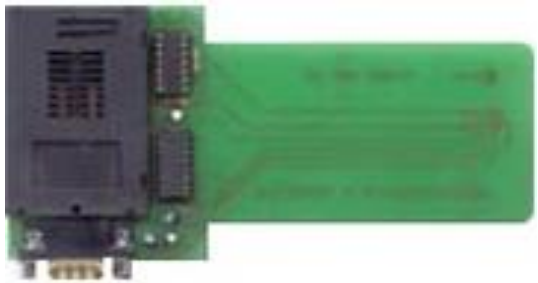
NFC phone
in active mode

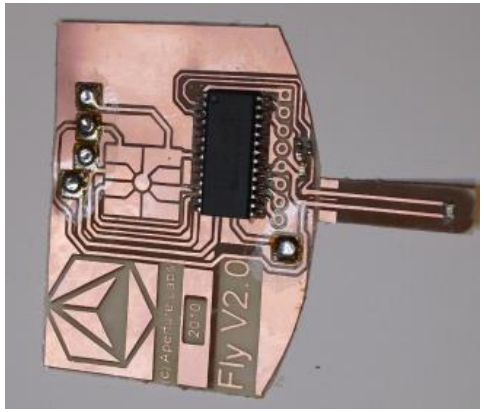
data
connection



NFC phone
in card emulation mode

Tools for protocol analysis





Example protocol attack on Dutch ov0 chipcard

- one-time programmable memory for invalidating disposable public transport card.

  initially 0x00F0, set to 0xF8FF to invalidate card
- Flaw: *we can change an invalid tag so that some terminals fail to recognize it as invalid; can you guess the flaw?*
  - remaining 3 lock bits can still be set to one, so that lock bytes become 0xFFFF
  - flaw in terminals: tags with lock bytes 0xF8FF are recognized as invalid, but tags with 0xFFFF are not

  Can you guess the terminal code that causes this?

[Pieter Siekerman and Maurits van der Schee, Security Evaluation of the disposable OV-chipkaart, MSc thesis, UVA, 2007]

Example protocol attack: ABN AMRO e.dentifier2

Malicious software on the PC can by-pass user approval (pressing OK on the e.dentifier2) if USB connection is used

(3) side-channel attacks

Smartcard attacks

So far we discussed logical attacks (50$) to exploit flaws in

- crypto, security protocol (or the software implementing it)

Other possibilities

- Side channel attacks (5K$)
  - passive: power or timing analysis
  - active: fault injection (glitching or laser attacks)

- Physical attacks (100K$)
  - reverse engineering
  - probing, focussed ion beam, ...

These attacks may also be combined

Invasive vs non-invasive

- Logical & side-channel attacks are non-invasive
  - violate tamper-resistance *and* tamper-evidence
  - *can happen in a few minutes in mafia-operated shop or a tampered terminal*

- Physical attacks are always invasive
  - tamper-evident, so only violate tamper-resistance
    - ie you destroy a few chips in the process
  - *requires hours to weeks in laboratory*

Side-channel analysis

example side channel:

pizza deliveries to the Pentagon

Side-channel analysis

monday evening

tuesday evening

What evening is the invasion taking place?

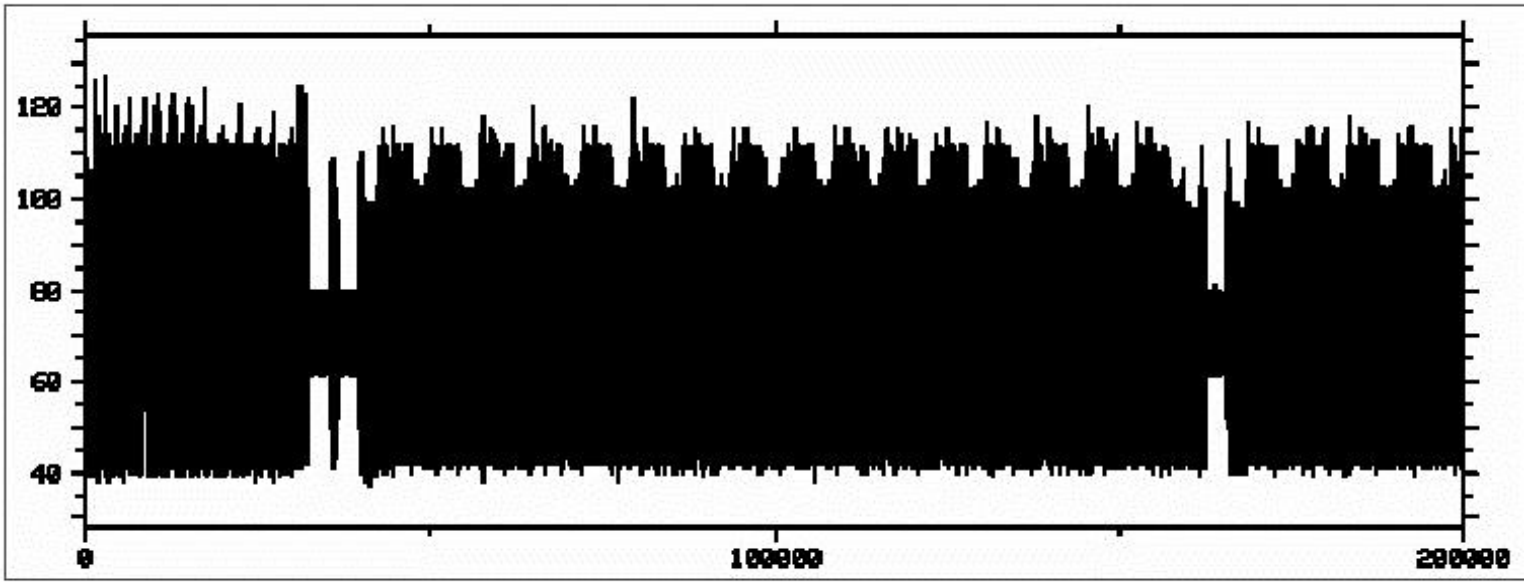
Side-channel analysis

- Side-channel = any other channel than the normal I/O channel that may be observed

- Possible side-channels:
  - power consumption
  - timing
  - electro-magnetic radiation
  - ....

Very powerful !

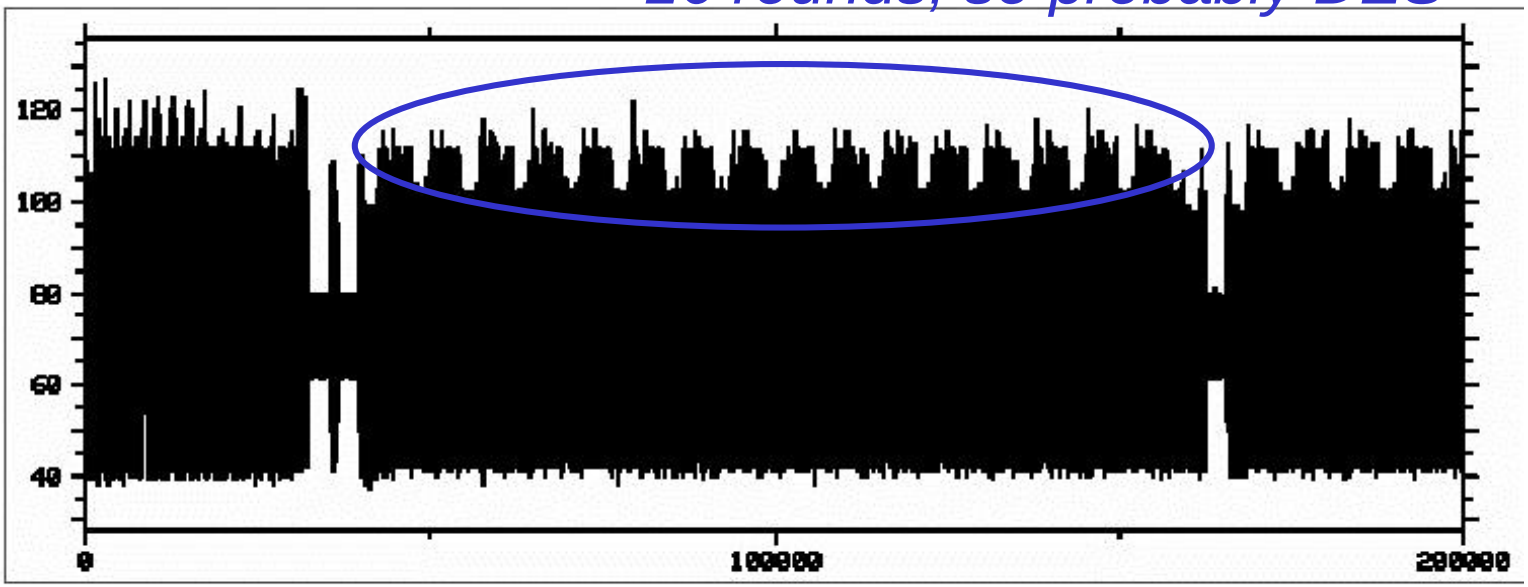
Power consumption of a smartcard



What is this card doing?

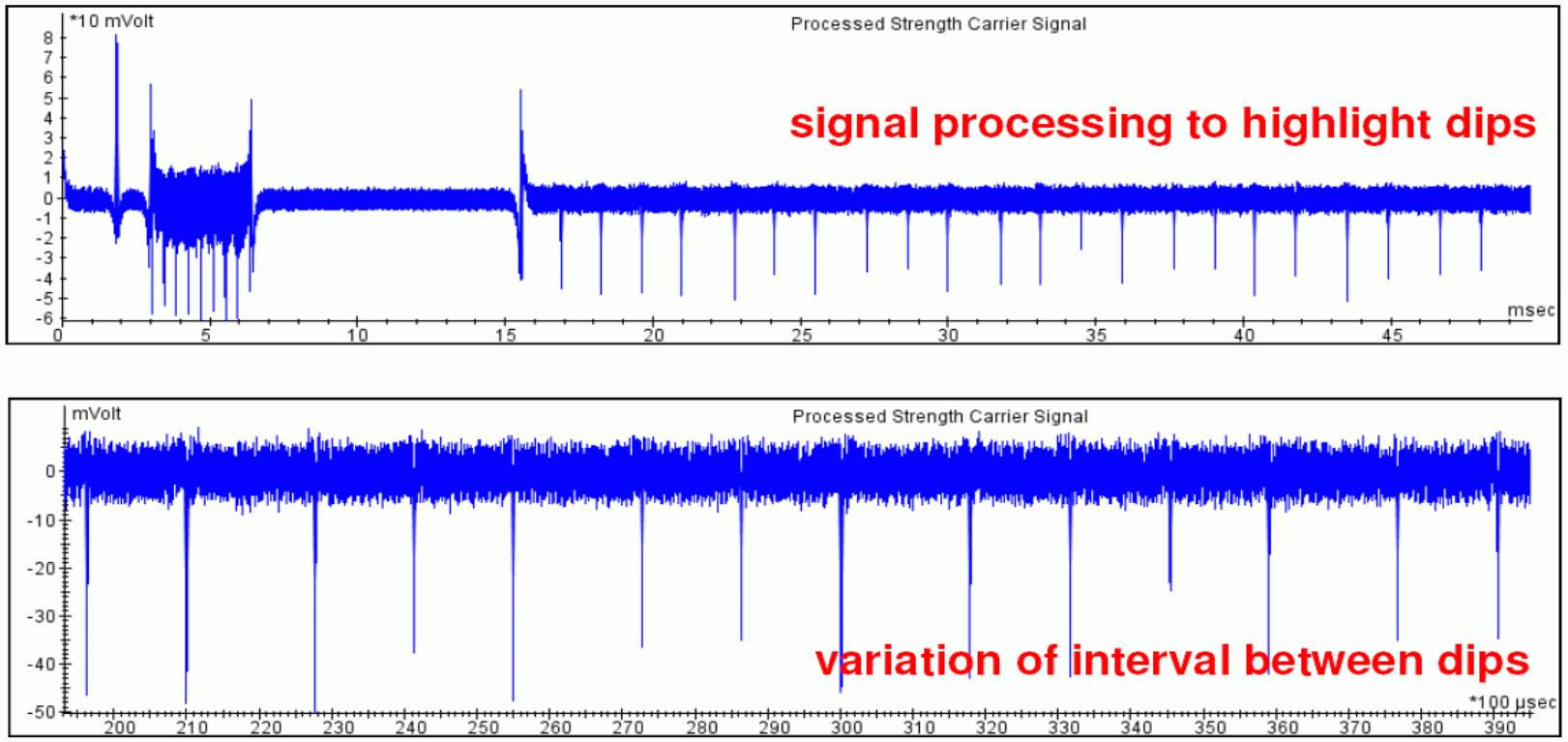
This is a DES encryption!

16 rounds, so probably DES



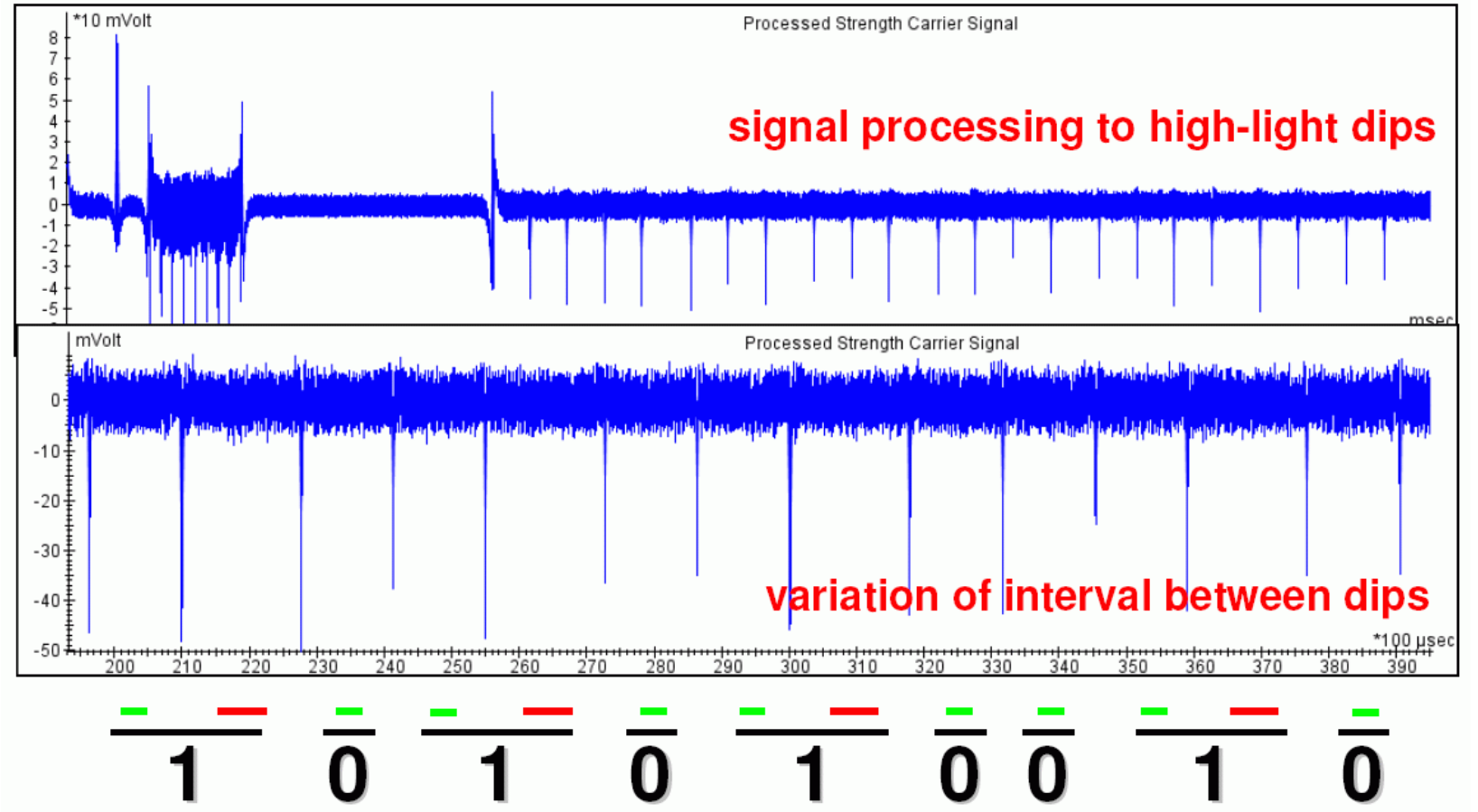
What is the key?

Power trace detail of RSA encryption



Source: presentation by Fred de Beer of Riscure at Safe-NL, June 2006

SPA: reading the key from this trace!



Source: presentation by Fred de Beer of Riscure at Safe-NL, June 2006

Power Analysis

- Simple Power Analysis - SPA
  - analyse an individual power trace
    - to find out the algorithm used
    - to find the key length
    - worst case: to find the key
- Differential Power Analysis - DPA
  - statistically analyse many power traces to find out the key

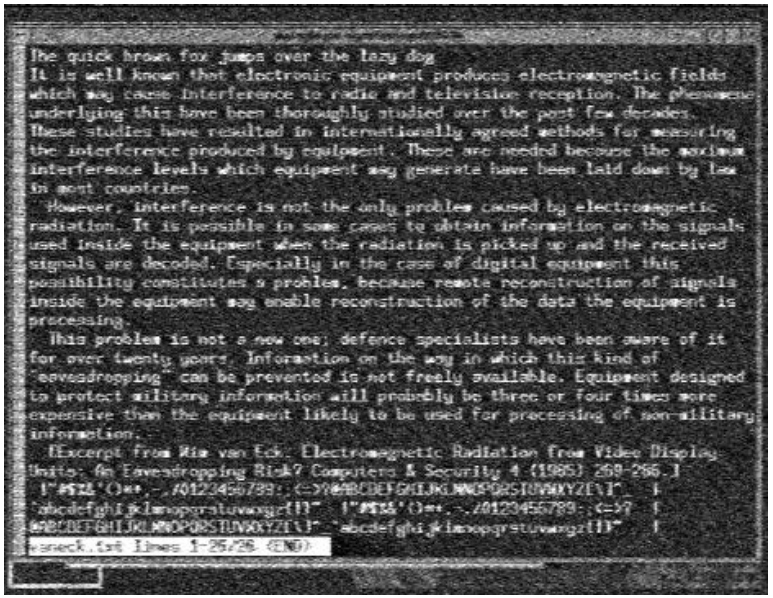
DPA has been the most serious threat to smartcards in the past 10 years!

Equipment for side-channel analysis in our lab

Other side-channel attacks: EMSEC/TEMPEST

Side-channel analysis is not just a problem for smartcards:

Electro-magnetic emission from all electronic equipment - incl. displays and keyboards - can leak information



laptop screen
at 10 meter
through
3 (thin) walls

Dutch voting machines were banned (officially) because of EM emissions violated secrecy of the votes.

Other side channel attacks: timing

Timing attack on the password check in the TENEX operating system allowed passwords to be guessed:

Response time for rejecting an incorrect password depended on which character was the first wrong character in the password guess

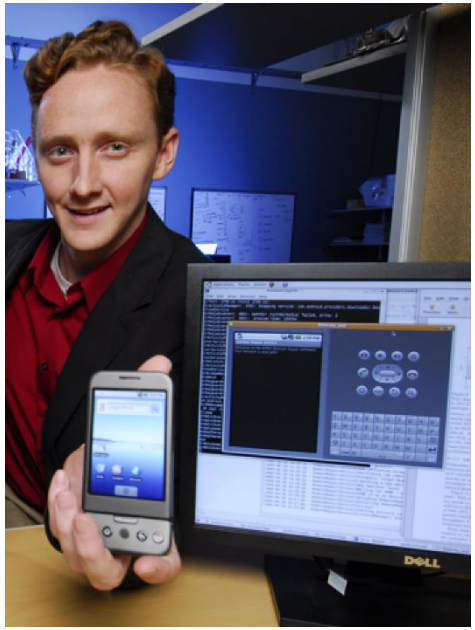
- 1st character wrong => quick reponse,
- 2nd character wrong => slower response, ...

The attack cleverly used page faults to make the delay observable

Other side channel attacks: keyboard vibrations

The accelerometer in a smartphone
can pick up vibrations of
a key board on the same table

... and reconstruct the input



(4) active side-channel attacks

Attacks with fault injections

Faults may be introduced as part of attacks

- card tears removing the card from the reader halfway during a transaction
  - *homework exercise: try this when charging or paying with your chipknip!*
- glitching temporarily dipping the power supply
  - eg to prevent EEPROM write after trying a PIN code
- light attacks shoot at the chip with a laser
  - to flip some bits...

Spot the weakness! Hint: card tear

```
class PIN{
      int tryCounter = 3; // no of tries left
  byte[] pin;
 …
boolean check (byte[] guess) {
      if (tryCounter != 0) {
         if arrayCompare(pin, 0, guess, 0, 4)
               { tryCounter = 3;
                 return true;}
         else {tryCounter--;
                return false; }
      else return false}
```

Spot the weakness! Hint: c

```
class PIN{
      int tryCounter = 3; // no of t
  byte[] pin;
 …
boolean check (byte[] guess) {
      if (tryCounter != 0) {
         if arrayCompare(pin, 0, guess, 0, 4)
               { tryCounter = 3;
                 return true;}
         else {tryCounter--;
                return false; }
      else return false}
```

cutting power at this point will leave tryCounter unchanged

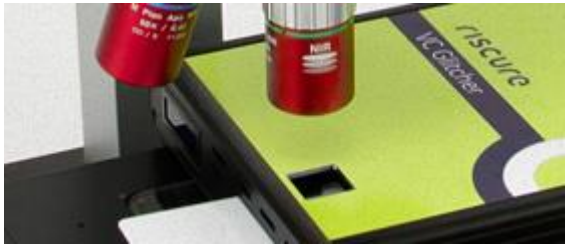
More secure code

```
class PIN{
      int tryCounter = 3; // no of tries left
  byte[] pin;
 …
boolean check (byte[] guess) {
      if (tryCounter != 0) {
         tryCounter--;
         if arrayCompare(pin, 0, guess, 0, 4)
               { tryCounter = 3;
                 return true;}
         else { // tryCounter--;
                return false; }
      else return false}
```

Remaining worries: *can timing of* ***`arrayCompare`*** *leak info ?*
Can ArrayIndexException for **`guess`** *of length < 4 leak info?*

laser attacks

laser mounted on microscope
with x-y table to move the card
and equipment to trigger timing



(5) physical/invasive attacks

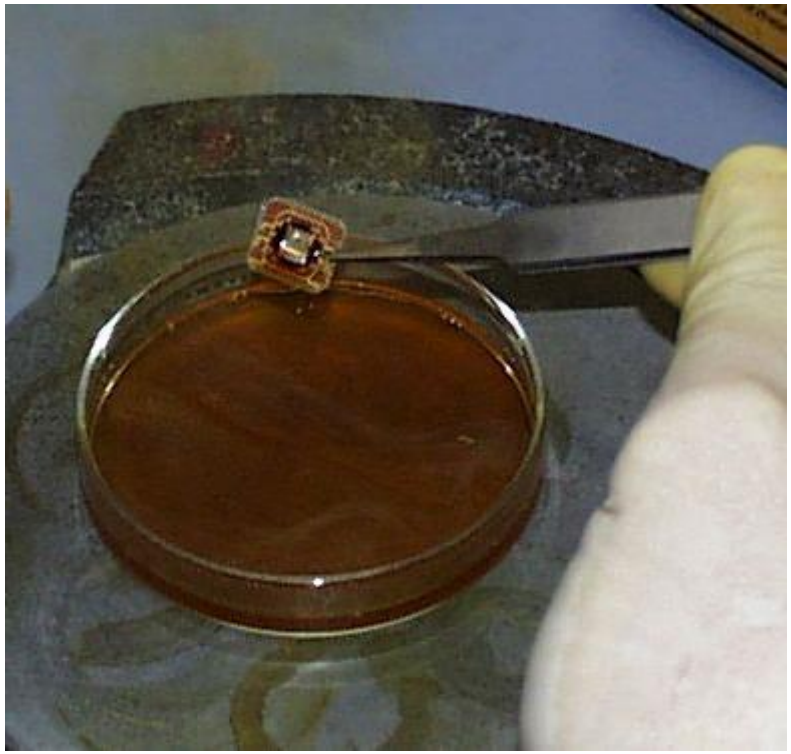
Physical, invasive attacks

- Much more costly than logical or side channel attacks.
  expensive equipment + lots of time & expertise
- Also, you destroy a few chips in the process...

Examples

- probing
- fibbing
- reading memory contents
- …

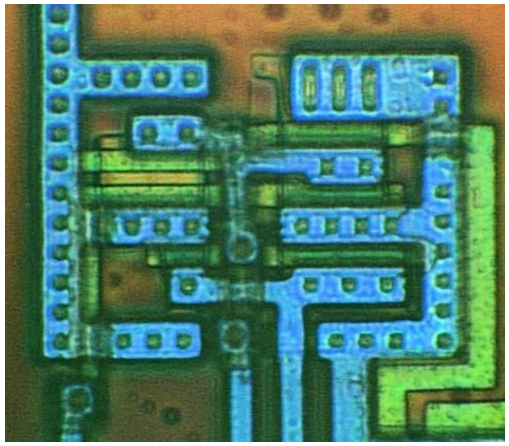
First step: removing chip from smartcard

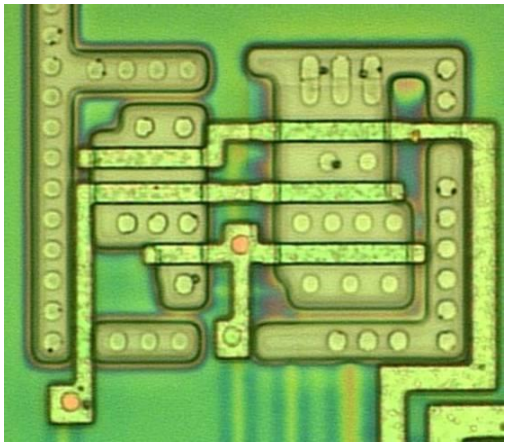


using heat & nitric acid

[Source: Oliver Kömmerling, Marcus Kuhn]

Optical reverse engineering

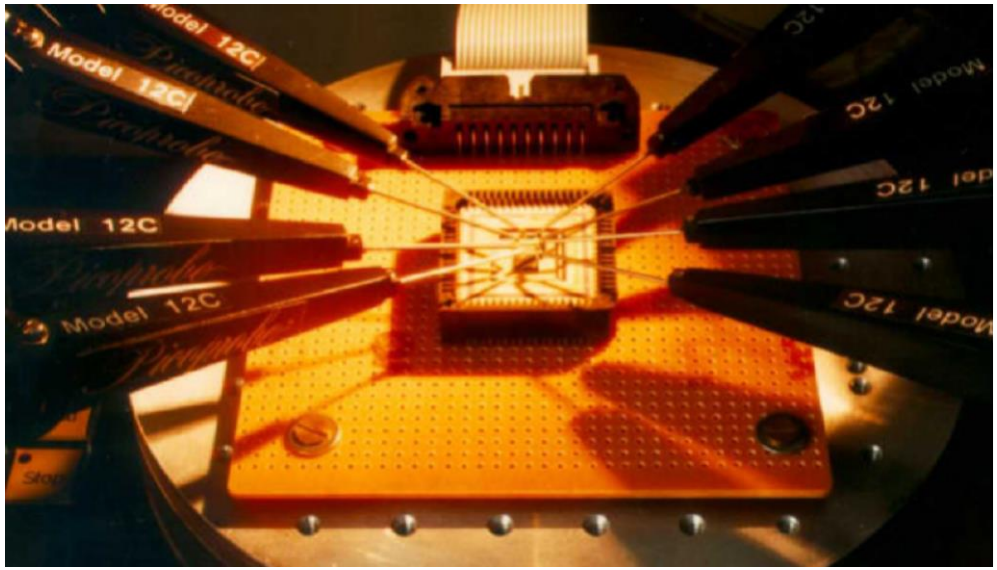




Physical attack: probing

Observe or change the data on the bus while the chip is in operation eg to observe keys

probing with
8 needles



Probing can be done using physical needles (>0.35 micron) or electron beam

Probing countermeasures

- use smaller circuitry

  reducing size makes many physical attacks harder
- hide the bus

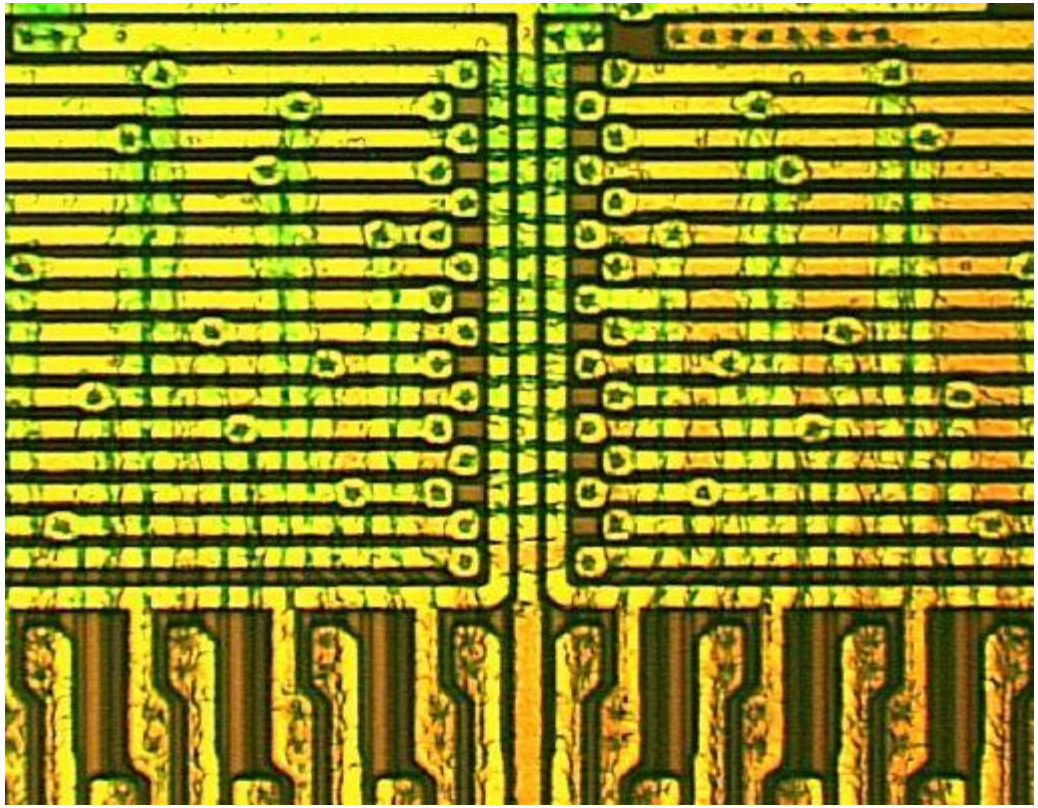
  glue logic, and bus on lower layers of chip
- scramble bus lines

  attacker has to optically reverse engineering this
- encrypting bus
- protective sensor mesh layer

  to prevent access to chip surface

  trend: accessing to chip surface from the back

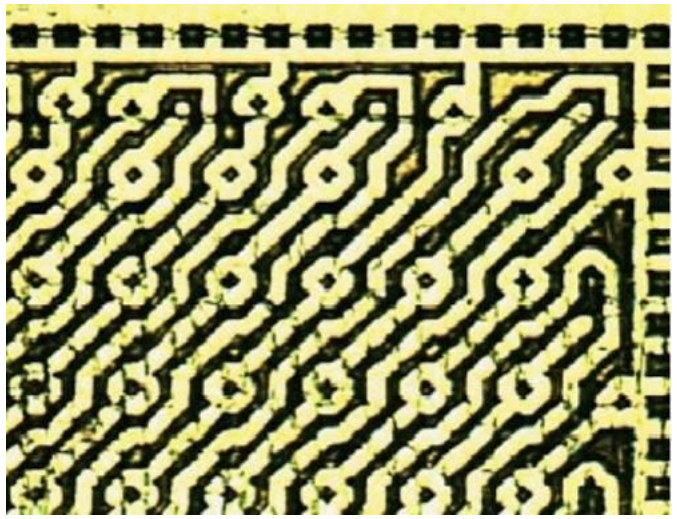
Visual reconstruction of bus permutation

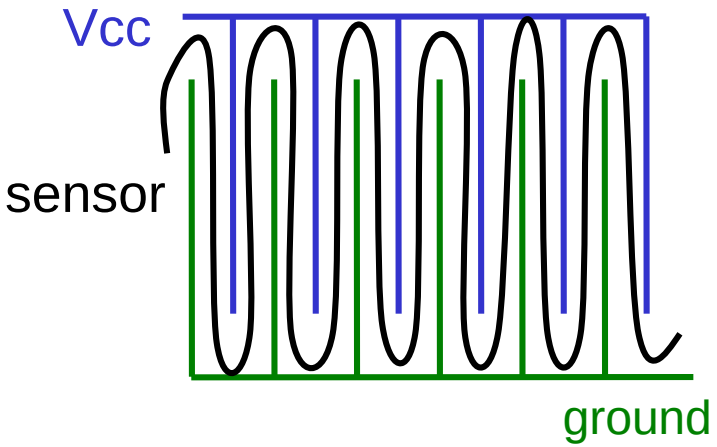


[Source: Oliver Kömmerling, Marcus Kuhn]

Protective sensor mesh

- sensor line checked for interruptions or short-circuits, which trigger alarm: halt execution or erase memory
- but.. external power supply is needed to eg. erase persistent memory
- attacker will fingerprint active countermeasures (eg by power consumption) to interrupt power supply in time



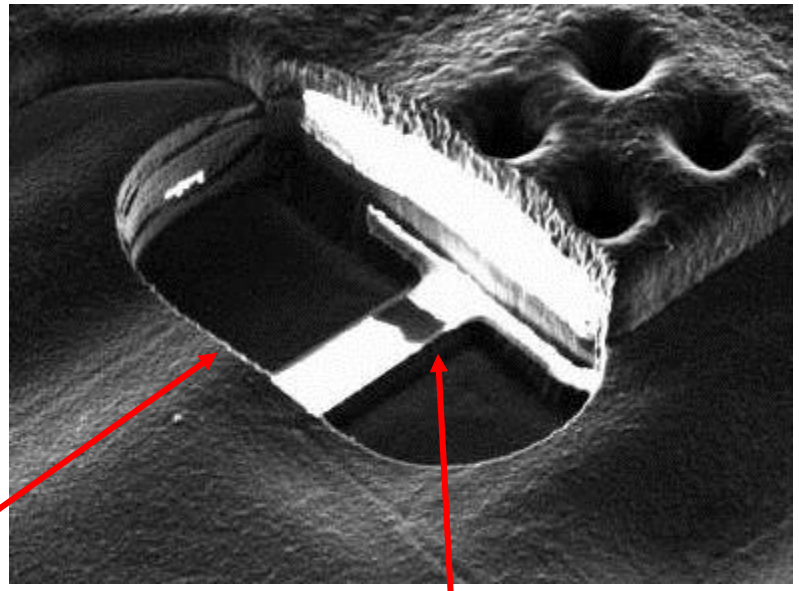


Physical attack: probing

FIB = Focussed Ion Beam

can observe or modify chip by

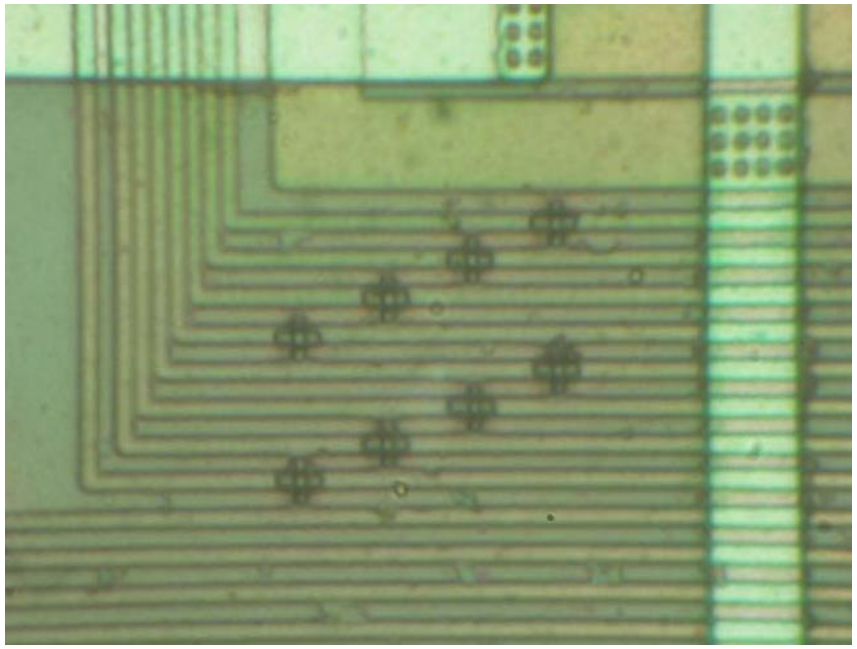
- drilling holes
- cutting connections
- soldering new connections and creating new gates



hole drilled in the chip surface

blown fuse

Using FIB in probing



Fibbing can be used to

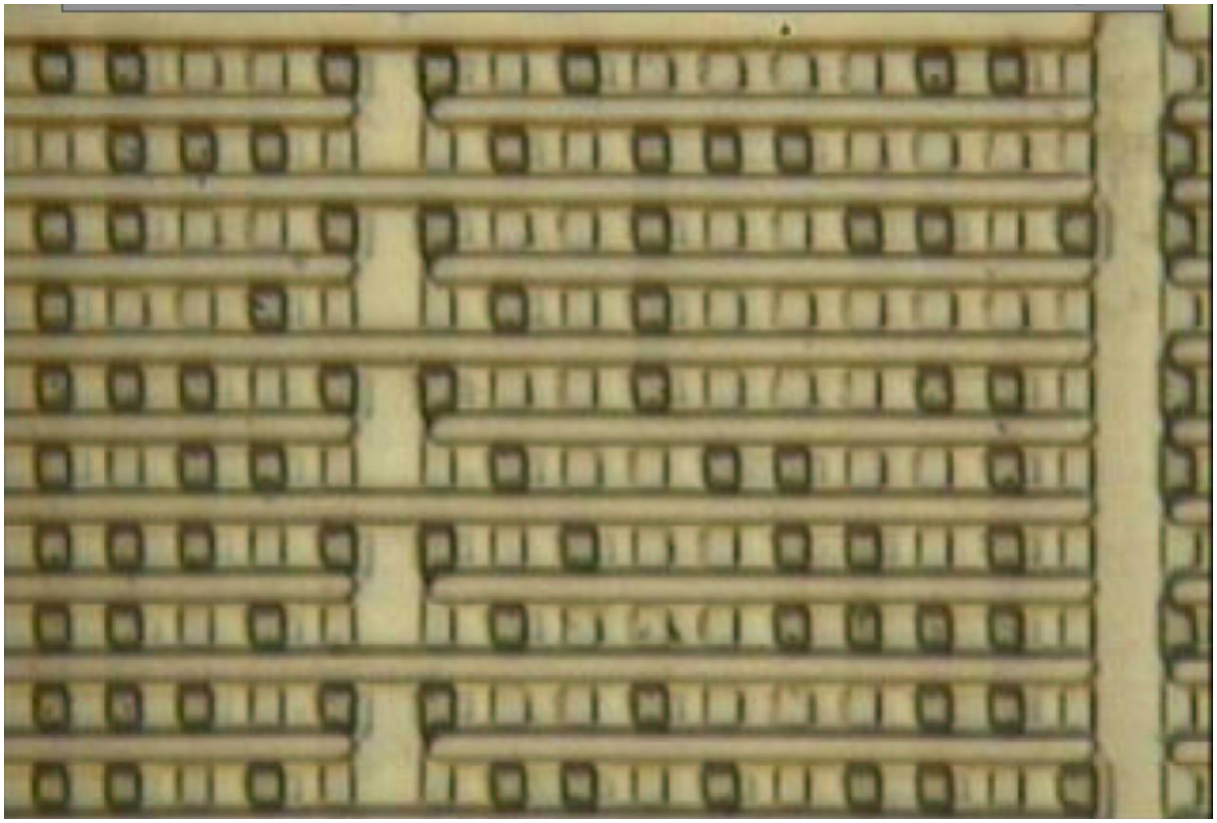
add probe pads for lines too thin or fragile for needles

surface buried lines

- poking holes through upper layers

Physical attack: extracting ROM content

Staining can optically reveal the bits in ROM: dark squares are 1 light squares are 0



[Source: Brightsight]

Physical attack: extracting RAM content

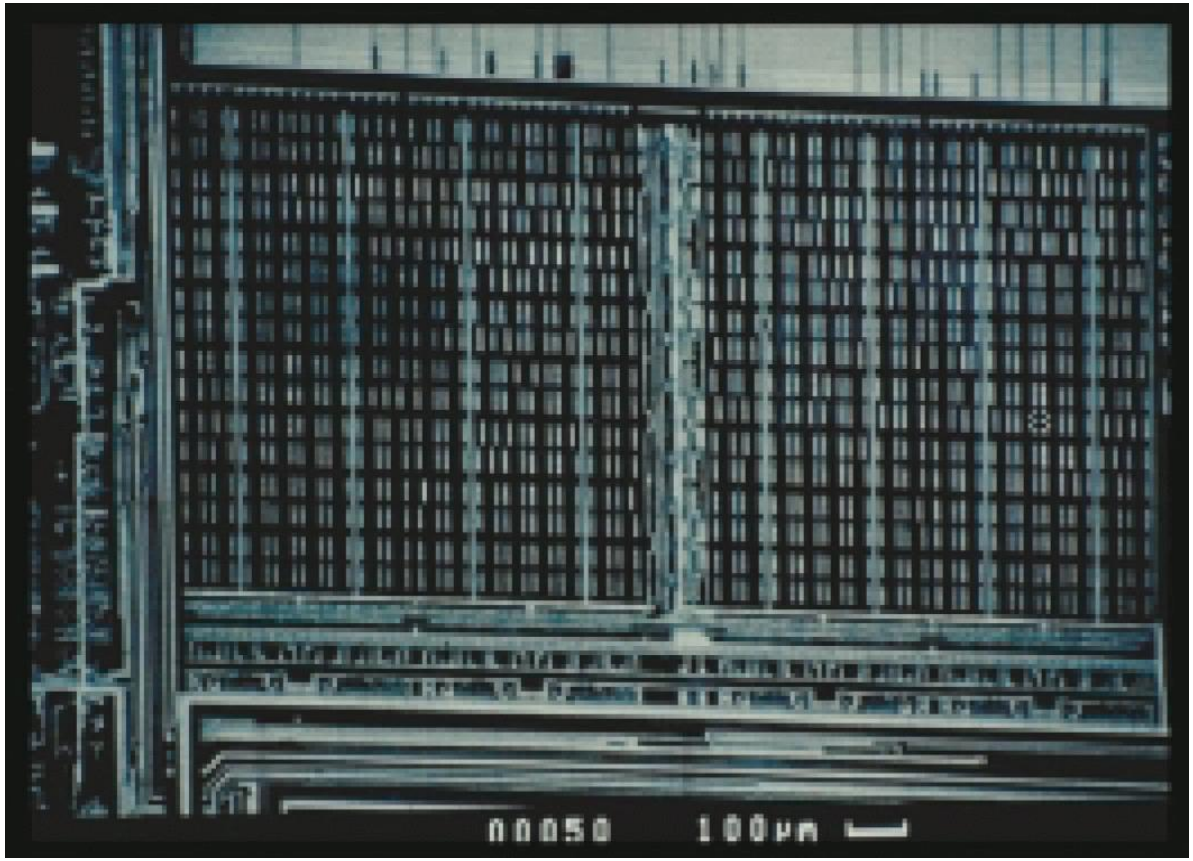


Image of RAM with voltage sensitive scanning electron microscope

memory extraction countermeasures

- obfuscate chip layout
- scramble or encrypt memo
- sensors

  low and high temperatures, light, clock frequency, voltage, …

  But… external power supply is needed to react when intrusion is detected

  Sensors can be destroyed when power is off => they must be tested periodically in normal operation

Conclusions

Things can go wrong at many levels

- card itself, and the crypto, card configuration & protocols,, software
- terminals & terminal software
- organisational

  issuance

  usage

  incl. personnel, procedures, …

What to worry about?

Choosing secure crypto primitives & key lengths is the easy part (3DES, AES, RSA,...)

The real worry is in

1. insecure implementions of these crypto primitives esp. in the face of side-channel attacks
2. insecure protocols using these primitives
3. software bugs in general

   also of software weaknesses wrt fault injections

Smartcards attacks - future

- Moral of the story – it's hard to keep secrets from motivated & well-funded attacker
- Ongoing arms race between smartcard manufacturers and attackers
- Some physical attacks becoming harder, due to improved countermeasures and smaller circuitry
- but side-channel attacks are here to stay
- and increasing complexity of software may introduce more opportunities for logical attacks

Why are smartcards everywhere?

- Cryptography provides a building block for security solutions, but also introduces 2 security problems:

1. key management & distribution

2. who/what do we trust to *store* & *use* crypto keys?

Smartcards provide a possible solution

Humans are incapable of securely storing high-quality cryptographic keys, and they have unacceptable speed and accuracy when performing cryptographic operations.

They are also large, expensive to maintain, difficult to manage, and they pollute the environment.

It is astonishing that these devices continue to be manufactured and deployed. But they are sufficently pervasive that we must design our protocols around their limitations

Kaufman, Perlman, and Speciner