



Hardware Security with Cryptography

Prof. Bart Preneel
COSIC – KU Leuven – Belgium
Firstname.Lastname(at)esat.kuleuven.be
<http://homes.esat.kuleuven.be/~preneel>
December 2014

1

Outline:


securing hardware with cryptography

- secure memory
- cryptographic co-processor
- smart card

2

Memory and Programmable Logic

- most memory is insecure
 - can be read with standard device programmer
- difficult to securely and totally erase data from RAM and non-volatile memory
 - remnants may exist and be retrievable from devices long after power is removed
 - e.g., P. Gutmann, "Secure deletion from magnetic and solid-state memory devices," *Sixth USENIX Security Symposium*, 1996
 - e.g. cold boot attacks

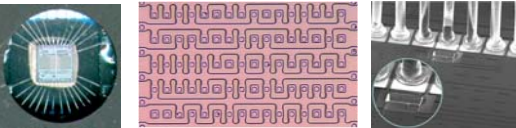


Memory and Programmable Logic (2)

- SRAM-based FPGAs most vulnerable to attack
 - must load configuration from external memory
 - bit stream can be monitored to retrieve data
- protect against I/O scan attacks
 - attacker cycles through all combinations of inputs to determine outputs
 - use unused pins to detect probing
- limit the amount of time that critical data is stored in the same region of memory
 - periodically flip the stored bits

Memory and Programmable Logic (3)

- chip decapping and die analysis attacks
 - attacker can visually recreate contents or modify die (E.g.: to obtain crypto key or remove security bit)
 - tools: chip decappers, Scanning Electron Microscope, Voltage Contrast Microscopy, Focused Ion Beam
 - increased protection: memory scrambling (cf. infra)



Secure Memory: Fuses

- **security fuses and boot-block protection**
 - enabled for "write-once" access to a memory area (OTP or one-time programming)
 - limited functionality but can be useful

| | |
|----------------------------|---|
| memory sector 1: free | 1 |
| memory sector 2: free | 1 |
| memory sector 3: read only | 0 |

security bit

not without weaknesses:
e.g., PIC16C84 attack in which security bit is removed by increasing VCC during repeated write accesses

- PIC Microcontroller Discussion List, "Re: Code protect," Posted April 26, 1995 www.brouhaha.com/~eric/pic/84security.html

Secure Memory: Encryption

- solution: encrypt the memory (authenticated encryption)
- is this secure?
- not necessarily
 - can eavesdrop the data bus
 - can plug the secure memory in any other device

Secure Memory: Encryption (2)

- need to authenticate CPU
- this needs an authenticated key establishment protocol
- such a protocol needs to store a **key at both sides**
- do we have **secure key storage** in the CPU?
 - if yes: why don't we encrypt immediately in the CPU?

Secure Memory: Encryption (3)

- can public key cryptography solve this problem?
 - no, still needs secure key storage in the CPU
- better solution:
 - derive the key from user password
 - limit the number of attempts
 - key still at some stage in software!

Example of “secure” memory: SecuStick

- on plug-in: Windows program pops up and asks for password
- self-destructs if wrong password entered n-many times
- attempt counter stored in flash memory chip

- write-enable pin connected to GND: infinite number of attempts to guess the password
- key logger on PC
- password is checked in software routine on PC: changing return value from "0" to "1" gives full access

“Secustick gives false sense of security
April 12, 2007: <http://tweakers.net/reviews/683>

Cryptographic co-processor

- symmetric cryptography
 - device authentication
 - secure boot with hash value
 - secure boot with MAC algorithm
 - secure update/data authentication with MAC algorithm
- + public key cryptography
 - secure boot with digital signature
 - secure update/data authentication with digital signature
- + hardware RNG
 - key generation and cryptographic protocols

MPC885 Security Core (2003)

- DES Execution Unit (DEU)
 - DES, 3DES with two key (K1, K2, K1) or three Key (K1, K2, K3)
 - ECB and CBC modes for both DES and 3DES
- AES Unit (AESU)
 - AES-128, AES-192, AES-256
 - ECB, CBC, and Counter modes
- Message Digest Execution Unit (MDEU)
 - MD5, SHA-1, SHA-256
 - HMAC
- Supports IPsec, SSL/TLS, SRTP, and 802.11i protocol processing
- SW compatible with MPC184/MPC185 drivers

MPC8272 Security Core (2004)

- Public Key Execution Unit (PKEU) that supports the following:
 - RSA and Diffie-Hellman with programmable field size up to 2048-bits
 - Elliptic curve cryptography, F^{2m} and F(p) modes with programmable field size up to 511-bits
- DES Execution Unit (DEU)
 - DES, 3DES with two key (K1, K2, K1) or three Key (K1, K2, K3)
 - ECB and CBC modes for both DES and 3DES
- AES Standard Unit (AESU)
 - AES-128, AES-192, AES-256
 - ECB, CBC, Counter, and CCM modes
- Message Digest Execution Unit (MDEU)
 - MDS, SHA-1, SHA-256
 - HMAC
- ARC Four Execution Unit (AFEU)
- Random Number Generator (RNG)
- SW compatible with MPC184/MPC185 drivers

Slide 13

Secure boot with hash algorithm

- need OTP in crypto processor for hash value
- cannot update code

14

Secure boot with MAC algorithm

- need secret key in crypto processor
- easy to update code

15

Secure boot with digital signature algorithm

- need authenticated public key in crypto processor
- easy to update code

16

Secure update

- with MAC algorithm: requires shared secret key
- with digital signature: requires public key
- need protection against downgrading

17

How to circumvent secure boot/update?

- software attack after the boot process
- obtain the shared MAC keys
- obtain the signing key (Sony PlayStation)
- modify the public key

18

Software update

- growing importance: apps, O/S, firmware (FPGA), processor microcode
- use secure update
- only allow newer versions to be loaded into product (so attacker can not make revert to old versions with known security flaws)
- sensitive software should perhaps not be released in clear on a web page (could be disassembled and analyzed by attacker)
 - example: firmware for satellite encryption

19

Outline:

securing hardware with cryptography

- secure memory
- cryptographic co-processor
- smart card

20

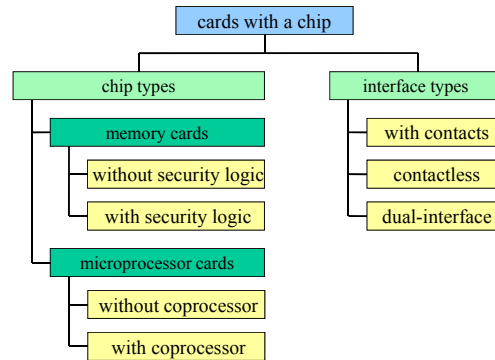
Smart cards

- types of smart cards
- layout and security
- assessment of smart cards

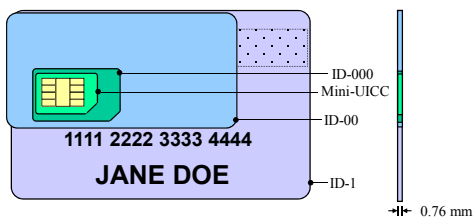
slide credit: Prof. Dr. Andreas Steffen

21

Smart card types



Physical form factors (ISO 7816)




- ID-1 54 x 85.6 mm (ISO 7810 credit card format)
- Visa Mini 40 x 66 mm (Visa/MC credit cards)
- ID-00 33 x 66 mm (mini card, rarely used)
- ID-000 15 x 25 mm (GSM SIM card)
- Mini-UICC 12 x 15 mm (Micro SIM card)

Contactless and dual-interface cards



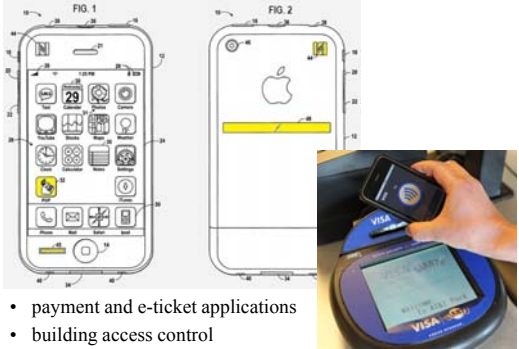
- proximity cards (ISO 14443): distance < 10 cm
- vicinity cards (ISO 15693): distance = 10 cm ... 1 m
- operating frequency: $f = 13.56$ MHz
- products: MIFARE (Philips, et al.), LEGIC (Kaba), PayPass (EMV)

Display Cards



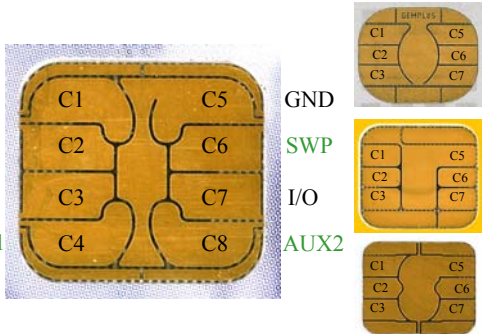
- use as One-Time-Password (OTP) generator
- display account balance
- based on bi-stable e-paper technology
- battery life > 3 years

Near Field Communication (NFC)

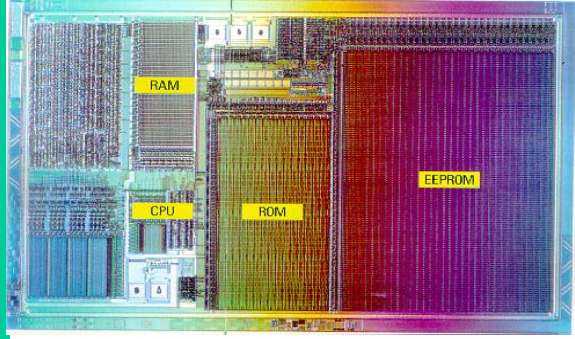


- payment and e-ticket applications
- building access control
- iPhone 5 (Apple patents), Android

Electrical Contacts (ISO 7816-2)

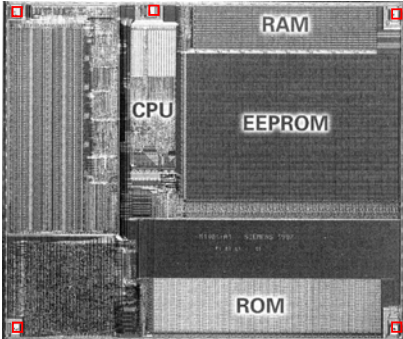


Smart card layout



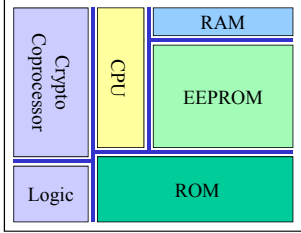
28

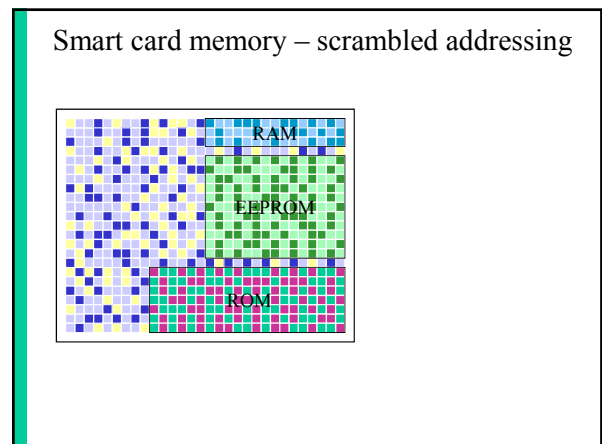
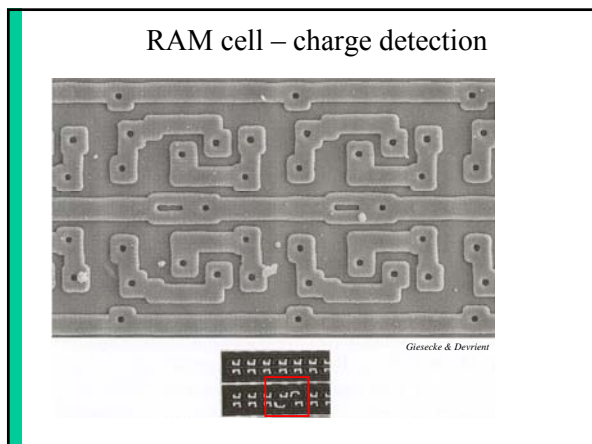
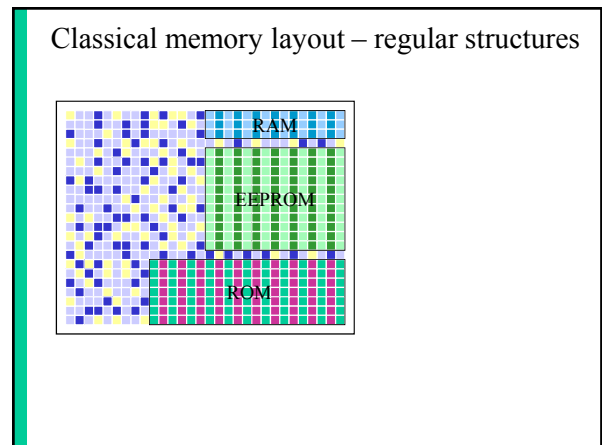
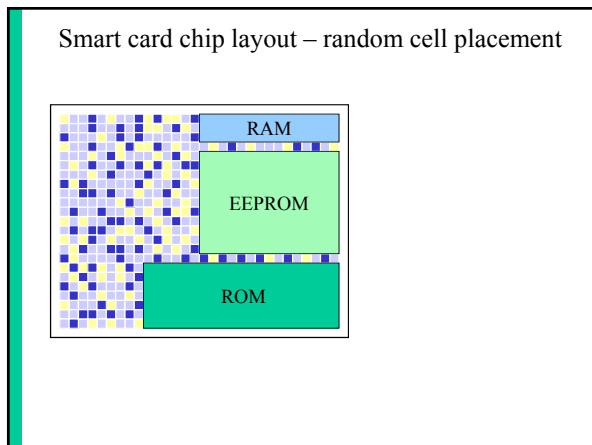
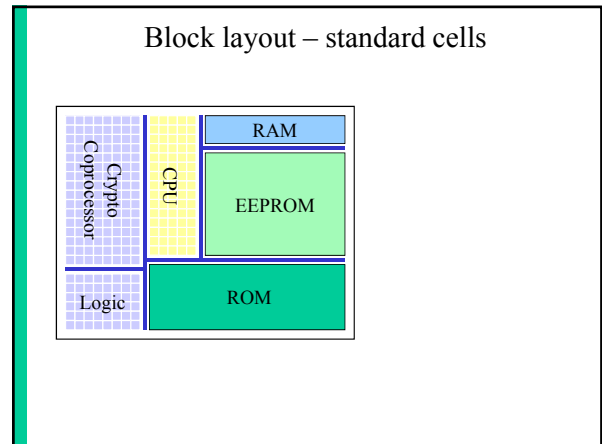
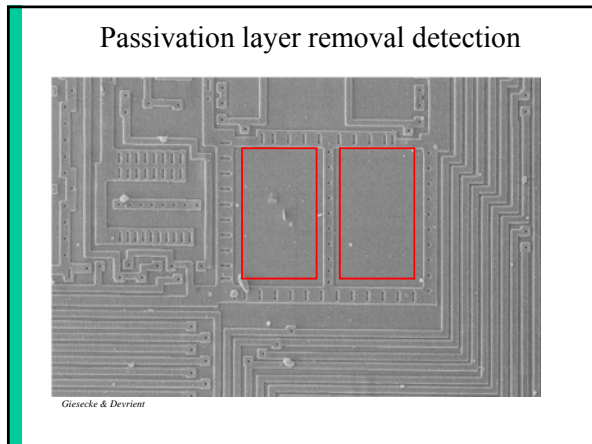
Another smart card



Infineon
SLE 66CX160S

Classical chip layout – floor planning





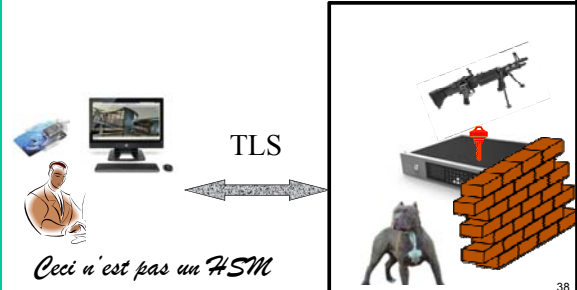
Smart card: discussion

- smart card: simple system that is easy to secure (well-defined perimeter and interface)
- secure O/S and file system
 - access control, counters
 - currently no longer very simple
- how to talk securely to a smart card: secure channel problem
 - how does the user transfer a PIN to the smart card?
 - how does the user know which data is submitted to the smart card to be signed?
 - how does the user know whether the smart card reports an error?
- highly efficient cryptographic coprocessor
 - not always accessible to the programmer
 - communication on standard I/O is very slow
- long tradition for physical protection
 - power and clock are external which makes a smart card more vulnerable

37

Secure hardware

- it is great to have secure hardware
- but how do you authenticate who/what can talk to it?
- one really needs a secure device including secure I/O



38