EMBEDDED SECURITY SYSTEMS 2015

Mirosław Kutyłowski

&grades: 40% lecture, 60% lab

exam, no tests during the course, exam in English unless...

short problems, skills examined not knowledge

lower bound: 40% 3, 50% 3.5, 60% 4, 70 % 4.5 80% 5.0

Objectives

presentation of architecture, limitations and fonalities of embedded systems used in security area C2 developing programming skills concerning cryptographic smart cards and FPGA

---

1. smart cards ≈8 hours
2. telecommunication systems ≈2 hours
2. HSM, TPM, remote attestation ≈4 hours
3. FPGA ≈4
4. sensor systems ≈2 hours
5. RFID tags ≈4 hours
6. CUDA and parallel programming ≈4 hours

---

1. SMART CARDS

cards of no-smart solutions:

- **embossed** - credit cards: reading does not require electricity, manipulation more difficult then with magnetic strip
- **magnetic:** ≈1000 bits, 3 tracks, track 1: 79 6bit chars, track 2: 40 4bit chars, track 3: 107 4-bit chars, limited density (movement in reader against the head), standard data, track 3 for read-write, no physical protection, cheap readers, accidental erasure by a nearby magnet, horror as ATM cards

smart cards: classification

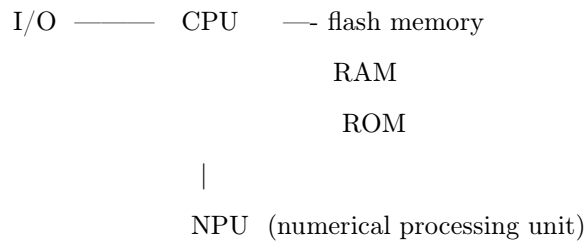
- memory cards (with security logic and without)

  usually memory: non-volatile EEPROM, serial communication, control logic: where you can write. Cheap. E.g. prepayed telephone cards
- processor: with coprocessor or without (as bad as: RSA in 20 minutes)
- contact or wireless
- multimegabyte

Contactless cards:

- energy: inductive (low!)
- small range (typicaly 10 cm)
- a reader may activate it from distance
- response with low energy, recognizable from a short distance only
- memory: kilobytes
- well sealed against corrosion
- main parts:
  - antenna (most area of the card)
  - electronic part: modulation, demodulation, clock generator, voltage regulation, reset generation
  - interface between RF interface and memory chip
  - access logic
  - application data: EEPROM, ROM

processor cards:



contacts:

- 8 fields, normally 6 used (2 for future applications), places for contacts strictly determined in standard
- ground (GND), voltage (Vcc) , I/O, clock (CLK), CTRL, sometimes: Single Wire Protocol (SWP), USB
- easy to destroy
- corrosion, mechanical scratches, not for intensive use

security tokens:

- type 1: USB tokens – contact interface like in USB, insert into a port after breaking out of a card
- type 2: small display (eg. 4 digits). input also possible: e.g. a card with 2 buttons (each one side of the card), battery inside

optical:

- writing technique - like CD (linear and not circular)
- area designated field according to standards, may leave place for contact interface of a chip and magnetic strip, less place for graphical part on the card
- megabytes ($\approx$6MB storage)
- redundancy, therefore not easy to destroy information
- usage: e.g. border control cards (Mexico-USA)

Physical properties:

standard format: 85.6x54 mm (ID-1), other formats for SIM cards (in larger ID-1 cards with stamping),

parameters:

- mechanical robustness (card and contacts)
- temperature resistance
- surfice
- electrostatic discharge
- electromagnetic susceptibility
- ultraviolet radiation
- X-ray radiation

Material: trade-off with different properties

PVC: polivinyl chloride, credit cards, cheap, problems with low and high temperatures, injection molding impossible, lifetime 2 years, cost factor: 1

ABS: mobile, termally stable up to 100 C, laser engraving poor, lifetime 3 years, cost factor: 2

PC: polycarbonate, ID cards, durable, 160 C, problems with hot stamping, lifetime 5 years, cost factor: 7, low scratch resistance,

PET: health cards, mechanical: very good resistance, lifetime 3 years, cost factor: 2.5

Graphical security means:

- Guilloche patterns - fine lines on the surface under the outer transparent foil, in case of any manipulation the pattern destoryed. Technique used on bank notes
- colored signature field - printed paper strip glued to the surface
- microtext - look like simple lines but something printed – used on bank notes, defence against photocopying, readable only under a loupe
- ultraviolet ink

- barcodes (one and two dimensional), two dimensional PDF 417 can encode up to 1000 bytes, error correction codes so that up to 25% of the surface can be damaged (dirty)
- hologram - few companies in the world, cheap, holograms are embossed holograms, holograms reflected in diffuse daylight (some holograms require laser light), permanently bonded to the surface microstructure,
- kinegram - as holograms, show different image from different angles.
- MLI: (multiple laser image) - small lenses, some are blackend by the laser. Looks like a hologram but can contain personalized information (holograms are always the same)
- embossing - like in credit cards (the characters are pressed with a considerable force). Rather old style...
- laser engraving (surface or inside, uder the coat) – equipment fairly expensive, used to personalize cards. However, it is slow (major slow down for production of ID cards). However - a professional forger can make corrections ...
- scratch field – nice for card delivery. The character printed under the coat are not readable even with ultraviolet, infrared light etc
- thermochrome (TC) display: not a real display, but can be reprinted with a special reader. heating a point to 120^oC makes a black dot. Heating the whole strip makes it almost transparent again
- MM (modulated feature) – hidden MM box, invisible, contains control digits for the contents of the magnetic strip. Used by POS and ATM terminals. Control digits computed with MM algorithm

Chip modules:

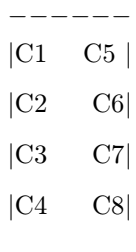
- the chip too fragile and too thick to be laminated on the surface. It is inserted inside
- electrical connections are the problem, automatic bonding of the gold wires to the back of contacts with ultrasonic welding
- Chip-on flex modules, stages of production:
  - tape with empty modules
  - gluing the dice into modules
  - bonding the dice
  - encapsulating the dice
- lead-frame: chip produced together with contacts and the simply inserted by a robot into the card body and glued

Electrical properties:

- 8 connections, 2 auxiliary and can be omitted or used e.g. for USB

  connections:
  - C1: Vcc voltage supply

- C2: RST reset
- C3 CLK clock
- C4 AUX1
- C5 GDN ground
- C6 SPU standard or proprietary use (SWP)
- C7 I/O
- C8 AUX2



- max 60 mA for 5V, max ambient temperature 50 degrees**,** 350 $\boldsymbol{\mu A}$ per megaherz, power consumption too low to cause overheating, power reduction e.g. for SIM in different phases of activity (low if the phone is not transmitting and using cryptoprocesor)
- contact C6 was for EEPROM erasing but not needed anymore, used for Single Wire Protocol
- voltage is a problem: 3V for SIM cards (batteries for smarphones weight optimization), 5V needed for EEPROM erasure. charge pumps applied
- no internal clock supply (potential risk: adversary may increase the clock frequency to create faults, fault cryptanalysis)
- problems with collisions on I/O line (too high currents would destroy interface components)
- protection against out of range voltages, electrostatic charges, precisely defined activationa and deactivation sequences: first ground, then voltage, then clock, warm reset when voltage increases on the reset line

Microcontrollers:

- area: manufacturing costs and durability (bending, torsion), typically 10mm^2, square shape

- must be integrated, “standard components” are not well suited due to size of the resulting circuit,
- native designs are proprietary, even a crime to check the layout
- semiconductor technology -> density increases -> chip area drops . But some problems: error probability, necessity to decrease voltage, ...
- extremely high reliability needed. So behind the “state-of-the-art” which is frequently instable
- memory small (e.g. 100KB), a 8-bit processor ok for less than 64KB, then extensions, usually CISC (complex instruction set computer) - instruction over a number of steps, some based on RISC (reduced instruction set computer), also 32 bit processors that needed also for interpreter based architectures (Java Card)

Memory types:

non volatile:

EPROM - UV erasure, not suited for smart cards,

EEPROM - electrical erasure, cell capacitors, discharged state=0, charged state=1, erase state -> non-erased (single bit), non-erased-> erased: page or sector, both slow, size 1.14 μm, 100.000-1.000.000 erasures, 2-10ms, tunneling efect - if there are electrons on the floating gate then they prevent flow in the substrate

flash - a different technique for writing: hot electron injection, write time „flash”, erase like EEPROM, size 0.47, 10.000-100.000 erasures, very fast writing, lower voltage (12V) than EEPROM (17V), NOR flash: free read of individual cells, but complicated circuits, or NAND flash: dense but reading full blocks

ROM - connections broken – memory via a circuit, irreversible process - one disconnected never can be reconnected, lack of connection = 0, small: 0.54μm area size

volatile:

RAM - transistors, flip-flop, size 1.87 μm area size, erasures - unlimited, write: 70ns

---

auxiliary UNITs:

UART - universal asynchronous receiver transmitter, software solutions would be too slow

USB – USB connection has rigid timing requirements, they cannot be guaranteed by the regular chip, 12 MB/s (Full Speed), CRC and buffers on the endpoints

SWP single wire protocol - communication between SIM and NFC controller concurrently with the regular I/O, data sent with voltage modulation and returned with current modulation- full duplex,

timer - a 16 bit counter (or 16 bit), used for timeout detection, watchdog for security reasons

CRC cyclic redundancy clock, Reed Solomon codes,

- $x = x_1 ... x_k$ is the sequence to be encoded
- $p_x(a) = \sum_{i=1}^{k} x_i a^{i-1}$ polynomial over some finite field
- $c(x) = p_x(a_1)p_x(a_2)...p_x(a_n)$ is the code of x, where a_i is the ith power of the root of degree n. $p_x(a_i) = x_i$ (so this is a systematic encoding)

- $C(x) = x \cdot A$, Vandermode matrix, row 1: 1....1, row 2: $a_1, ... a_n$, row 3: a_1^2, ...,a_n^2, and so on
- properties: distance between the codewords: $n - k + 1$ (this is optimal), since two polynomials of degree k may have only $k - 1$ equal values
- it can correct half of it bits

RNG temperature etc. hard to implement,

pragmatic solutions: PRNG (sometimes poor),

be aware that the algorithm implemented is not original one (e.g. DSA but DSA+LFSR+...),

PRNG: the next value derived with the key from the previous values.

- Round Robin- eg 12 values in a buffer
- testing - NIST tests, good for excluding biased/faulty generators, no security guarantees
- hardware Trojans: faults in the circuitry that are not changing the layout-wires, but e.g. the number of electorns in the substrate (invisible during the audit, but may be used to "break randomness" if the manufacturer knows what are the faulty places

Clock multiplication: external clock cannot have frequency over 5MHz. Internally we can increase it a few times with a multiplication circuit. Potentially: one could adjust the speed to adjust energy usage (problems with intereference of oscillators with the GSM, UMTS communication)

MMU: memory management unit for monitoring boundaries between the application programs (strict separation). must be tailored to the opearting system of the chip

JAVA accelerator: approaches 1) dedicated hardware component, hish speed but takes place, 2) native instructions for java

Symmetric crypto coprocessor: 75 microseconds per DES, 150 per 3DES

asymmetric coprocessor:

RSA up to 2048, problematic key generation, probabilistic time

EC 160-256 bits: create DSA, random numbers problematic

hash functions: SHA, Keccak, SHA1 in PL

memory for keys:

- masterkey, derived keys, dynamic keys (session)
- PIN: master or deriving from master key, PIN updates in different memory, problems of nonuniformity of PIN (no leading zeroes, etc), subclasses where strategy gives higher chances

---

Data

Abstract Syntax notation, ASN.1: basic types (boolean, integer, octet string, bitstring), constructed data types, (page 111)

encoded via TLV structures: (Tag, Length,Value), tags for frequently used data types are in a standard,

- tag:1-2 bytes, the first byte: b8, b7 define the class: univerdsl, spplivstion, context-specific, private class, b6: data object primitive or constructed, b5-b1: tag code, if all ones then the second byte specifies the tag code
- Length: 1-4 bytes:
    - 1 byte: 00 to 7F: encode length 0-127
    - 2 bytes: 1st byte 81, 2nd byte encodes length 0-255
    - 3 bytes: 1st byte 82, 2nd and 3rd bytes encode length 0-65535
    - ...

BER -basic Encoding Rules - list for specific rules

properties: not too flexible, but not too high overhead, much better than XML
