The Ins and Outs of Programming Cryptography in Smart Cards

. . . and announcing the launch of OpenCard

Pascal Paillier

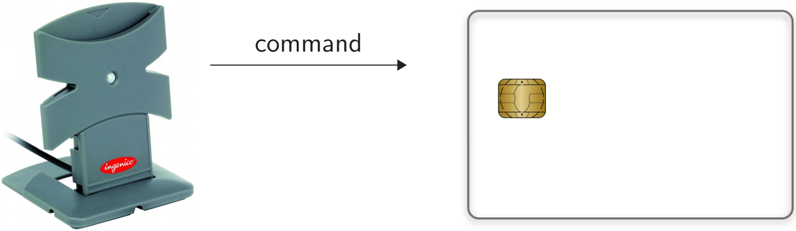
CryptoExperts

Real World Crypto 2015 – Jan 2015

What are Smart Cards?

What are Smart Cards?

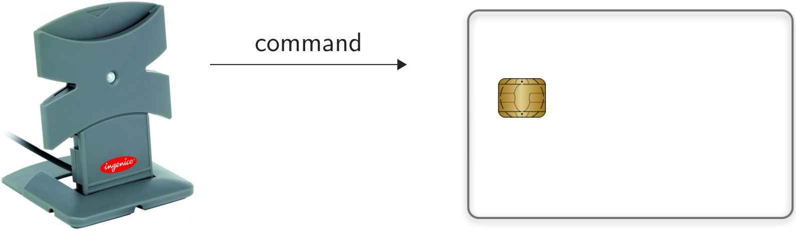
Command packet: | header | data | Le | (APDU-C)



command

What are Smart Cards?

Command packet: header | data | Le (APDU-C)

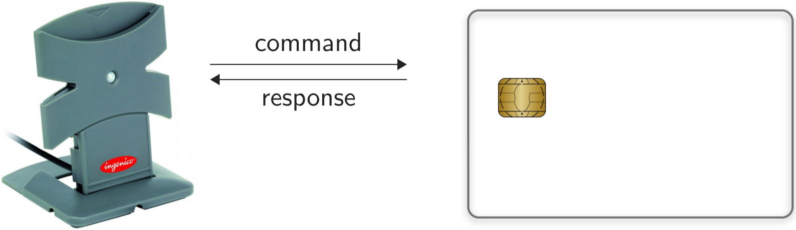


command

internal processing

What are Smart Cards?

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

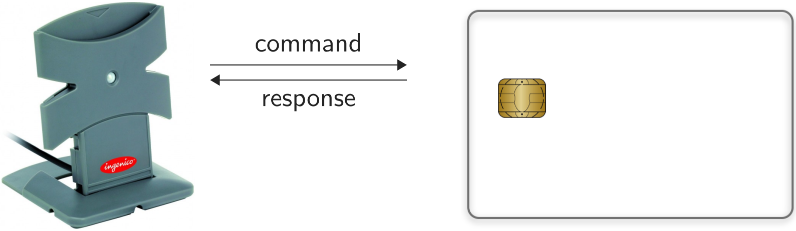


command

response

What are Smart Cards?

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



command

response

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

Native vs Virtual Applications

Native cards

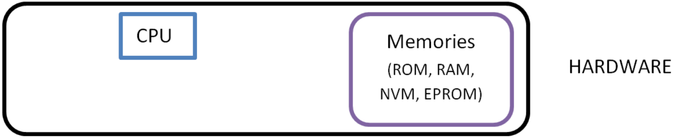
Native vs Virtual Applications

Native cards

CPU
HARDWARE

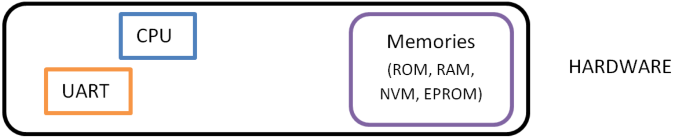
Native vs Virtual Applications

Native cards



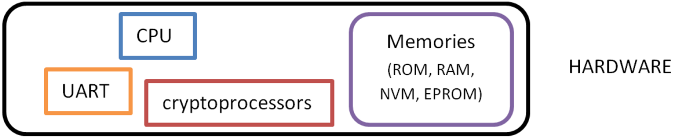
Native vs Virtual Applications

Native cards



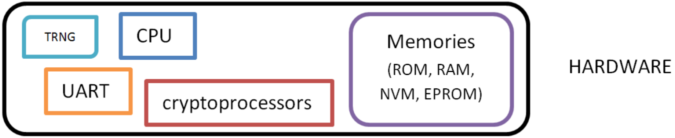
Native vs Virtual Applications

Native cards



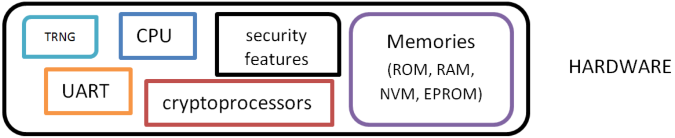
Native vs Virtual Applications

Native cards



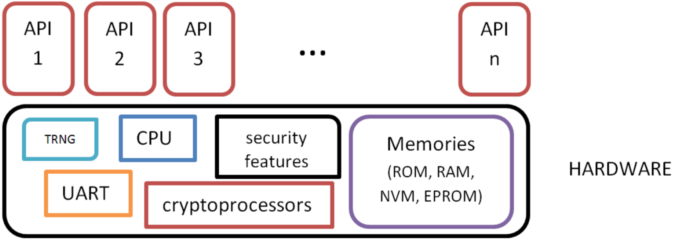
Native vs Virtual Applications

Native cards



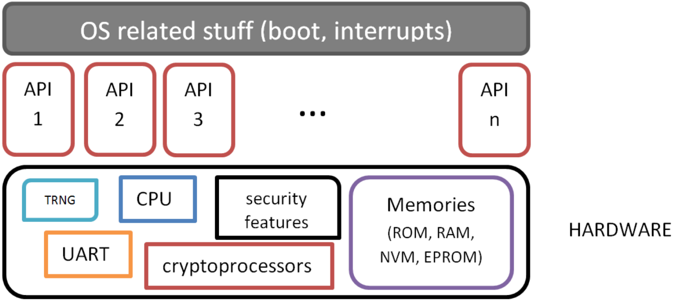
Native vs Virtual Applications

Native cards



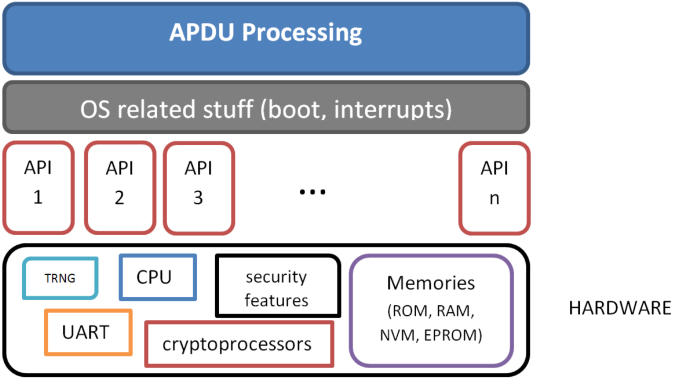
Native vs Virtual Applications

Native cards



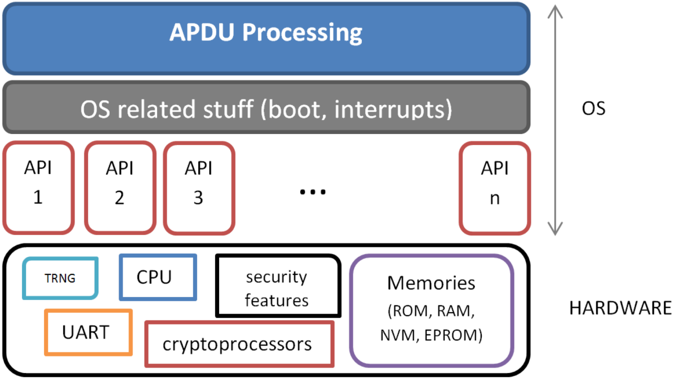
Native vs Virtual Applications

Native cards



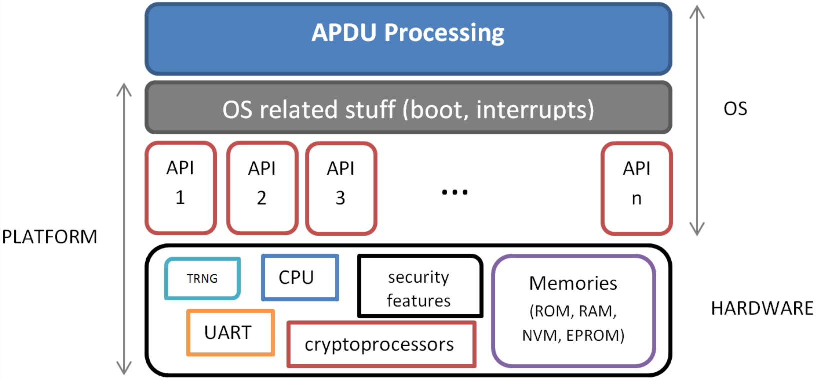
Native vs Virtual Applications

Native cards



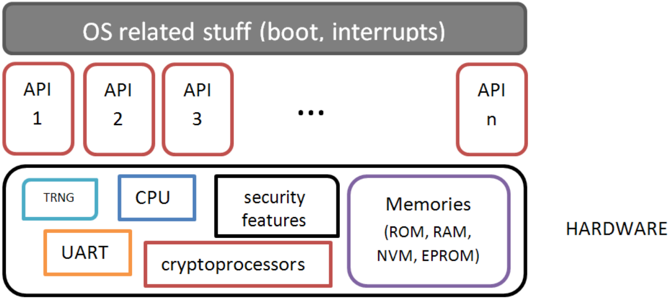
Native vs Virtual Applications

Native cards



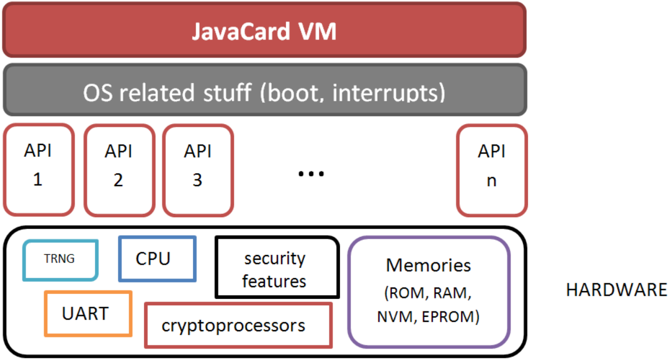
Native vs Virtual Applications

VM-based cards



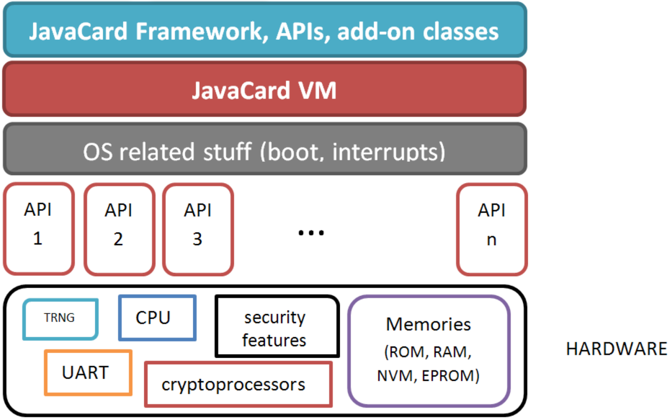
Native vs Virtual Applications

VM-based cards



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

TRNG

CPU

security features

Memories (ROM, RAM, NVM, EPROM)

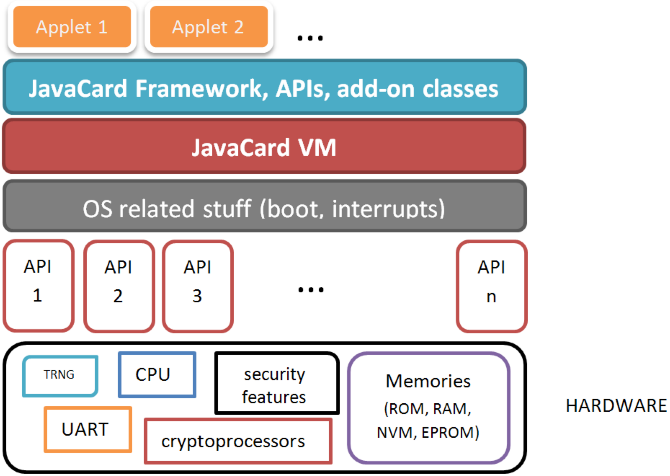
UART

cryptoprocessors

HARDWARE

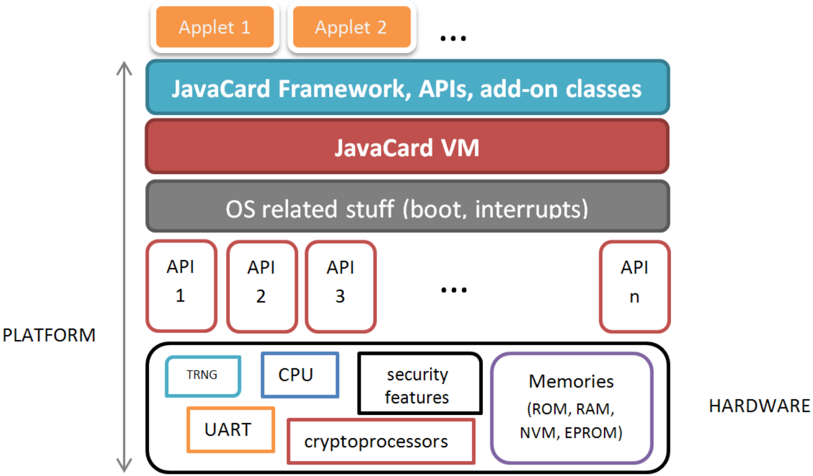
Native vs Virtual Applications

VM-based cards


Applet 1
Applet 2
...
JavaCard Framework, APIs, add-on classes
JavaCard VM
OS related stuff (boot, interrupts)
API 1
API 2
API 3
...
API n
TRNG
CPU
security features
Memories (ROM, RAM, NVM, EPROM)
UART
cryptoprocessors
HARDWARE

Native vs Virtual Applications

VM-based cards



Applet 1

Applet 2

...

JavaCard Framework, APIs, add-on classes

JavaCard VM

OS related stuff (boot, interrupts)

API 1

API 2

API 3

...

API n

TRNG

CPU

security features

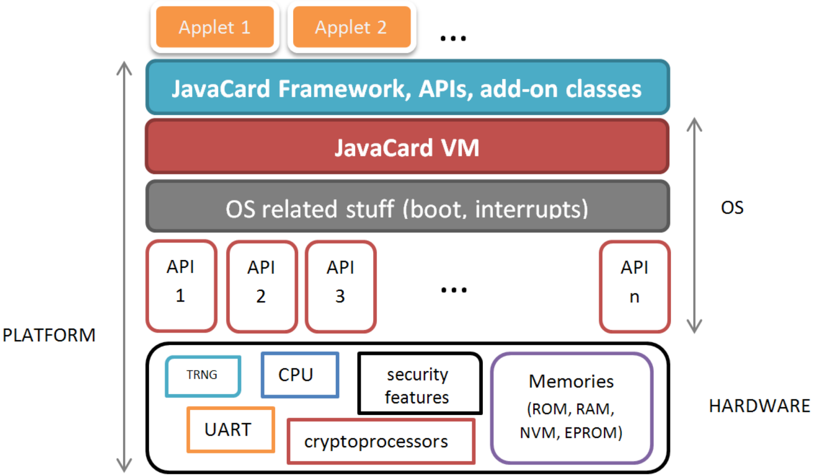
Memories (ROM, RAM, NVM, EPROM)

UART

cryptoprocessors

Native vs Virtual Applications

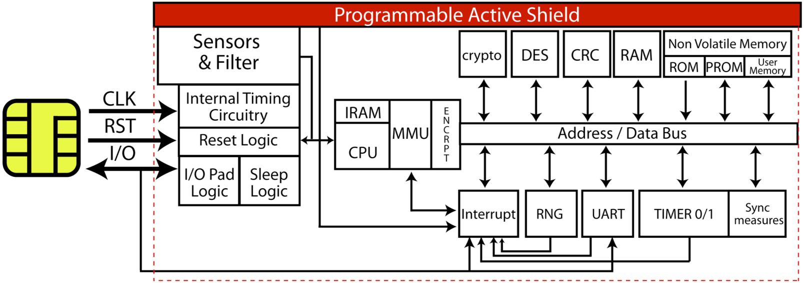
VM-based cards



Smart Card Concepts & Standards

EAL5+
FLASH
EEPROM
TRNG
EID
FAULTS
SIM
TIMING
APPLETS
JAVACARD
CORTEX
READER
APDU
E-PASSPORTS
LOYALTY
EMA
NFCSENSORS
CRYPTOLIBRARIES
ISO14443
SIDE-CHANNELS
CONTACTLESS
GLOBALPLATFORM
EMVCO
COMMONCRITERIA
ISO7816
MULTOS
BIOMETRY
EMV
FIPS140-2
PKCS
ITSEF
DPA
ARM
LEAKAGE
SPA

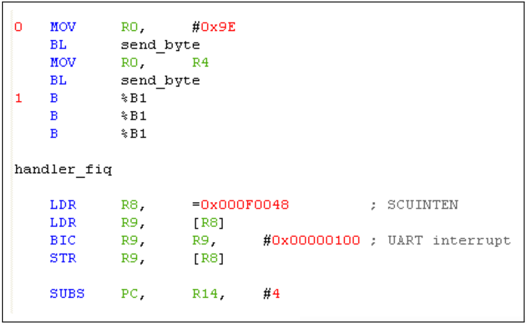
Typical Hardware Architecture



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

```
        MOV 33H, #0
        MOV R0, #30H
again:
        MOV A, @R0
        JZ finish
        MOV C, P
        MOV ACC.7, C
        MOV SBUF, A
        INC R0
        JNB TI, $
        CLR TI
        JMP again
finish:
        JMP $
```



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

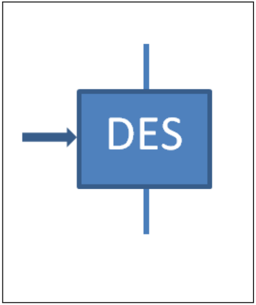
Embedded Cryptoprocessors

All shapes and sizes.

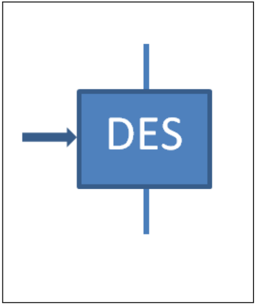
Embedded Cryptoprocessors

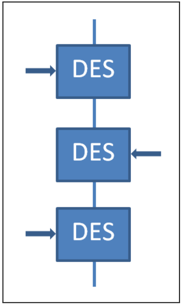
Shush! NDA required...

Embedded Cryptoprocessors

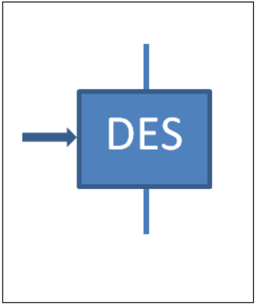


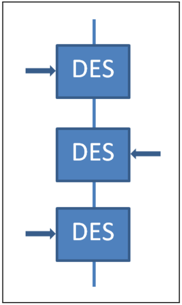
Embedded Cryptoprocessors

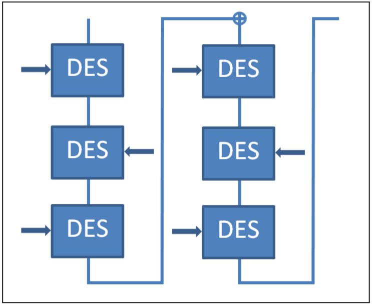




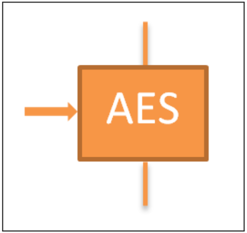
Embedded Cryptoprocessors



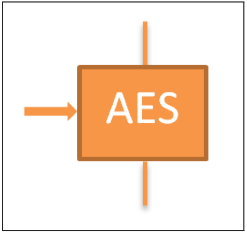


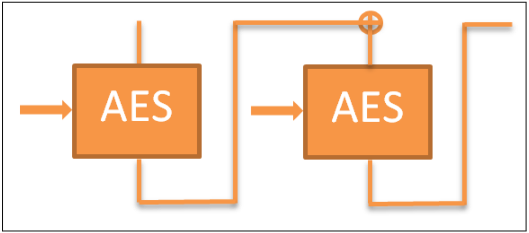


Embedded Cryptoprocessors



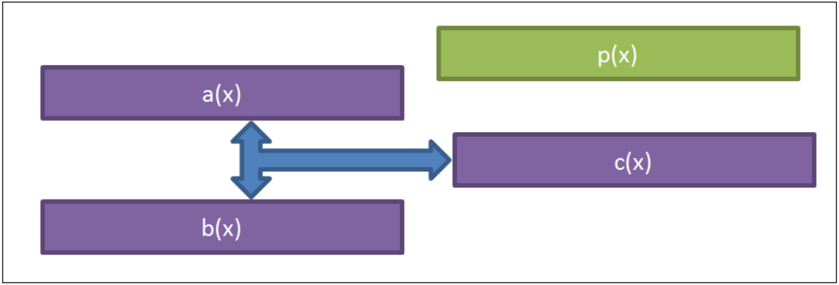
Embedded Cryptoprocessors





Embedded Cryptoprocessors

Binary fields



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 Card Programming in Practice

Smart cards are a **close** technology.

Smart Card Programming in Practice

Smart cards are a **close** technology.

You may only purchase semi-open javacards or MultOS cards

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

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)

opencard

by CryptoExperts

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

Announcing OpenCard (mid 2015)

opencard

by CryptoExperts

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

Launch by Q2 2015 on www.cryptoexperts.com/opencard.
Check it out, make your own cards and have fun :)