The Ins and Outs of Programming Cryptography in Smart Cards

...and announcing the launch of OpenCard

Pascal Paillier
CryptoExperts

Real World Crypto 2015 – Jan 2015
What are Smart Cards?

Real World Crypto 2015 – Jan 2015
What are Smart Cards?

Command packet: header data Le (APDU-C)
What are Smart Cards?

Command packet: header data Le (APDU-C)
What are Smart Cards?

Command packet: header data Le (APDU-C)

Response packet: data SW (APDU-R)
What are Smart Cards?

Command packet: header data Le (APDU-C)

Response packet: data SW (APDU-R)

black-box oracle
What are Smart Cards?

Command packet: header data Le (APDU-C)
Response packet: data SW (APDU-R)

contactless interface
What are Smart Cards?

Command packet: header data Le (APDU-C)
Response packet: data SW (APDU-R)

dual interface

Real World Crypto 2015 – Jan 2015
Native vs Virtual Applications

Native cards
Native vs Virtual Applications

Native cards
Native vs Virtual Applications

Native cards
Native vs Virtual Applications

Native cards

[Diagram showing hardware components: CPU, UART, Memories (ROM, RAM, NVM, EPROM)]
Native vs Virtual Applications

Native cards
Native vs Virtual Applications

Native cards

HARDWARE

TRNG
CPU
MEMORIES
(UART, CRYPTOPROCESSORS, ROM, RAM, NVM, EPROM)
Native vs Virtual Applications

Native cards

[Diagram showing components such as TRNG, CPU, UART, security features, cryptoprocesors, Memories (ROM, RAM, NVM, EPROM), and HARDWARE]
Native vs Virtual Applications

Native cards
Native vs Virtual Applications

Native cards

![Diagram showing OS related stuff (boot, interrupts) and hardware components including TRNG, CPU, security features, crypto processors, and memories (ROM, RAM, NVM, EPROM).]
Native vs Virtual Applications

Native cards

Diagram showing layers of processing including APDU, OS-related stuff, APIs (1 to n), TRNG, CPU, security features, UART, cryptoprocessors, and memories (ROM, RAM, NVM, EPROM).
Native vs Virtual Applications

Native cards

```
<table>
<thead>
<tr>
<th>API 1</th>
<th>API 2</th>
<th>API 3</th>
<th>...</th>
<th>API n</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRNG</td>
<td>CPU</td>
<td>security features</td>
<td>Memories (ROM, RAM, NVM, EPROM)</td>
<td></td>
</tr>
</tbody>
</table>

HARDWARE

OS

APDU Processing

OS related stuff (boot, interrupts)
Native vs Virtual Applications

Native cards

Real World Crypto 2015 – Jan 2015
Native vs Virtual Applications

VM-based cards

Real World Crypto 2015 – Jan 2015
Native vs Virtual Applications

VM-based cards
Native vs Virtual Applications

VM-based cards
Native vs Virtual Applications

VM-based cards

Real World Crypto 2015 – Jan 2015
Native vs Virtual Applications

VM-based cards

Real World Crypto 2015 – Jan 2015
Native vs Virtual Applications

VM-based cards

JavaCard Framework, APIs, add-on classes

JavaCard VM

OS related stuff (boot, interrupts)

API 1
API 2
API 3
...
API n

PLATFORM

TRNG
CPU
security features

UART
cryptoprocesors

Memories
(ROM, RAM, NVM, EPROM)

HARDWARE

OS

Real World Crypto 2015 – Jan 2015
Smart Card Concepts & Standards
Typical Hardware Architecture
CPU Cores

- The 8-bit era
  - Motorola 68HC05, Intel 8051, AVR AT90
- Then 32-bit RISCs took over
  - ARM7-TDMI, ARM9/11, SmartMIPS
  - Cortex M3, M0

```assembly
0  MOV   R0,  #0x9E
   BL    send_byte
   MOV   R0,  R4
   BL    send_byte
1  B     %B1
   B     %B1

handler_fiq

   LDR   R8,  =0x000F0048   ; SCUINTEN
   LDR   R9,  [R8]
   BIC   R9,  R9,  #0x00000100 ; UART interrupt
   STR   R9,  [R8]
   SUBS  PC,  R14,  #4
```

JMP $
Embedded Cryptoprocessors

All shapes and sizes.
Embedded Cryptoprocessors

Shush! NDA required...
Embedded Cryptoprocessors

Real World Crypto 2015 – Jan 2015
Embedded Cryptoproccessors
Embedded Cryptoprocessors
Embedded Cryptoprocessors
Embedded Cryptoprocessors

AES

AES

AES
Embedded Cryptoprocessors

Binary fields

```
<table>
<thead>
<tr>
<th>a(x)</th>
<th>p(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>b(x)</td>
<td>c(x)</td>
</tr>
</tbody>
</table>
```

Real World Crypto 2015 – Jan 2015
Arithmetic processors
Arithmetic processors

The good, the bad and the ugly.
Arithmetic processors

The good: full set of operations in hardware

- modular additions, subtractions, multiplications
- regular additions, subtractions, multiplications
- variable operand length with automatic adjustment
- extra support like logical operations, modular inverses, exponentiation
- hardware-enhanced side-channel resistance
- operand in shared RAM memory
- fully parallel to CPU
Arithmetic processors

The bad: much less flexible :(

- modular additions, subtractions, multiplications
- variable operand length
- no extra support
- hardware-enhanced side-channel resistance?
- fully parallel to CPU
Arithmetic processors

The ugly: just a
- big Montgomery multiplier with
- coarse-grain scalability
- huge side-channel leakage
- CPU may be idle when co-processing things
Arithmetic processors

Complexity metrics often seem "unnatural"...
Arithmetic processors

Complexity metrics often seem "unnatural"…

\[ x^{p-2} \mod p \] much faster and secure than GCD
Arithmetic processors

Complexity metrics often seem ”unnatural”…

\[ x^{p-2} \mod p \] much faster and secure than GCD

Mandatory re-design of time-critical algorithms such as random prime number generation
Smart cards are a close technology.
Smart cards are a close technology.

You may only purchase semi-open javacards or MultOS cards.
Smart cards are a close technology.

You may only purchase semi-open javacards or MultOS cards.

Significant slow-down factor.
Smart Card Programming in Practice

Smart cards are a close technology.

You may only purchase semi-open javacards or MultOS cards.

Significant slow-down factor

No direct access to CPU or cryptoprocessors
Announcing OpenCard (mid 2015)

- fully, truly open smart card that anyone can program in C and/or native code without NDA
- 32-bit ARM core, ≃600 kB of memory, ≃18 kB of RAM
- native access to DES/3DES, AES and RSA co-processors

Real World Crypto 2015 – Jan 2015
Announcing OpenCard (mid 2015)

- 3rd party extensions downloadable from OpenCard Market
- ideal for programming your own embedded crypto libs and try advanced applications with pairings, lightweight blockciphers, etc.

Check it out, make your own cards and have fun :)