Secure Elements and NFC/HCE technologies: the foundations of trusted mobile services based on the emerging Cloud of Secure Elements (CoSE)

Pr Pascal Urien

Pascal.Urien@Telecom-ParisTech.fr

CoFounder of the EtherTrust Company

Agenda

- Introduction
  - Towards Trust for a Connected World
- About Secure Elements
  - Goals and Definition
  - Mask Age
  - Application Age
  - NFC Age
- Payments
- TLS/SSL Secure Elements
  - Concepts
  - Applications
- Ideas About Secure Elements in the Cloud
- Cloud of Secure Elements
  - RACS
- Questions

Introduction

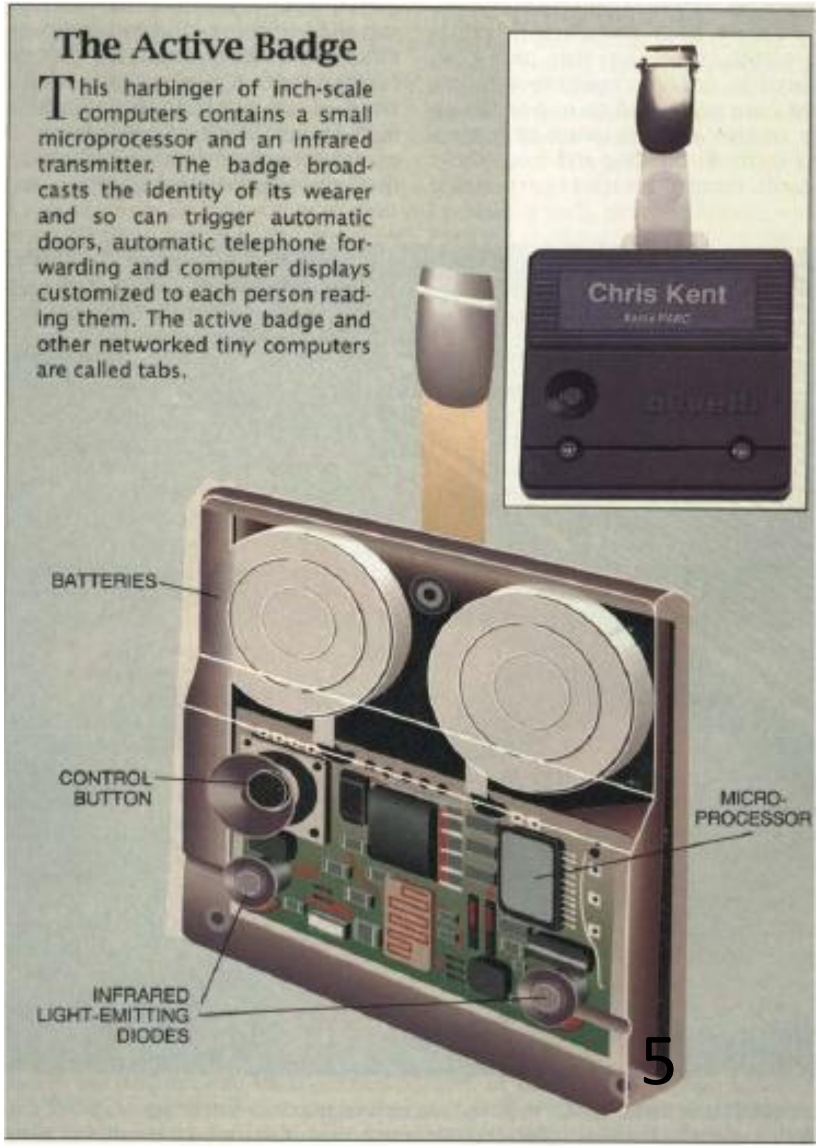
Some thoughts about the emerging connected world…

Three Information Ages

- The Manuscrit Age
  - *De bello Gallico*, written by Gaius Julius Caesar, 50 BCE
- The Printing Age
  - Johannes Gutenberg, 1440
- The Connected Age
  - 21^{st} century
  - Anytime, Anywhere

Mark Weiser, The Connected World Premise, 1991

- The Computer for the 21st Century, **Mark Weiser**, Scientific American September 1991
  - The technology required for ubiquitous computing comes in three parts: cheap, low power **computers** that include equally convenient displays**, software** for ubiquitous applications and a **network** that ties them all together
  - Even today, although active badges and self-writing appointment diaries offer all kinds of convenience, **in the wrong hands their information could be stifling**…
  - Fortunately, **cryptographic techniques** already exist to secure messages from one ubiquitous computer to another and to safeguard private information stored in networked systems.



Internet Services go to Mobiles…

Company	**2013**
Samsung	299,794.9
Apple	150,785.9
Huawei	46,609.4
LG Electronics	46,431.8
Lenovo	43,904.5
Others	380,249.3
Total	**967,775.8**

Smartphones Gartner (February 2014)

Operating System	**2013**
Android	120,961.5
iOS	70,400.1
Microsoft	4,031.8
Others	41.6
Total	**195,435.0**

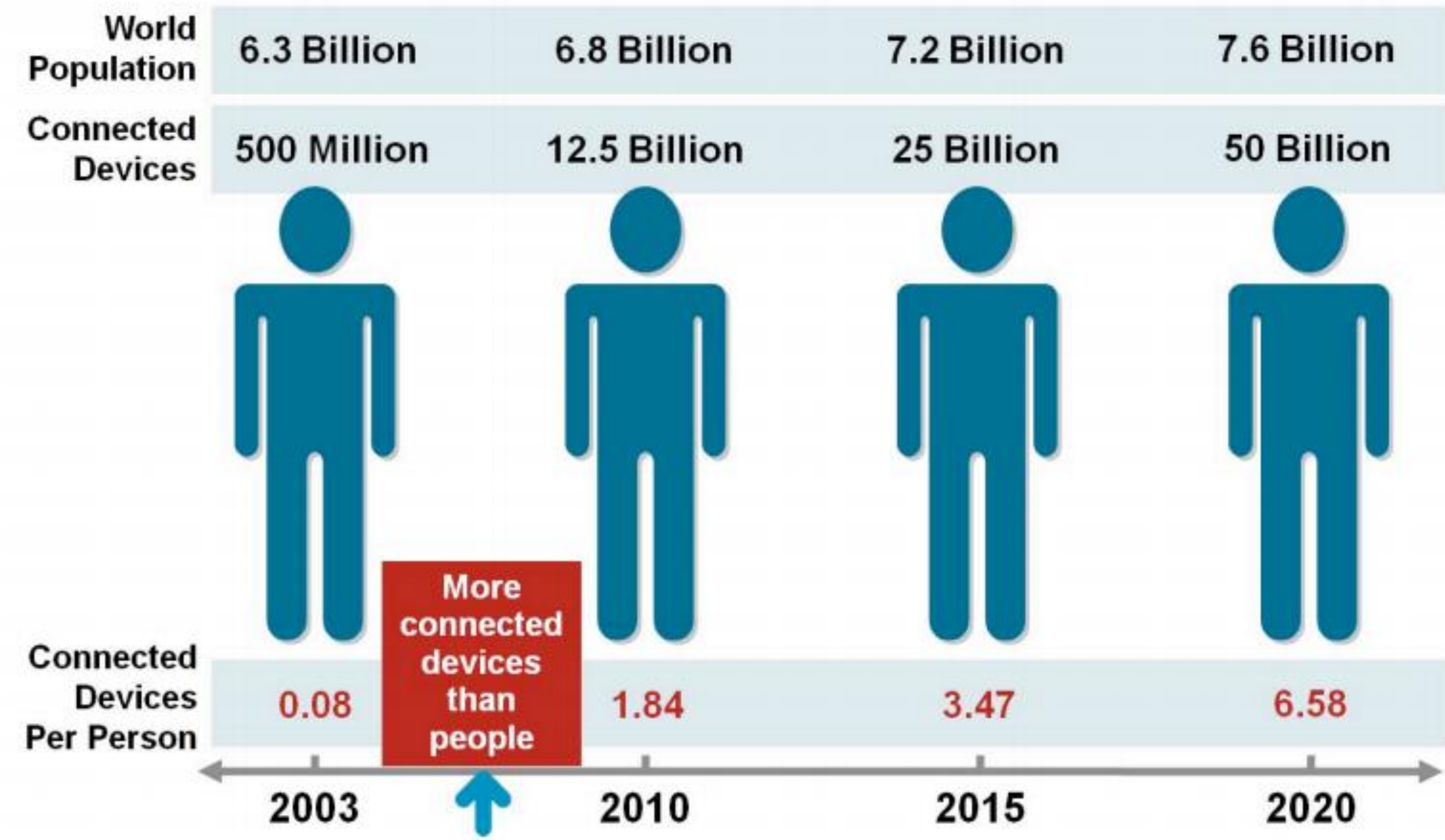
Tablets Gartner (February 2014)

x 1000

According to Gartner 2014
315,967.5 PC sold in 2013

One in three mobile phones to come with NFC by 2017, Berg Insight, 2013

The Emerging Connected World



Source: Cisco IBSG, April 2011

The Internet of Things, How the Next Evolution of the Internet Is Changing Everything,
Dave Evans, Cisco White Paper 2011

About Secure Elements

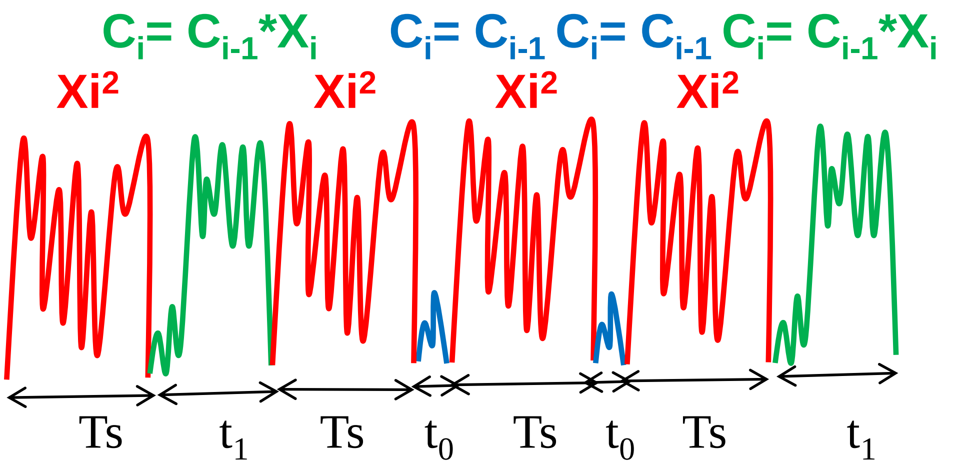
Trust for Security

Tamper Resistant Devices

Introduction to secure microcontroller 1/3

Single Protocol Attack (SPA)

- $C = M^e = M * M * \ldots * M$ (e operations)
- $e = e_0\, 2^0 + e_1\, 2^1 + e_i\, 2^i + \ldots + e_{p-1}\, 2^{p-1}$, $e_i = 0$ or 1, $d_i = e_i\, 2^i$
- $C = M^{d0} * M^{d1} * M^{di} * \ldots * M^{dp-1}$
  - $C_0 = M_0 = M^{d0} =$ 1 or M
  - $C_i = C_{i-1} * M^{di}$
  - $C = C_{p-1}$



- Algorithm
  - Begin i=0
    - $X_0 = M$
    - $C_0 = X_0^{\ d0} =$ 1 or M, this calculation needs a time T_0
  - Loop i<p
    - $X_i = X^2_{i-1} = X_{i-1} * X_{i-1}$, this calculation needs a time Ts
    - If $e_i=0$ Then $C_i = C_{i-1}$, needs a "short" time t_0
    - If $e_i=1$ Then $C_i = C_{i-1} * X_i$, this this calculation needs a "long" time t_1

Works for RSA, DH, ECC

- $T_{RSA} = T_0 + Ts + t_{e1} + Ts + t_{ei} + \ldots + Ts + t_{ep-1}$

Introduction to secure microcontroller 2/3 Differential Powered Attack (DPA)

- Paul C. Kocher, Joshua Jaffe, Benjamin Jun: **Differential Power Analysis**. CRYPTO 1999: 388-397
  - Covariance, $cov(X,Y) = \sigma_{X,Y} = E(XY) - E(X)E(Y)$
  - Correlation coefficient, $\rho_{X,Y} = \sigma_{X,Y} / \sqrt{V(X)V(Y)}$, $\rho_{X,Y} \in [-1, 1]$
  - $E(XY) = E(X)E(Y) + \rho_{X,Y} \sqrt{V(X)V(Y)}$
  - $E(XY) = E(X)E(Y) + \rho_{X,Y}\, \sigma(X)\, \sigma(Y)$
- Let's assume :
  - A key domain of 2^p values, $i \in [0, 2^p -1]$
  - A physical effect, such as power consumption, with an input value k, $X_i(k,t)$
  - A function Y correlated to the secret key i, and working for all input value k, $Y_i(k)$
  - and for each key i, $<Y_i(k)>_k=0$
  - For each wrong key
    - $\rho_{X,Y}=0$, $<X_i(k,t).Y_i(k)>_k = <X_i(k,t)>_k <Y_i(k)>_k =0$
  - For the right key (j), $\rho_{X,Y}$ #0
    - $<X_j(k,t).Y_j(k)>_k = \rho_{X,Y}\, \sigma(X)\, \sigma(Y)$

Introduction to secure microcontroller Fault Injection 3/3

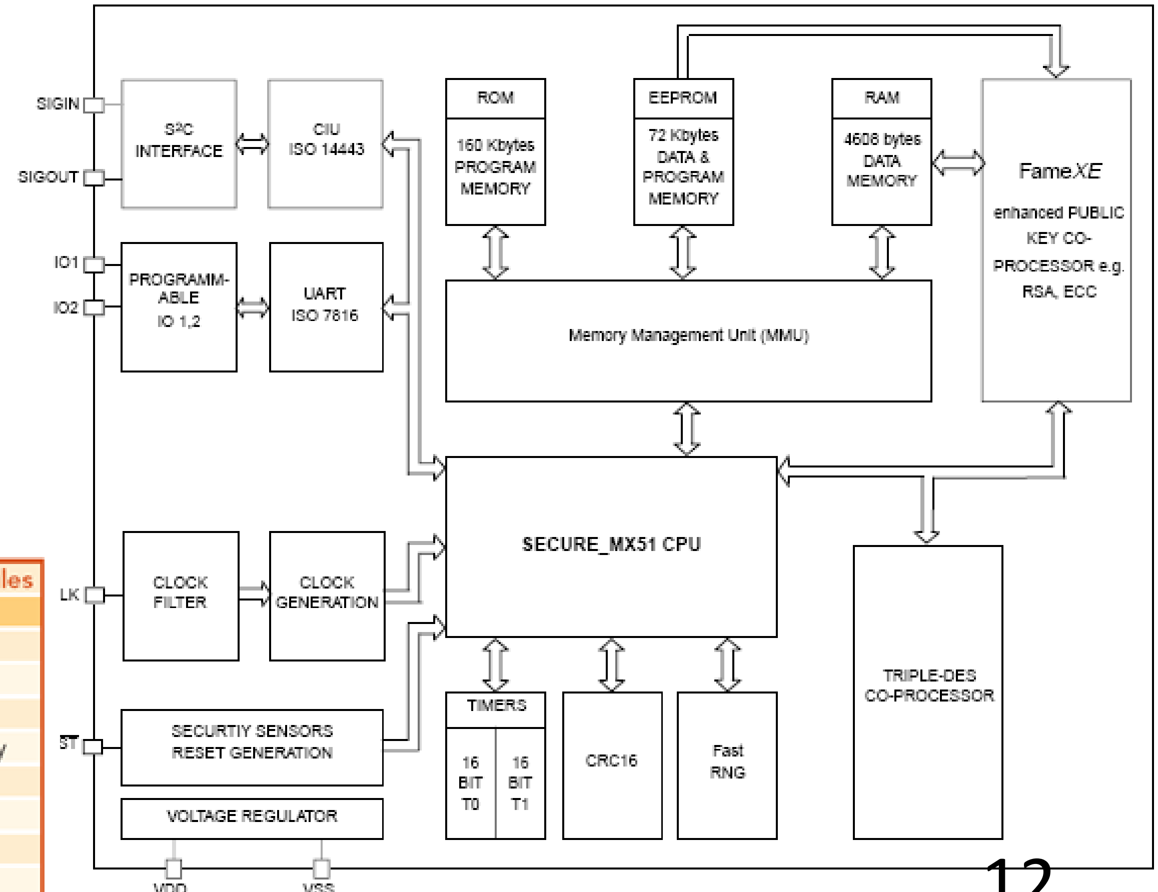
- Dan Boneh, Twenty years of attacks on the RSA cryptosystem, Notices of the American Mathematical Society (AMS), Vol. 46, No. 2, pp. 203-213, 1999
- Example, the ***Bellcore Attack***
  - $N = pq$
  - Chinese Remainder Theorem $E = x^s \bmod pq = a\,E1 + b\,E2$
    - $a=1 \bmod p$, $a=0 \bmod q$, a is a multiple of q
    - $b=1 \bmod q$, $b=0 \bmod p$, b is a multiple of p
    - $E1 = x^s \bmod p$, $E2 = x^s \bmod q$
  - If a ***computing fault*** E1' is created in place of E,
    - If E1-E1' is not divisible by p, then
    - $\gcd(E'-E, N) = \gcd(a(E1'-E1), N) = q$

What is a Secure Element ?

A Secure Element (SE) is a Secure Microcontroller, equipped with host interfaces such as ISO7816, SPI or I^2C .

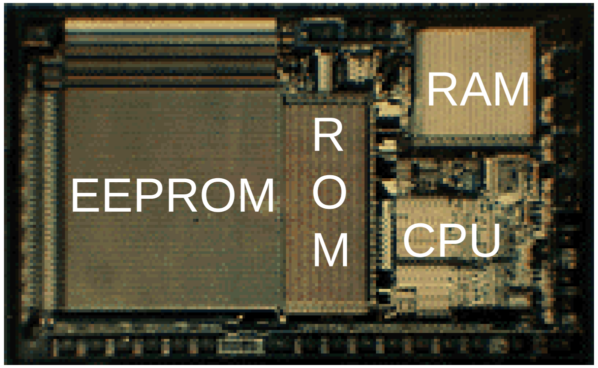
OS JAVACARD JCOP
GP (Global Platform)
ROM 160 KB
EEPROM 72 KB
RAM 4KB
<u>Crypto-processor</u>
3xDES, AES, RSA, ECC
Certification CC EAL5+
Security Certificates EMVCo

	NFC secure modules
Product features	PN65L
Embedded NFC IC	PN532
Available host interfaces	serial, SPI, I²C
Embedded Secure IC	P5CN072
OS for secure device	JCOP or 3rd party
Stacked passive component IC	yes
Package thickness	1.2 mm
Package size	7x7 mm²
Package type	HLQFN48

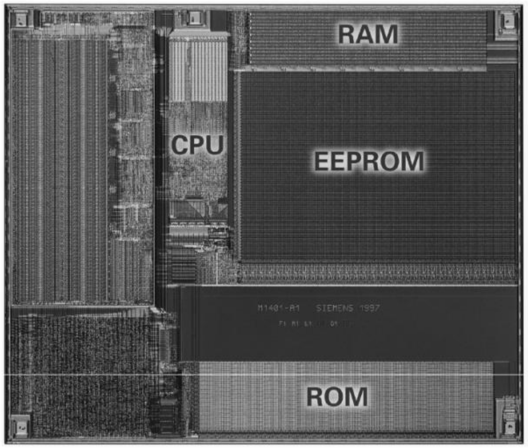


EXAMPLE: NXP PN532

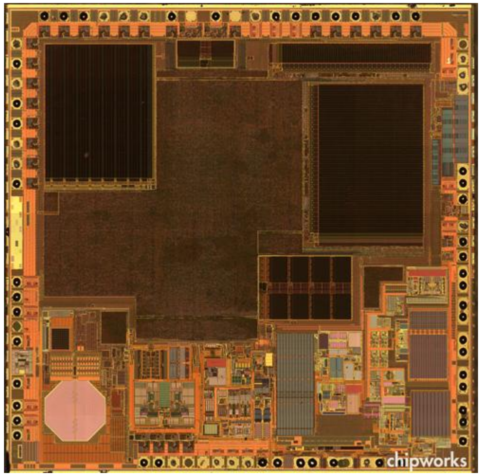
Example of Secure Elements



1988, the 21 (BO') chip
First Bank Card Chip



Siemens SIM chip, 1997



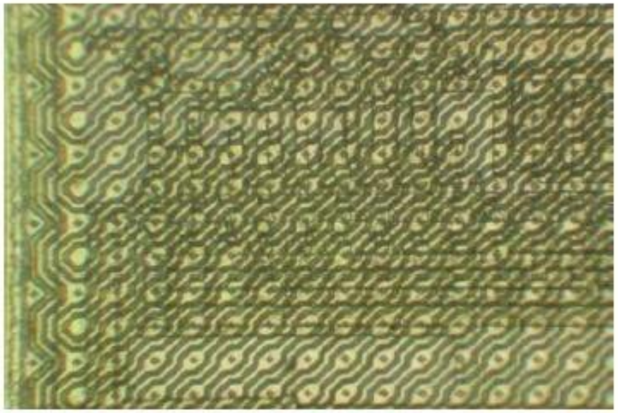
PN65 NFC Controller, 2010

Maxwell's equations are not secure.

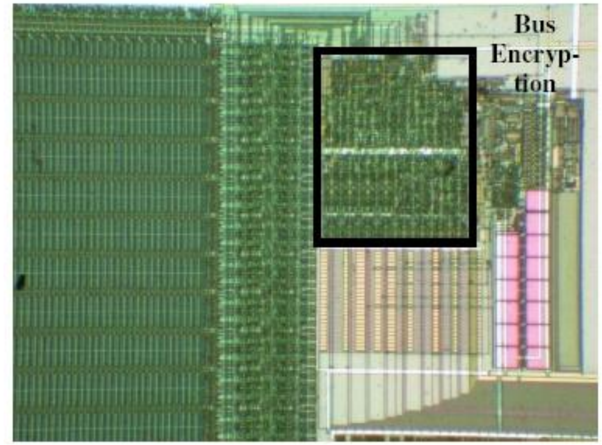
Electronics chips work with these equations.

Therefore secure microcontroller is a construction

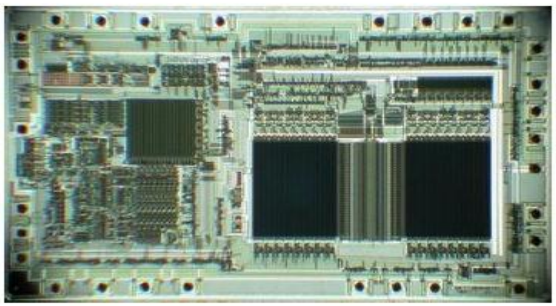
Secure Hardware



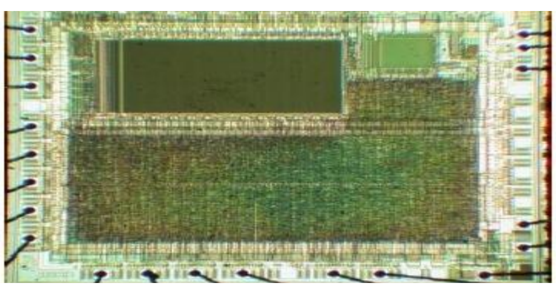
Top metal sensor on ST16 smartcard



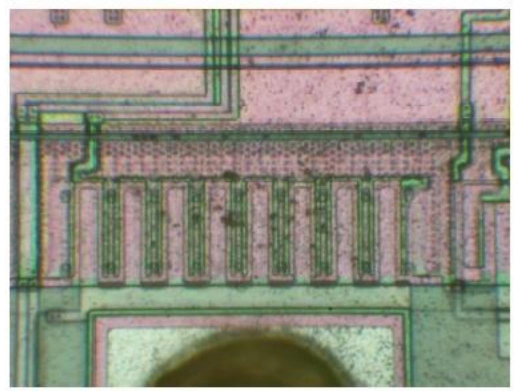
Hardware bus encryption module in Infineon SLE66 family smartcard chip



MC68HC705PA microcontroller with clearly distinguishable blocks



SX28 microcontroller with ‘glue’ logic design



Second metal layer and polysilicon layer on microchip PIC16F877A microcontroller

Top metal layer on microchip PIC16F877A microcontroller

Some figures for the ST22 secure microcontroller

Algorithm	Function	Time
RSA 1024 bits	Signature with CRT	79.0 ms
	Signature without CRT	242.0 ms
	Verification (e=0x10001)	3.6 ms
RSA 2048 bits	Signature with CRT	485.0 ms
	Signature without CRT	1.7 s
	Verification (e=0x10001)	11.0 ms
DES	Triple	18 µs
	Single	8 µs
SHA-1	512-bit Block	194 µs
AES-128	Encryption including subkey computation	85 µs
Key generation	1024 bits key 2048 bits key	2.7 s 23.1 s

The Three Ages of Secure Elements

The Mask Age

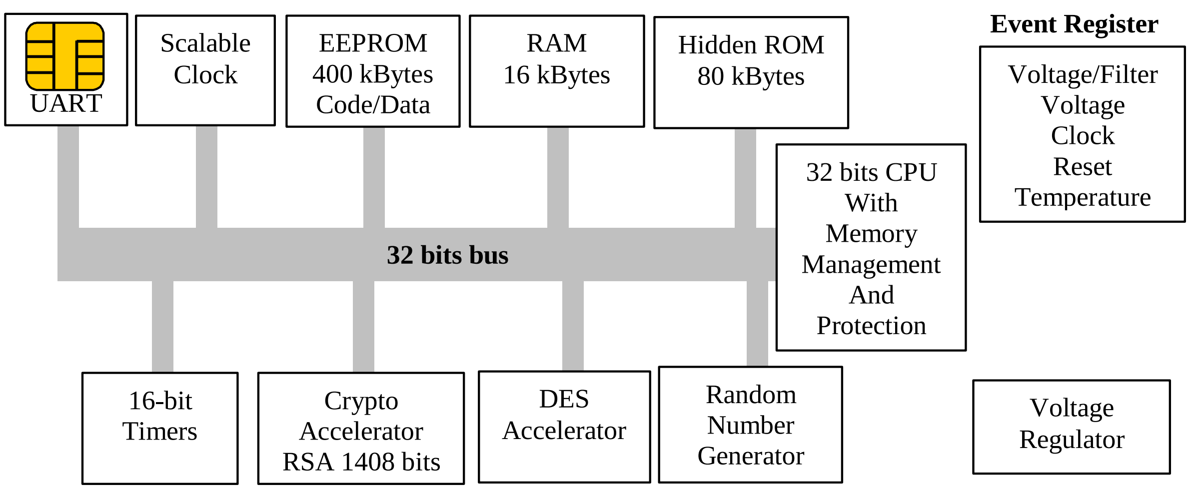
The Application Age

The NFC Age

THE MASK AGE

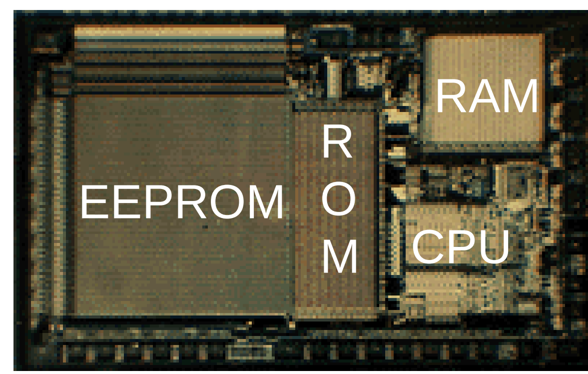
The Mask Age

- The Self-Programmable One-chip Microcomputer (SPOM) was invented by the end of the seventy years; it is often called "smartcard".
  - It is a secure micro-controller.
  - The system is protected by physical and logical countermeasures
- The operating system is fitted for each application.
  - **The OS is buried during the manufacturing process in the ROM, and for this reason is usually referred as a "mask"**



The Mask Age Milestones

- 1980, First BO' French bank card chip, from CP8
- 1988, SIM card specifications
- 1990, First ISO7816 standards
- 1991, First SIM devices
- 1995, First EMV standards



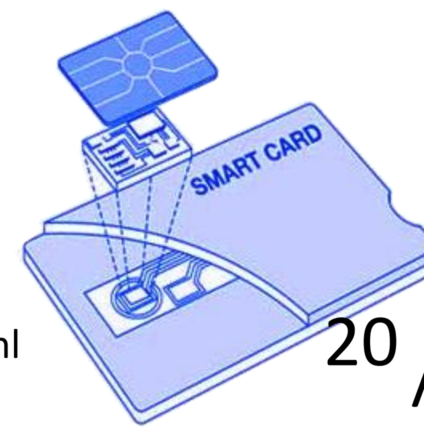
1988, the 21 (BO') chip

What is a Smartcard ?

- A smartcard is a SPOM
  - Self Programmable One Chip Microcomputer, born in 1980
  - 7 billions smartcards produced in 2012
  - 0,6 billion of contactless devices (*)
- Tamper resistant device
  - A Secure Microcontroller
  - Security is enforced by physical and logical countermeasures
- Typical chip area 5mm x 5mm
- Memories size
  - ROM 28 - 256 KB Area Factor 1
  - E^{2}PROM 64 - 128 KB Area Factor 4
  - RAM 4 - 8 KB Area Factor 16

- CPU
  - Classical 8 bits processors, 1 - 3 MIPS (Clock 3.3 MHz)
  - 32 bits RISC processors
- Communication port
  - ISO7816 serial link 9600 to 230,400 bauds
  - USB (ISO7816-12), 10 Mbit/s
- Binary Encoding Rules
  - A five bytes Header
    - CLA INS P1 P2 P3
  - An optional payload of P3 (LC) bytes
  - An optional response of P3 (LE) bytes,
    - which ends with a two bytes status word SW

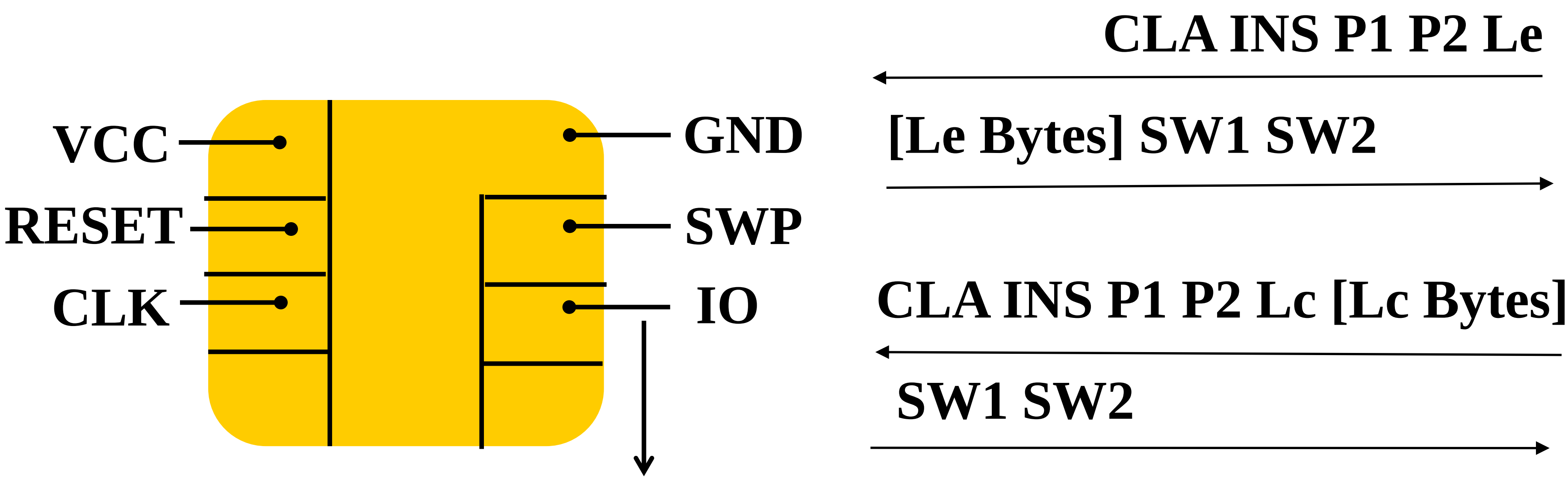
ISO 7816-4 commands are called APDU, their maximum size is about 256 bytes.



* http://www.eurosmart.com/index.php/publications/market-overview.html

What a smartcard does (*) ,

- The five operations of a smartcard are :
  - 1-input data, 2- output data, 3- read data from non volatile memory (NVM), 4- write or erase data in NVM, 5- compute a cryptographic function.”



ISO 7816-2 Serial Link

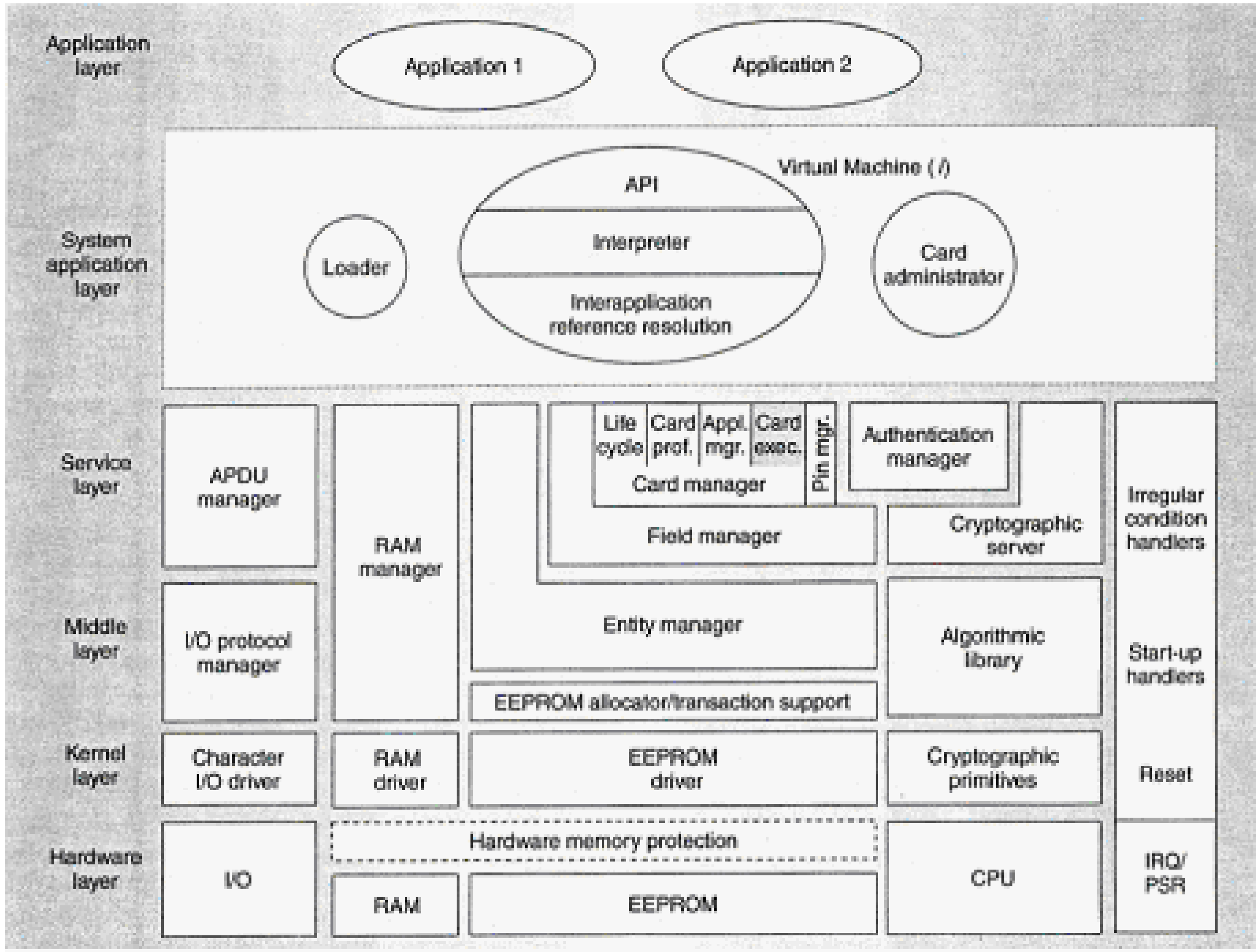
(*) Guillou,L.C, Ugon, M, Quisquater,J.J “Smartcard: a Standardized Security Device Dedicated to Public Cryptology”, 1992.

THE APPLICATION AGE

The Applications Age

- Invention of Java Card, in 1996
- A java virtual machine (JVM) runs applications written in *javacard*, a subset of the java language.
- Embedded applications are identified by a number, the Application Identifier (AID), whose maximum size is 16 bytes.
- This milestone marks the birth of Secure Elements (SE)
  - **A SE is a secure microcontrollers equipped with general purpose operating systems, supporting a virtual machine and able to download, delete, and execute applications.**
- It also enables the splitting up between hardware and software.
- These components may also be included in electronic chips with classical pinout, performing additional tasks such as NFC controller

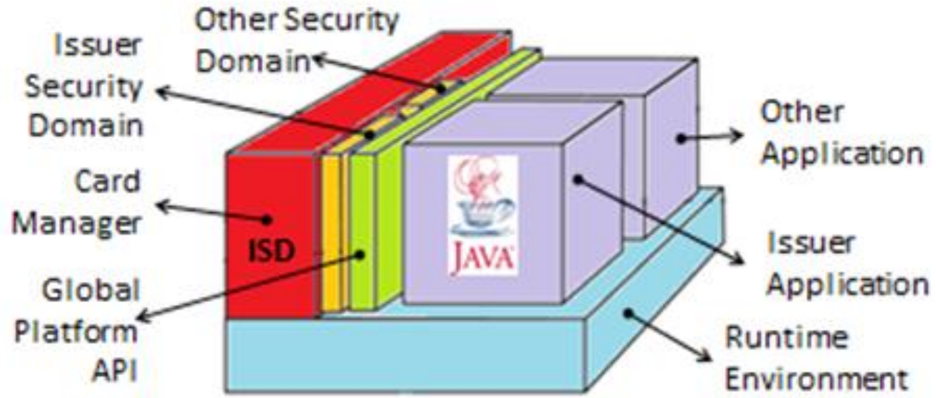
Example of a javacard OS



Tual, J.-P. "MASSC: a generic architecture for multiapplication smart cards", Micro, IEEE Volume: 19 , Issue: 5, 1999

The Applications Age

- The development of the javacard technology creates a need for trusted management of embedded software.
  - **The Global Platform specifications enforce the applications lifecycle i.e. downloading, activation and deletion in secure elements.**
  - All these operations are handled by a card manager entity, hosting an Issuer Security Domain (ISD) identified by an AID
- Applications
  - EMV cards
  - ICAO passports
  - USIM cards
  - PKCS Cards



Main Administration (GP) Commands

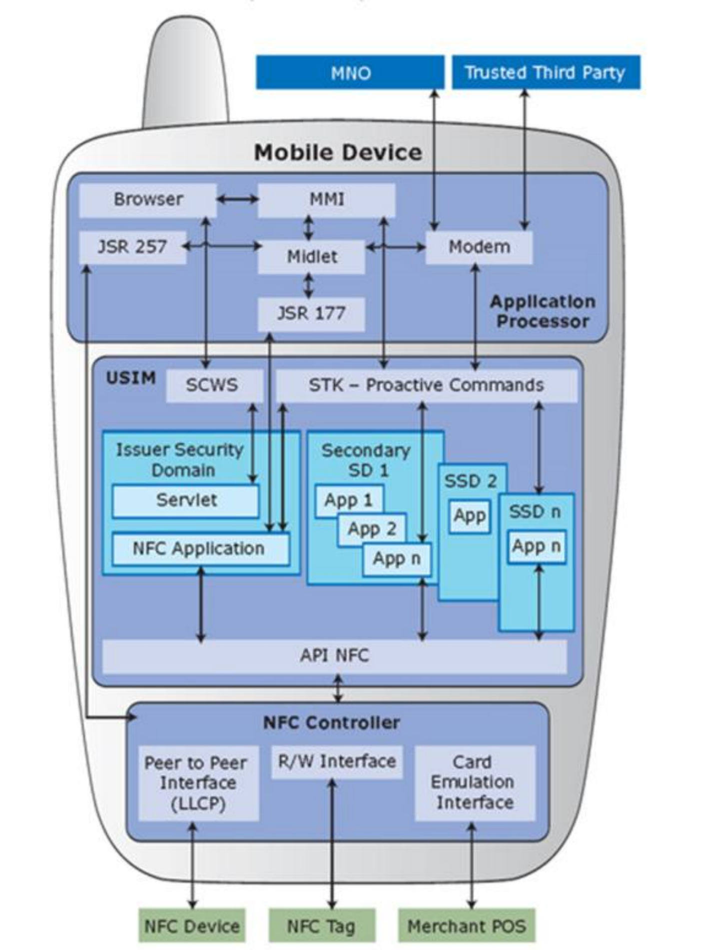
Commands	Topic
SELECT	Activate an embedded application. The Issuer Security Domain (ISD) is the application in charge of GP operations
INITIALIZE UPDATE	Initialize mutual authentication with ISD
EXTERNAL AUTHENTICATE	Ends mutual authentication with ISD
DELETE	Delete a package or an application
INSTALL	Allocate memory before package loading or instantiate an application
LOAD	Load a package

Javacard binaries are stored in CAP (Converted Applet) file, i.e. a set of .class files
Packages and applications are identified by AID

SCP01 Mutual Authentication

- Select ISD: A0 00 00 00 03 00 00 00 00
  - **>> 00 A4 04 00 08 A0 00 00 00 03 00 00 00 00**
  - **<< 6F 19**
  - **>> 84 08 A0 00 00 00 03 00 00 00 A5 0D 9F 6E 06 40 51 23 05 21 14 9F 65 01 FF 90 00**
- initialize-update
  - CLA=80 INS=50 P1=00 (key version), P2=00 P3=08, host challenge = 9D B1 90 58 6D 84 B6 96
    - **>> 80 50 00 00 08 9D B1 90 58 6D 84 B6 96**
    - **<< 00 00 23 25 00 47 30 90 18 09 FF 01 57 99 34 CB BC AE 75 9B 90 4C 79 38 1B 9A E2 79 90 00**
  - Key diversification data 10 bytes 00 00 23 25 00 47 30 90 18 09
  - Key information 2 bytes FF 01 FF=Key Version Number 01=Channel Protocol Identifier,
  - Card challenge 8 bytes 57 99 34 CB BC AE 75 9B
  - Card cryptogram 8 bytes 90 4C 79 38 1B 9A E2 79
- EXTERNAL AUTHENTICATE
  - P1 = Security level = 00 = No secure messaging expected, P2 = 0
  - Host cryptogram = 29 E5 5B 81 89 02 99 E0
  - Host MAC = E8 4A 14 89 66 54 7A 6C
    - **>> 84 82 00 00 10 29 E5 5B 81 89 02 99 E0 E8 4A 14 89 66 54 7A 6C**
    - **<< 90 00**
- Authentication Done

THE NFC AGE



MNO
Trusted Third Party
Mobile Device
Browser
MMI
JSR 257
Midlet
Modem
JSR 177
Application Processor
USIM
SCWS
STK – Proactive Commands
Issuer Security Domain
Servlet
NFC Application
Secondary SD 1
App 1
App 2
App n
SSD 2
App
SSD n
App n
API NFC
NFC Controller
Peer to Peer Interface (LLCP)
R/W Interface
Card Emulation Interface
NFC Device
NFC Tag
Merchant POS

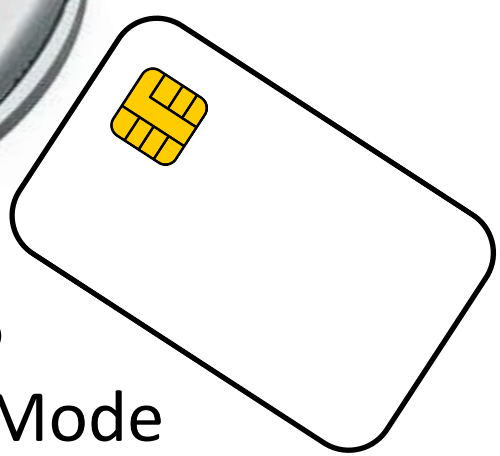
From ISO 7816 to ISO 14443

- The basic idea of Wi-Fi design was Wireless Ethernet.
- The basic idea of ISO 14443 design was Wireless (ISO 7816) Smartcard.
  - **Contrary to IEEE 802.11 there is no security features at the radio frame level.**

ISO 14443

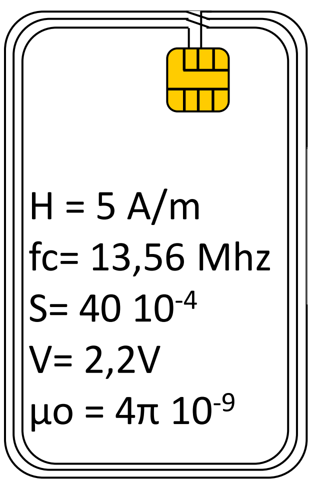
Contactless Mode

$$V = 2\pi\, f_c\, S\, \mu_o\, H$$



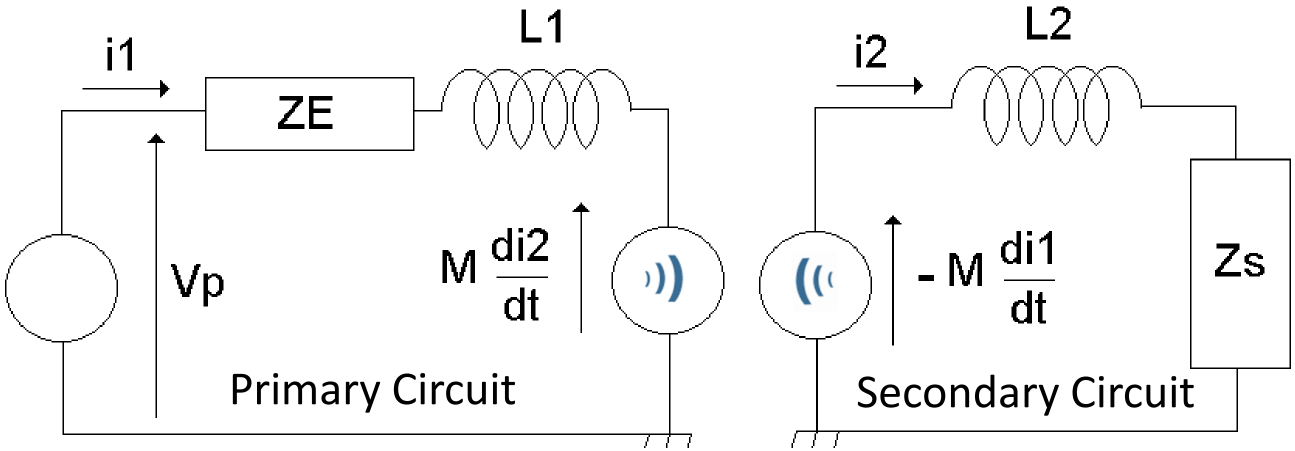
ISO 7816

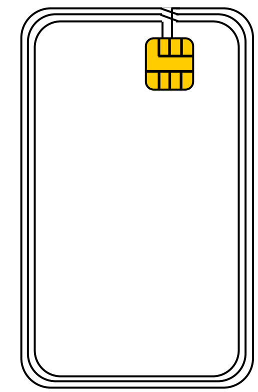
Contact Mode



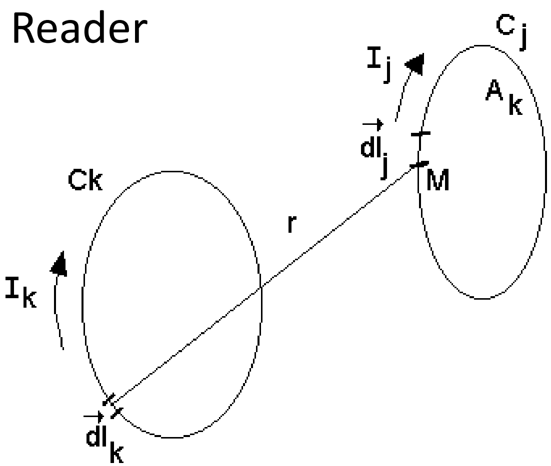
About Inductive Coupling

Reader





Card



Neumann formula

$$\boxed{\phi_{jk} = M_{jk}\, I_k = \frac{\mu_0 I_k}{4\pi} \oint_{C_j} \vec{dl}_j \oint_{Ck} \vec{dl}_k}$$

$$\boxed{M_{jk} = M_{kj}}$$

$$\begin{vmatrix} \Phi 1 \\ \Phi 2 \end{vmatrix} = \begin{vmatrix} L1 & M \\ M & L2 \end{vmatrix} \begin{vmatrix} i1 \\ i2 \end{vmatrix}$$

No energy propagation,
Is a vicinity proof.

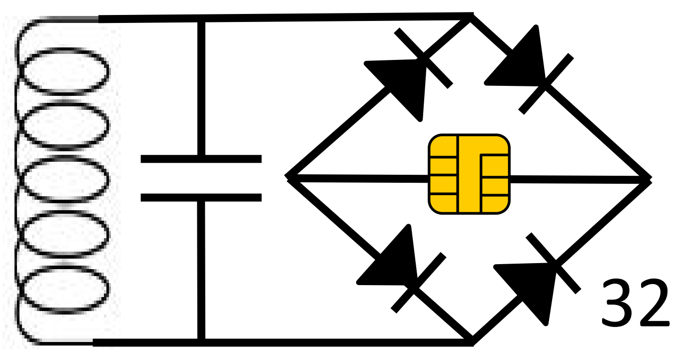
The Energy is conservative, i.e.
The Energy delivered by the primary circuit P_{PRI} = i1 . (M ω i2),
is equal to the energy consumed by the secondary circuit P_{SEC} = i2 . (M ω i1)

The NFC Age Milestones

- 1994, Mifare 1K
  - In 2011 Mifare chips represent 70% of the transport market.
- 2001, ISO 14443 Standards (13,56 Mhz)
  - Type A (Mifare)
  - Type B
  - Type F (Felica)
- 2004, NFC Forum
  - Mifare (NXP), ISO14443A, ISO14443B, Felica (Sony)
  - Three functional modes :
    - Reader/Writer, Card Emulation, Peer to Peer
- NFC controllers realizes NFC modes

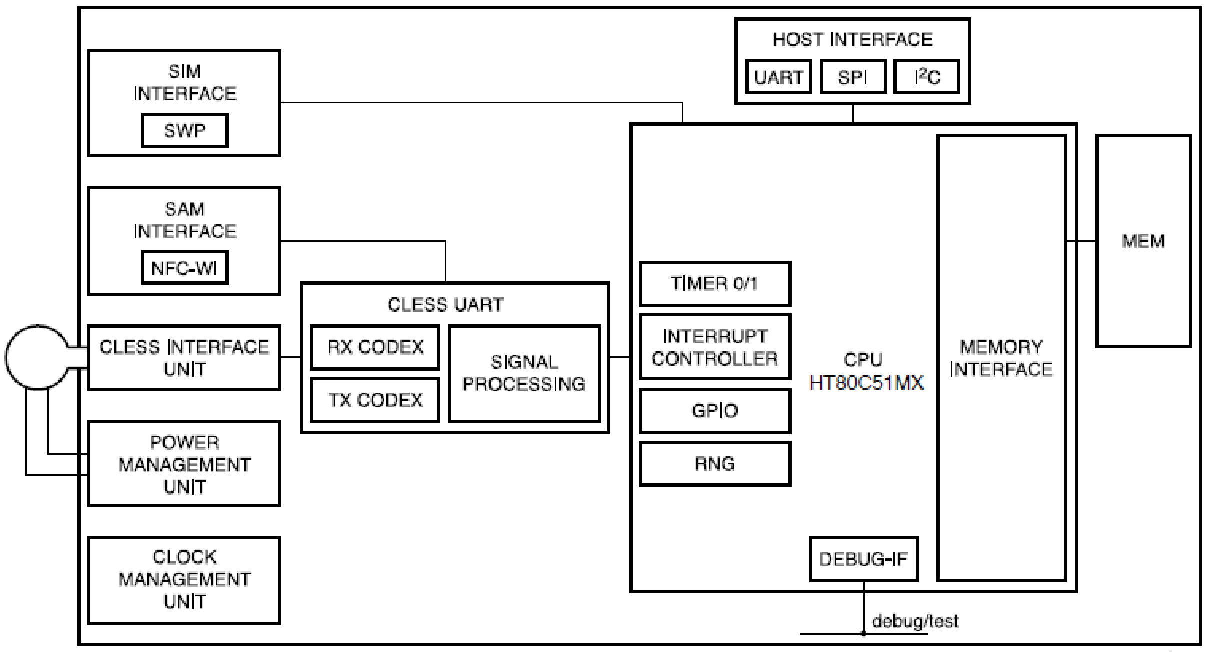
The Near Field Communication (NFC) Age

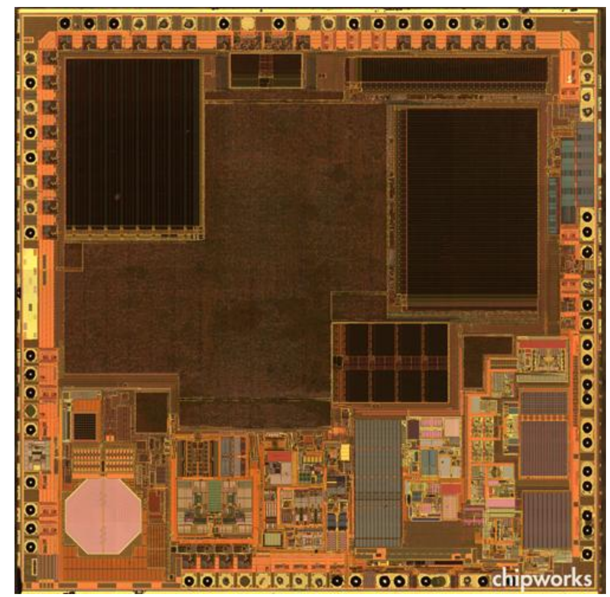
- The early Near Field Communication technology was normalized by the ISO 14443 standards at the beginning of the 21st century
- It was targeting contactless applications, mainly for ticketing services.
  - It is possible to draw a parallel with the Wi-Fi technology that delivers seamless IP services; **however and contrary to Wi-Fi framework, ISO 14443 doesn't natively support security facilities (i.e. data privacy and integrity)** for radio packets,.
- ISO 14443 works at the 13,56 MHz frequency, the secure microcontroller is fed by an inductive process.
  - For a magnetic field of 5A/m and a loop area of 8x5 cm^2 the resulting voltage is about 2,2V (per loop).
  - Several data coding schemes are defined (referred as A, B, and F), which support a throughput ranging from 104 to 848 Kbits/s.
- NFC standards build an extended framework over ISO 14443. They defined three working modes:
  - **reader/writer,** it is a Card Acceptance Device (CAD=reader), compatible with ISO 14443 standards, feeding a contactless device, and supporting **typeA, typeB and typeF** radio coding schemes.
  - **card emulation**, it is a set of protocols working with reader/writer appliances, which are realized by secure elements or by pure software means.
  - **Peer to Peer (P2P**), two devices, the "Initiator" and the "Target" establish a NFC session using protocols (incompatible with ISO 14443) defined by the NFCIP-1 standard



NFC and Secure Elements

- Some NFC Controllers embed a Secure Element
- In that case the card emulation mode may be managed by the embedded secure element
- This is the **Google** and **Apple** Secure Element Model



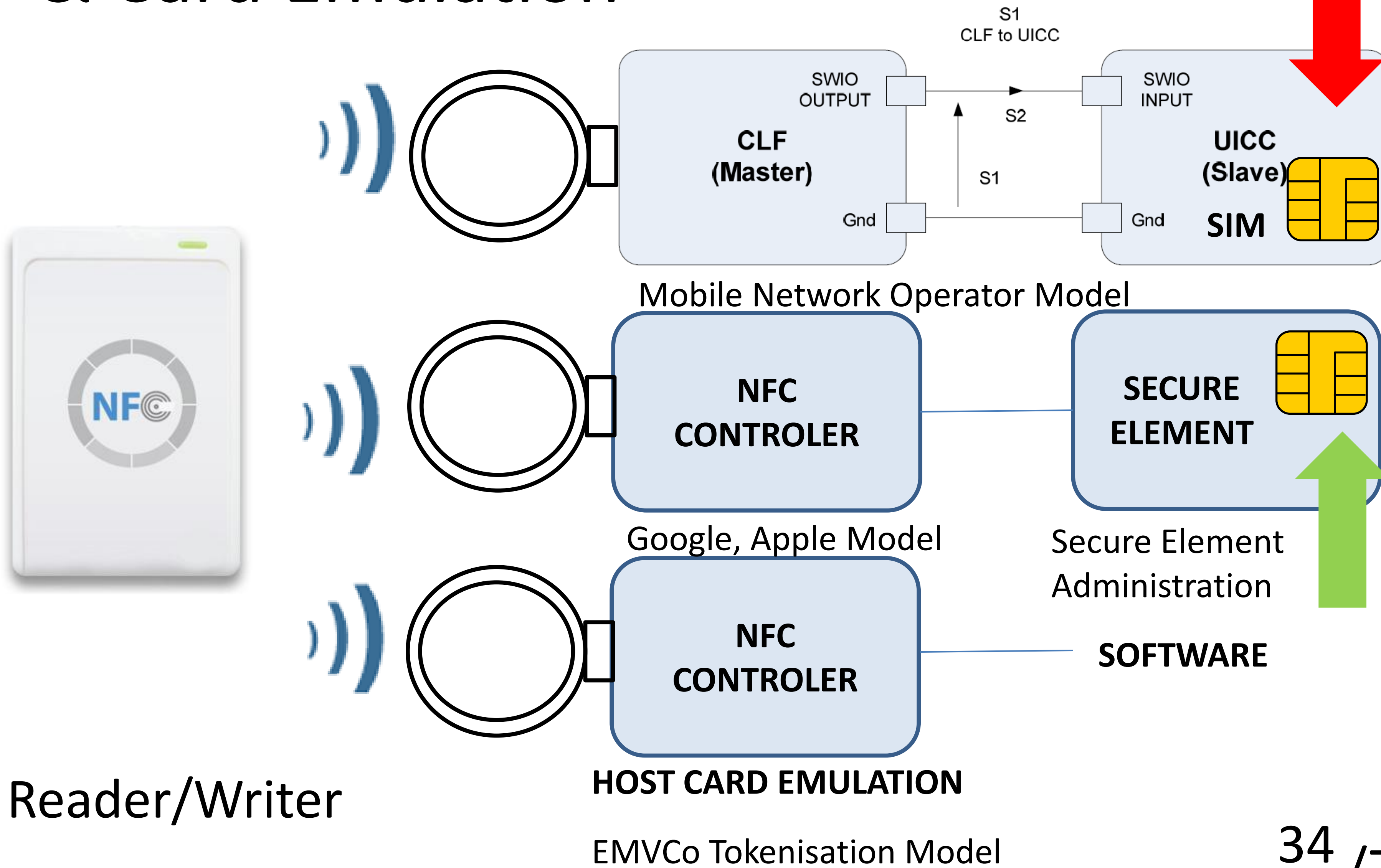


www.chipworks.com

Reader/writer ISO 14443 –A-B, MIFARE, FeliCa®, NFC Forum tags, ISO 15693
Card Emulation ISO 14443 –A-B-B', MIFARE, FeliCa RF , SWP
RAM 5Ko, ROM 128 Ko, EEPROM 52 Ko

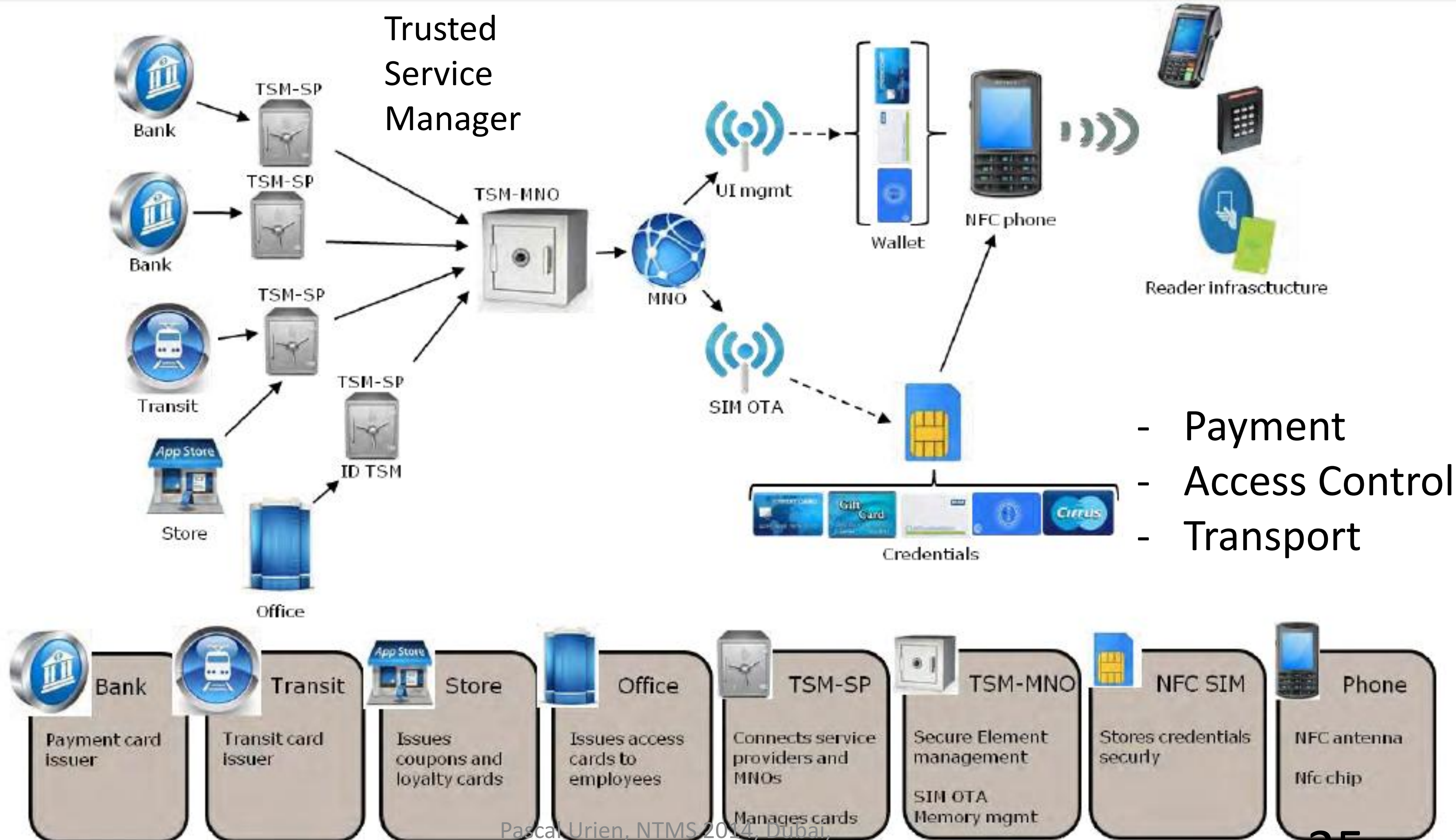
NFC Reader/Writer & Card Emulation

OTA(Over The Air)
GP Administration
SMS, GSM 3.48



HID NFC White Paper: SIM centric Services

Trusted
Service
Manager



- Payment
- Access Control
- Transport

NFC ecosystem with the Secure Element in the SIM and one MNO

NFC standards

<table>
<tr><th>Activity</th><th colspan="7">Technology / Device Platform</th></tr>
<tr><td>Listen, RF Collision Avoidance, Technology Detection, Collision Resolution</td><td colspan="4">NFC-A
ISO 14443-2A
ISO 14443-3A</td><td>NFC-B
14443 -2B
14443 -3B</td><td colspan="2">NFC-F
ISO 14443-2A
ISO 14443-3A
FELICA</td></tr>
<tr><td>Device Activation</td><td rowspan="3">NFC-DEP Protocol
NFCIP-1</td><td>Type 1 Tag Platform</td><td>Type 2 Tag Platform</td><td>Type 4A Tag Platform</td><td>Type 4B Tag Platform</td><td>Type 3 Tag Platform</td><td rowspan="3">NFC-DEP Protocol
NFCIP-1</td></tr>
<tr><td>Data Exchange</td><td colspan="2" rowspan="2">Type 1, 2, and 3 Tag Half-duplex Protocol</td><td colspan="2" rowspan="2">ISO-DEP Protocol
ISO 14443-4</td><td rowspan="2">Type 1, 2, and 3 Tag Half-duplex Protocols</td></tr>
<tr><td>Device Deactivation</td></tr>
</table>

Reader/Writer – Card Emulation

NDEF

SNEP

LLCP

NFC-SEC

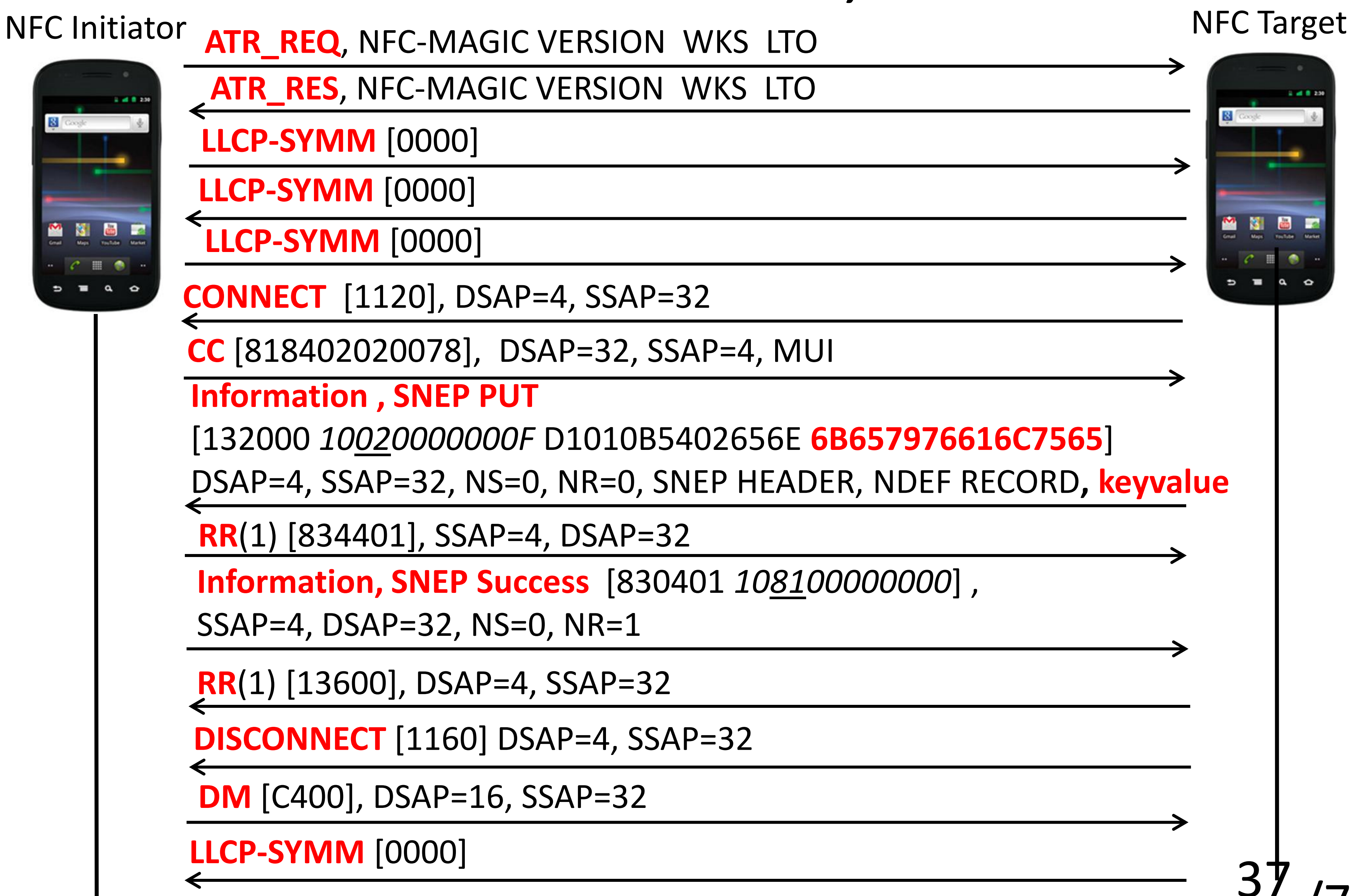
DEP

Passive Mode
Active Mode
NFCIP-1

P2P

*ISO/IEC_18092 standard and NFCIP-1 standards are similar
* DEP: Data Exchange Protocol

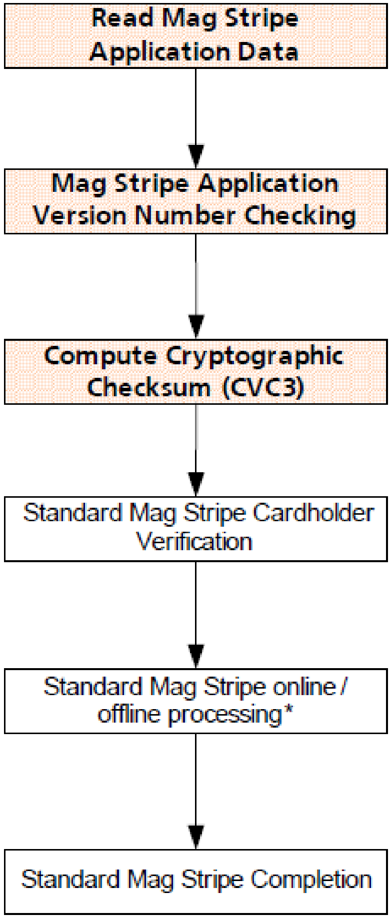
About NFC P2P SNEP, Android 4.x



PAYMENTS

About NFC Payments

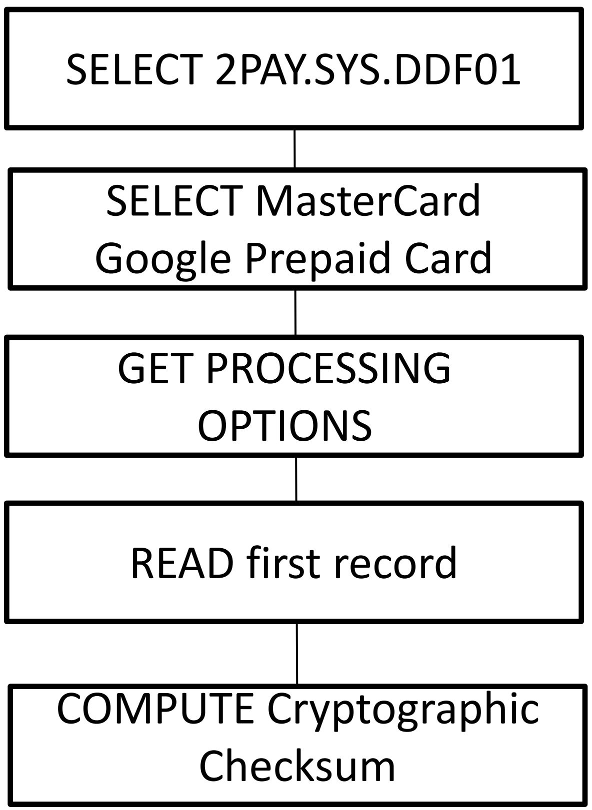
- Some NFC payments are based on the MasterCard PayPass specification
- There is two modes
  - Mag Stripe, a four digits CVC3 (*Card Verification Value*) is computed from a 3xDES and various parameters (PAN, ATC counter,…)
  - Contactless EMV
- The Secure Element securely performs calculations or runs the EMV application
- Contactless payments introduce a new paradigm, the virtualization of the bank card.
- The merchant terminal doesn't known where is running the payment application on the mobile side.

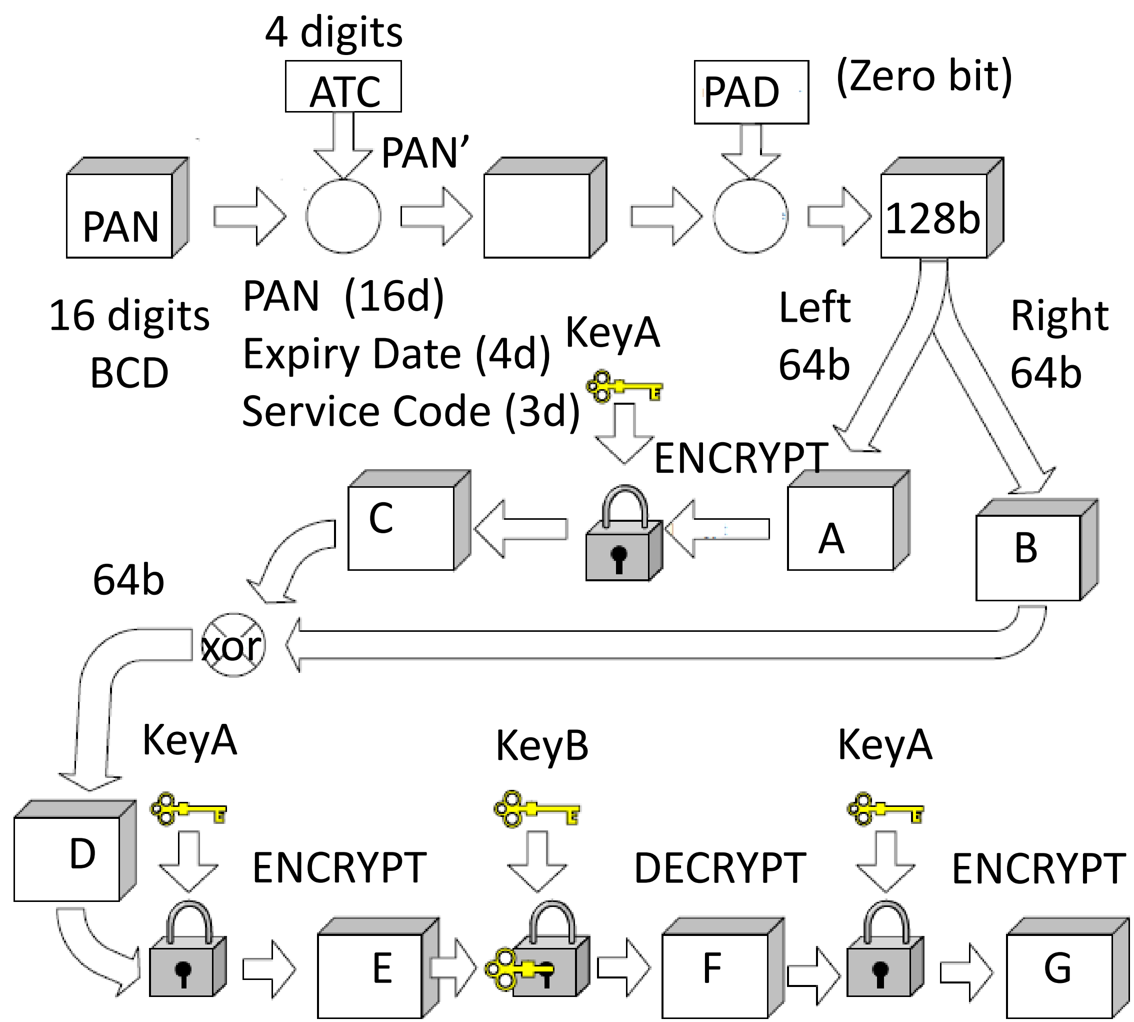


Where is running the SE application ?

* MasterCard® *PayPass™, M/Chip, Acquirer Implementation Requirements*, v.1-A4 6/06

Some Details with EMV Mag. Stripe*





G=A23FB35FC89AE3A9
23358939AFBFCAEA
2335893905152040

CVC3=233

*Visa Contactless Payment Specification Version 2.0.2 July 2006

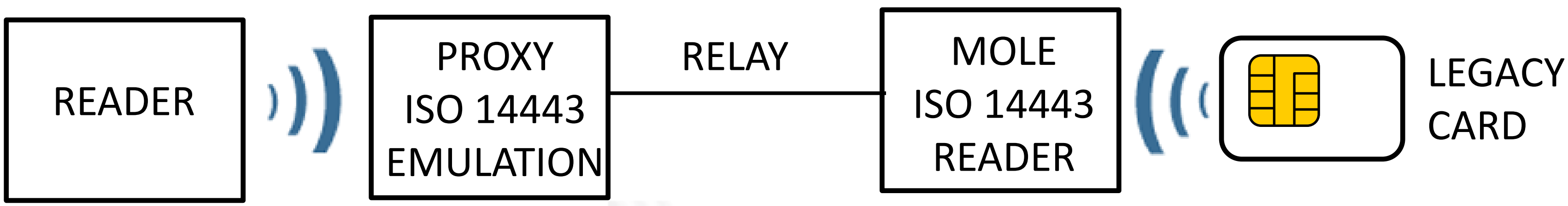
Details of ISO7816 commands

```
// SELECT 2PAY.SYS.DDF01
>> 00A404000E325041592E5359532E4444463031
<< 6F2C840E325041592E5359532E4444463031A51ABF0C1761154F10A000000004
1010AA54303200FF01FFFF8701019000
// Select MasterCard Google Prepaid Card
>> 00A4040010A0000000041010AA54303200FF01FFFF
<< 6F208410A0000000041010AA54303200FF01FFFFA50C500A4D61737465724361
72649000
// Get Processing Options
>> 80A80000028300
<< 770A820200009404080101009000
// Reader First Record
>> 00B2010C00
<< 706A9F6C0200019F62060000000000389F63060000000003C656294235343330
39393939303939373939395E202F5E3137313131303130303130303030303030
303030309F6401049F650200389F660203C69F6B135430999909979999D17111
010010000000000F9F6701049000
// COMPUTE Cryptographic Checksum (CVC3)
>> 802A8E800400000080
<< 770F9F610200389F600200389F360200129000
```

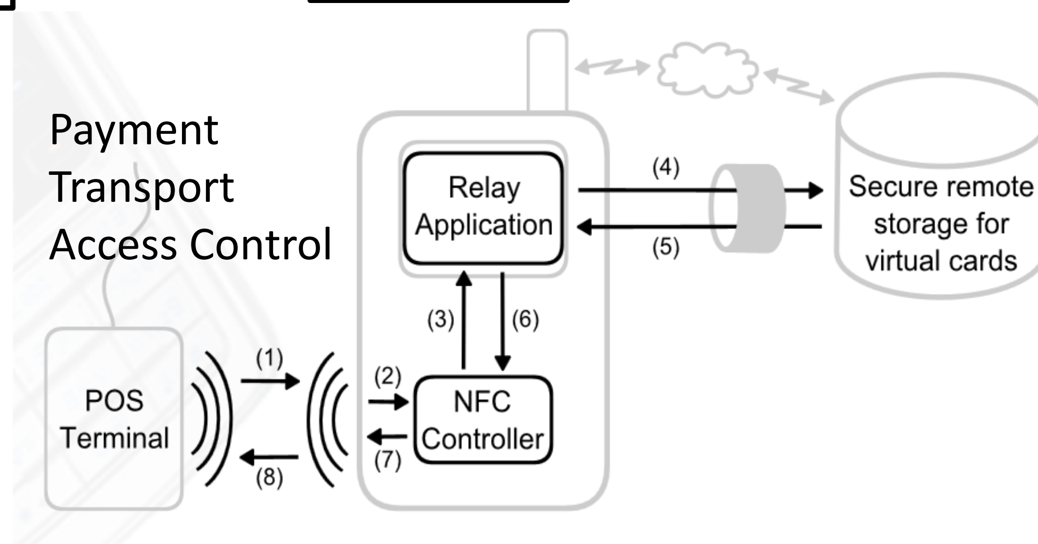
ISO 7816 commands are processed by the Secure Element (SE) OR by the Host Card Emulation (HCE)

Relays Attacks : Tricks or Tips

- In 2005, G.Hancke introduced the concept of the “Relay Attack”. The goal was a remote use of legacy Mifare card

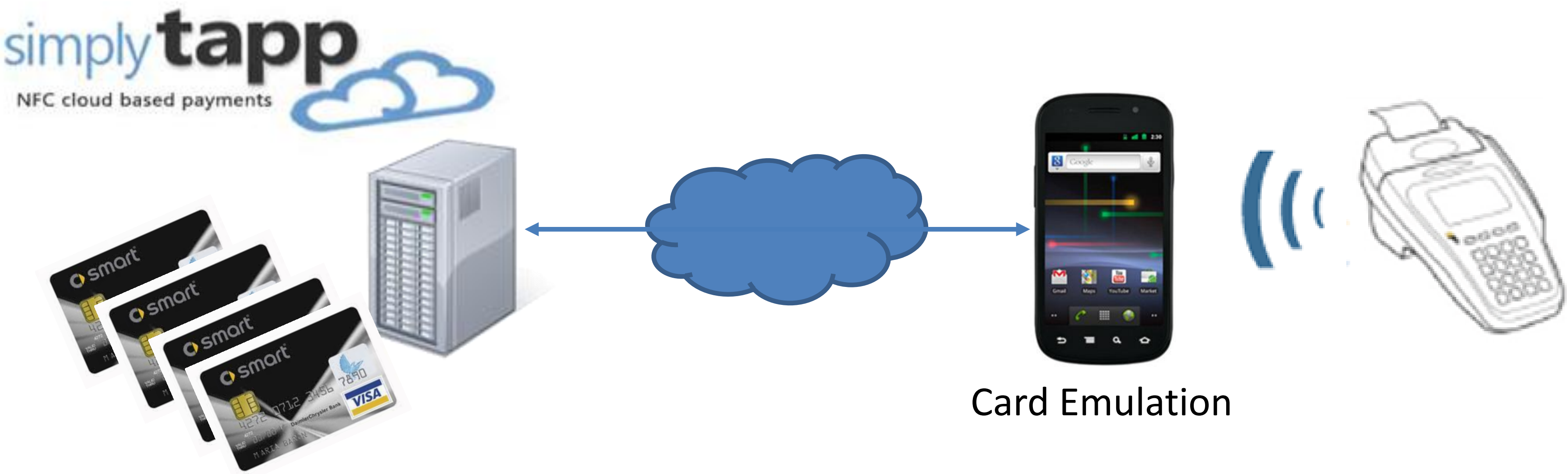


- The software card emulation in smartphone enables a new generation of relay attack.



Roland, M., "Software Card emulation in NFC-enabled Mobile Phones: Great Advantage or Security Nightmare ?", in proceedings of WSSI/SPMU, June 2012.

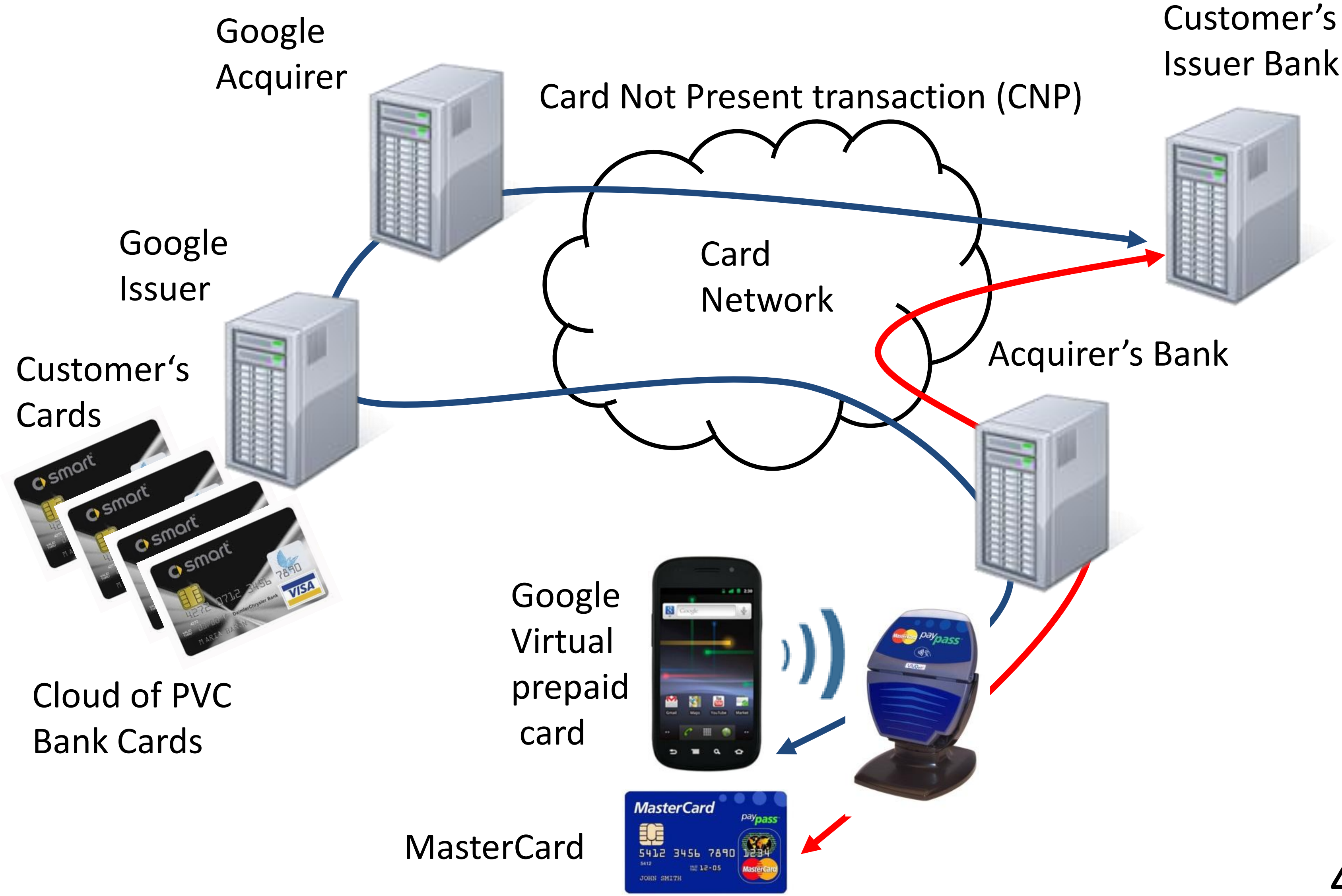
SIMPLY TAPP (2012)



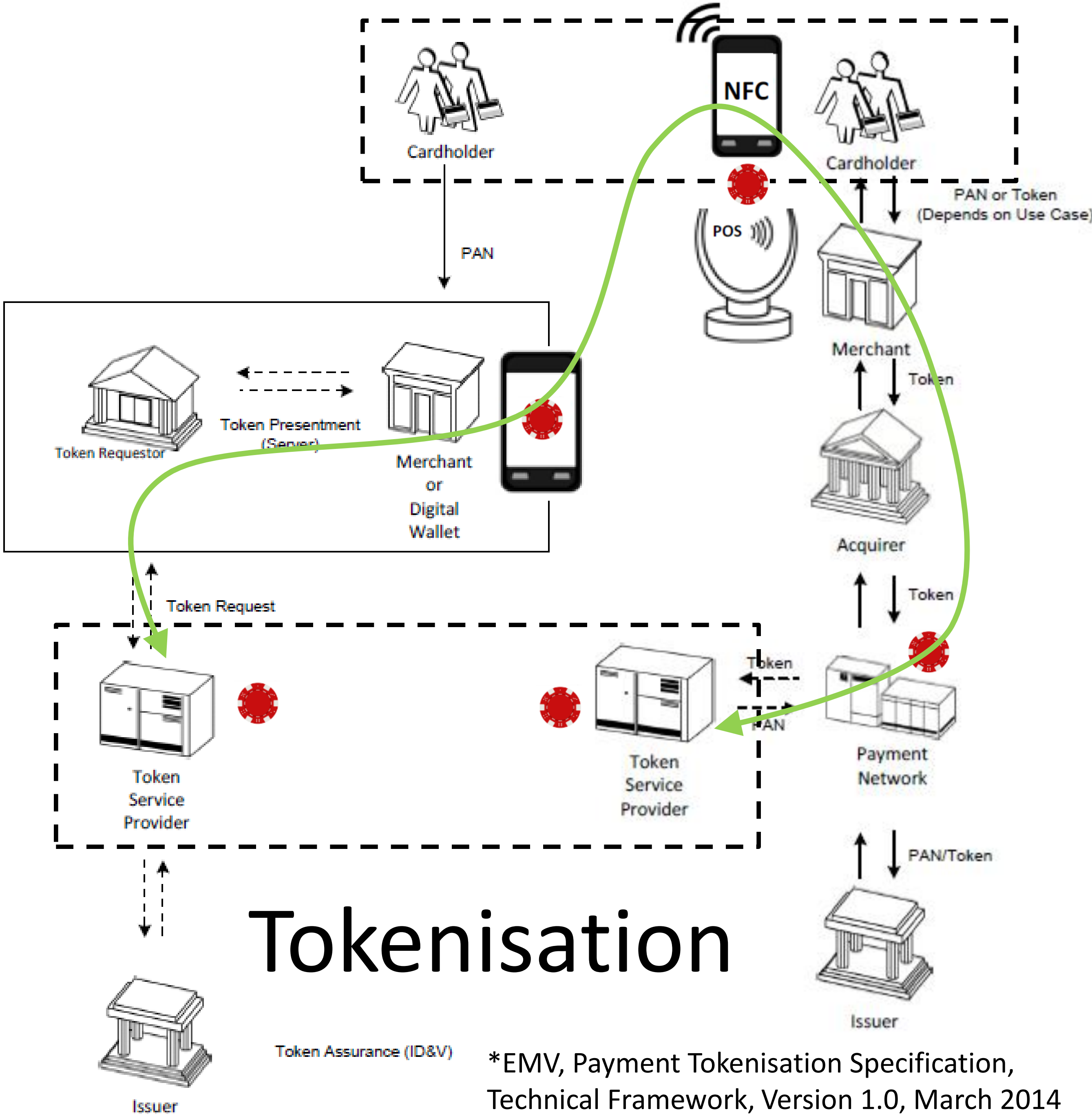
The CyanogenMod (9.0) Android open distribution supports the software card emulation

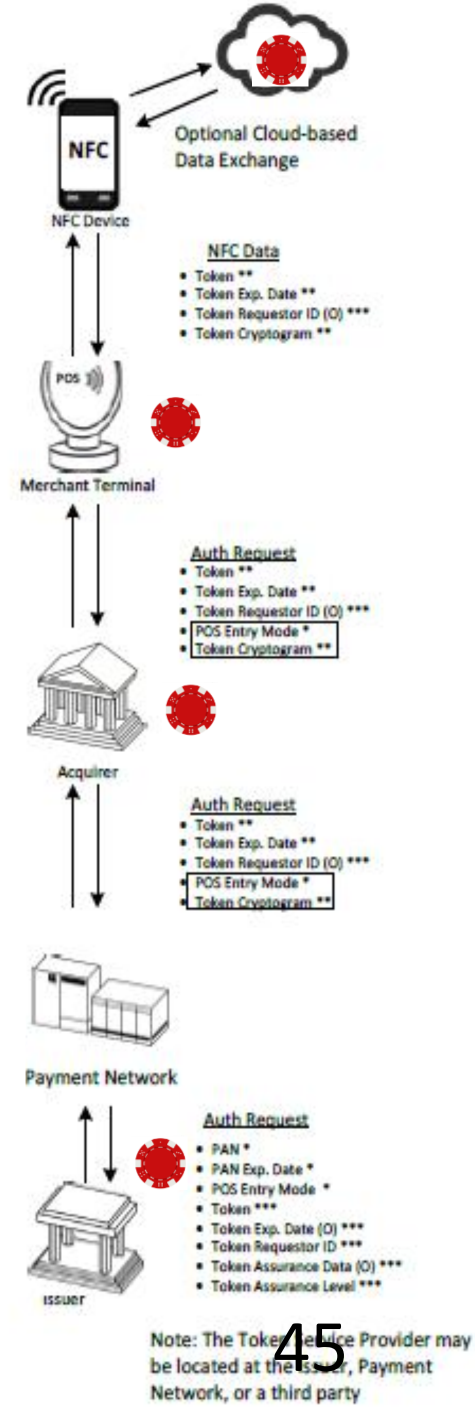
cyanogenmod

The Google Wallet 2 (2012)



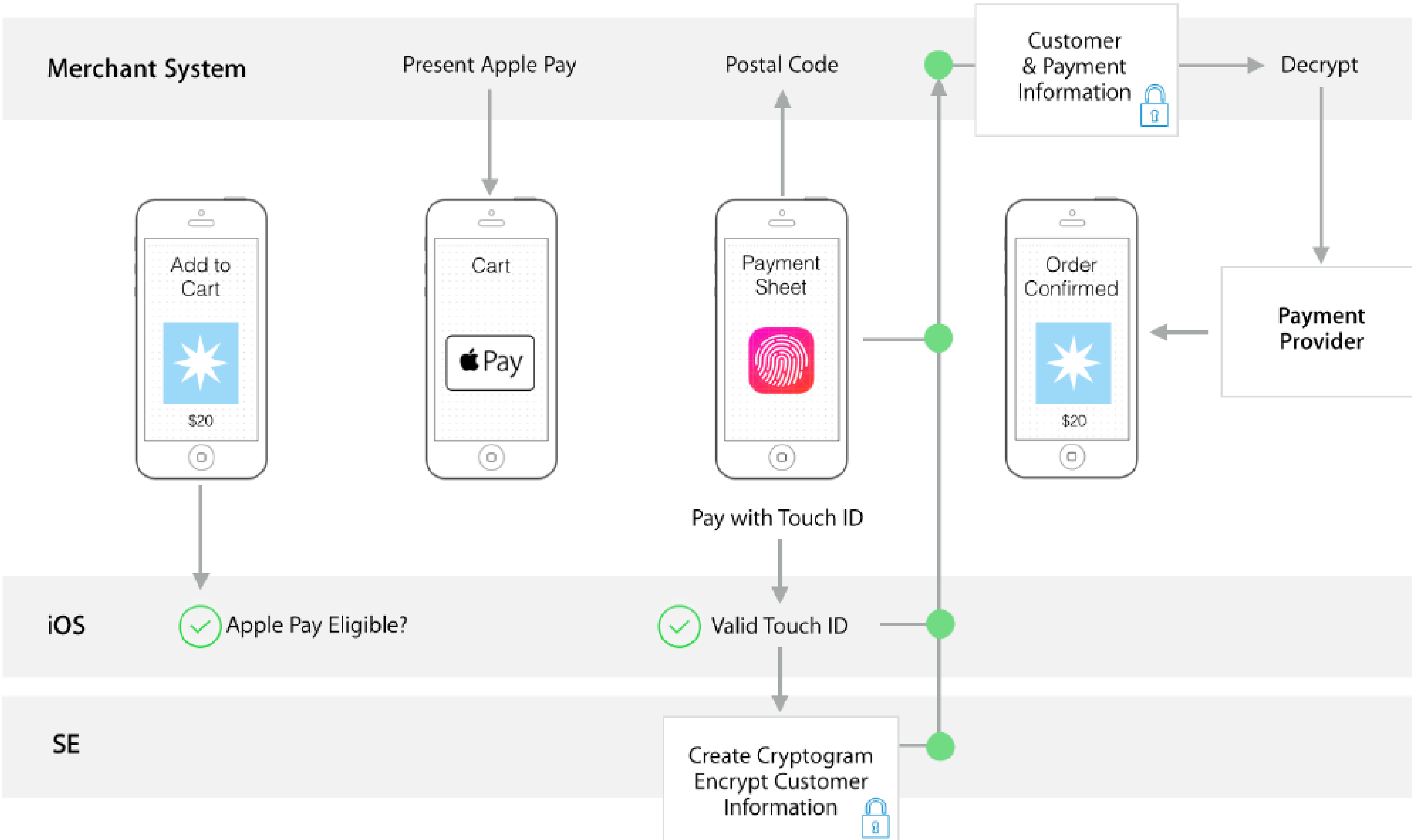
Tokenisation





*EMV, Payment Tokenisation Specification, Technical Framework, Version 1.0, March 2014

ApplePay: A Token Requestor

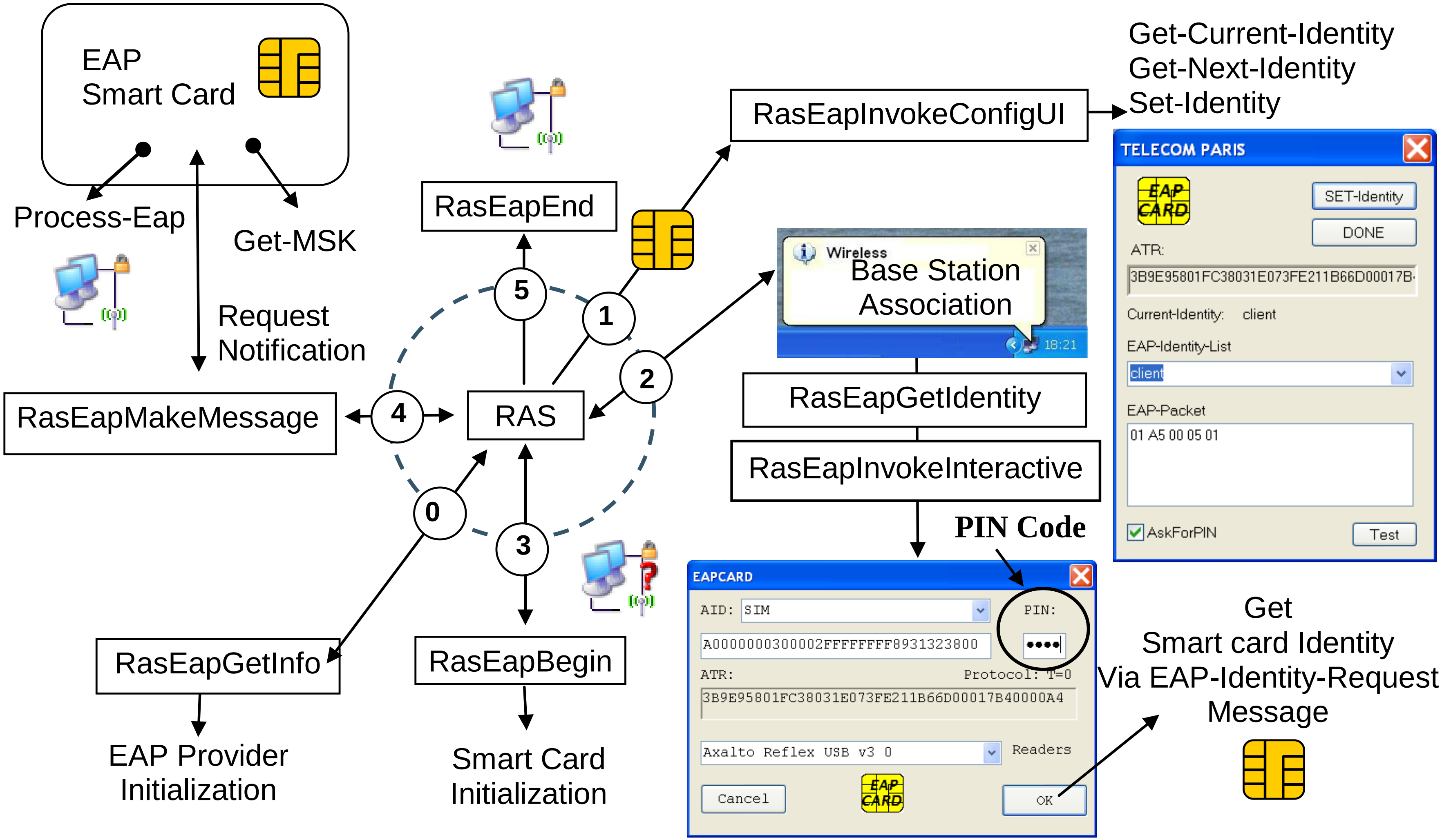


Getting Started with Apple Pay, Version 1.0, 2014

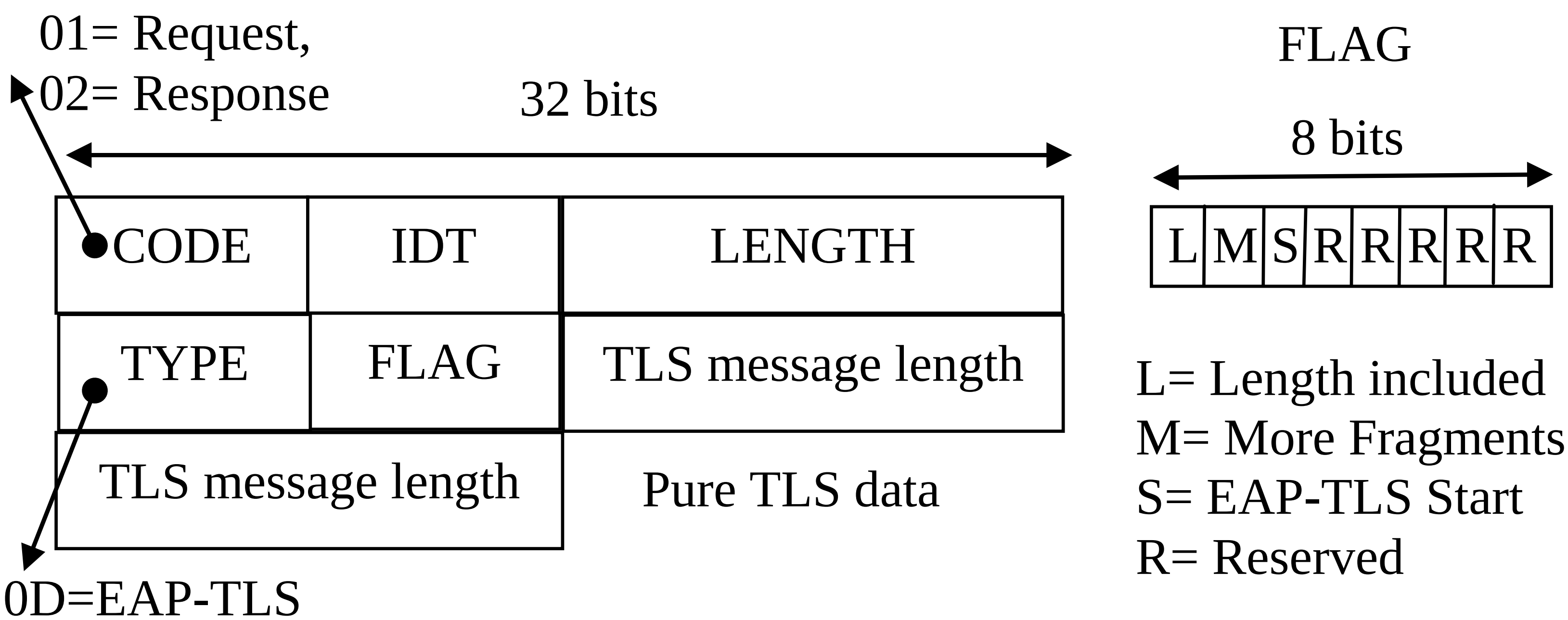
TLS Secure Elements

From EAP-TLS to TLS

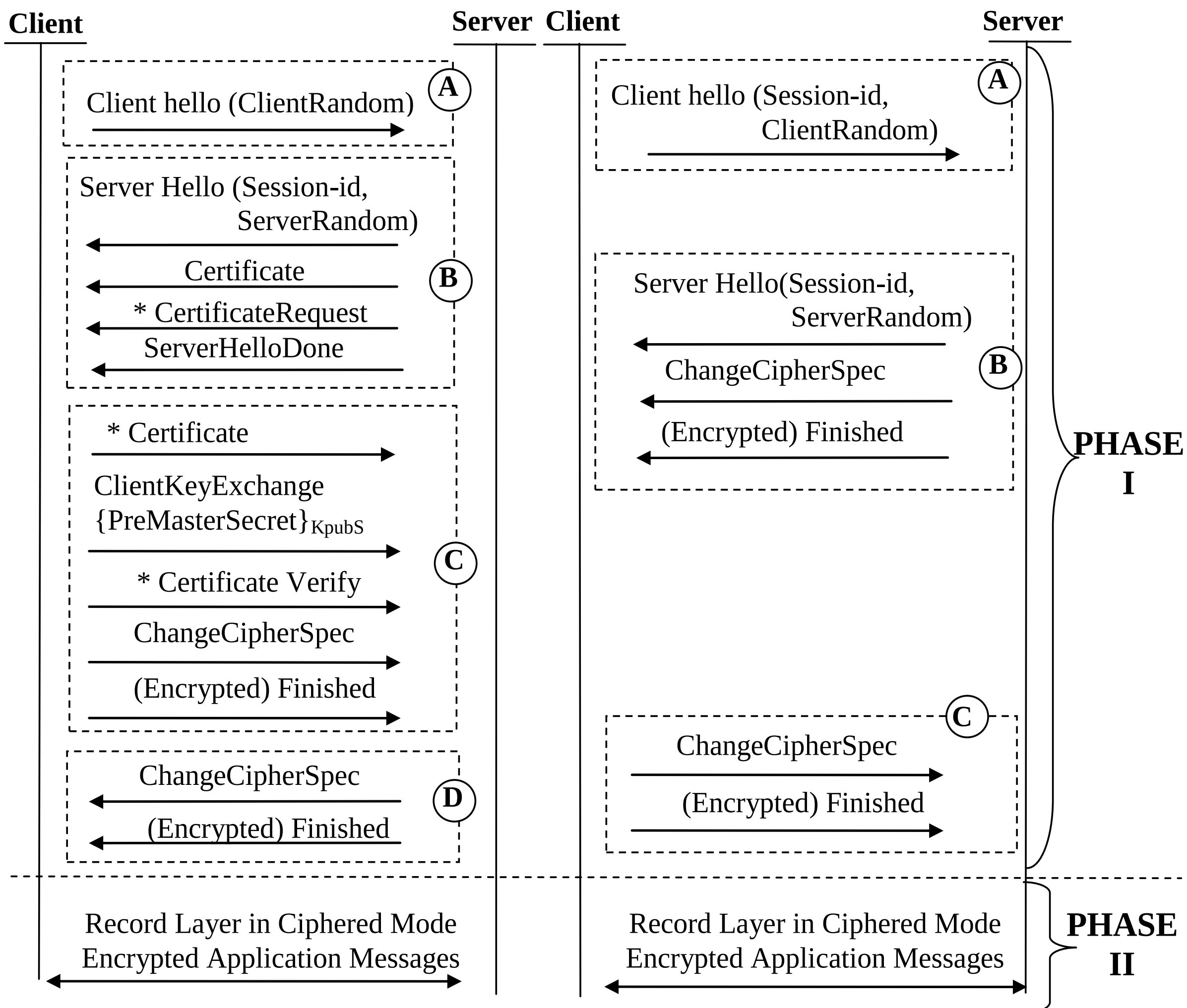
EAP Smartcard



EAP-TLS


01= Request,
02= Response
32 bits
CODE
IDT
LENGTH
TYPE
FLAG
TLS message length
TLS message length
Pure TLS data
0D=EAP-TLS
FLAG
8 bits
L M S R R R R R
L= Length included
M= More Fragments
S= EAP-TLS Start
R= Reserved

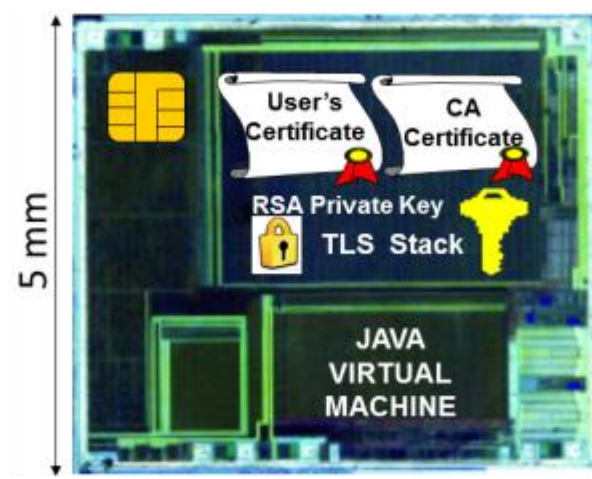
TLS Full & Resume mode



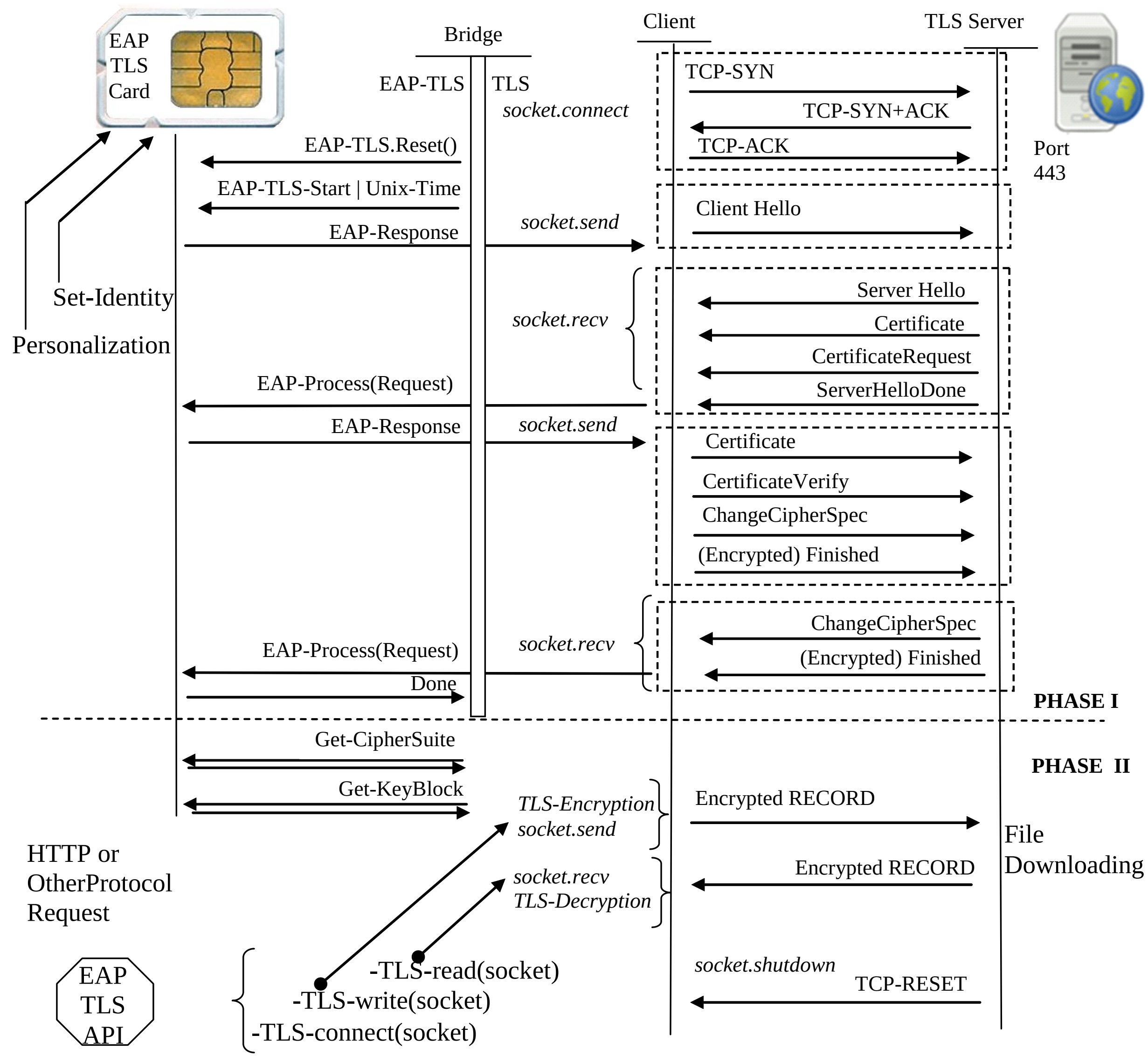
** Optional messages*

The (EAP) TLS smartcard

- All TLS packets are processed in a secure element
  - until the reception of the server finished message in the case of a TLS full session,
  - or the transmission of the client finished message in the case of a TLS abbreviated session.
- At this step the keys-block, a set of four keys used for TLS messages encryption and integrity, and the cipher-suite parameter that identifies the cryptographic algorithms used for these purpose, are transferred to the applicatione
- A dedicated API (EAP-TLS API) comprises three main procedures
  - ctx= TLS-Connect(socket).
  - buffer= TLS-Read(socket, ctx).
  - TLS-Write(socket, ctx, buffer).



EAP-TLS Card Choreography



Some (Best) Figures

Device	MD5/ bloc	SHA1/ bloc	RSA PUB	RSA PRIV	x3DES/ bloc	AES/ bloc	IO
USIM	2,6 ms	3,7 ms	150 ms	290 ms	5,2 ms	-	0,24 ms/byte
JavaCard	0,49 ms	0,93 ms	25 ms	560 ms	2,1 ms	2,6 ms	0,17 ms/byte

- Tfull = 230x MD5 + 230x SHA1 + 2x RSA-Pub + 1x RSA-Priv + IO[2500 bytes] + Tother
- Tresume = 75x MD5 + 75x SHA1 + IO[250 bytes] + Tother

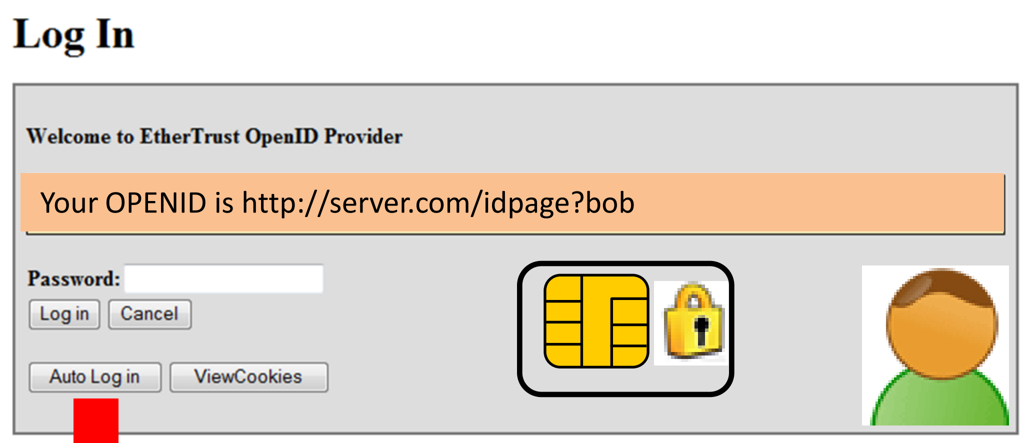
- For the javacard
  - Tfull = 2,0 s; Tother = 0,7s
  - Tresume= 0,5s; Tother= 0,35s
- For the USIM
  - Tfull = 4,3s; Tother= 2,65s
  - Tresume=1,65s; Tother= 1,45s

TLS Secure Elements

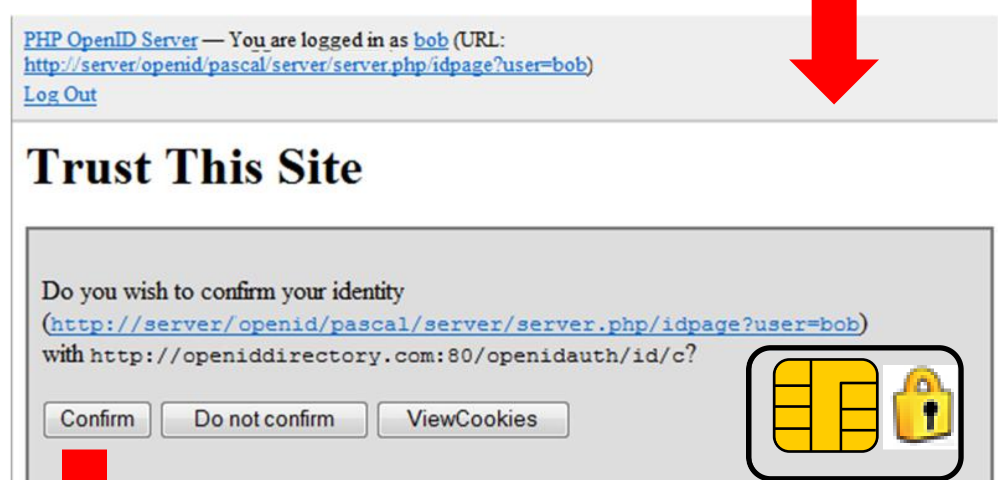
Client and Server Applications

TRUSTED OPENID

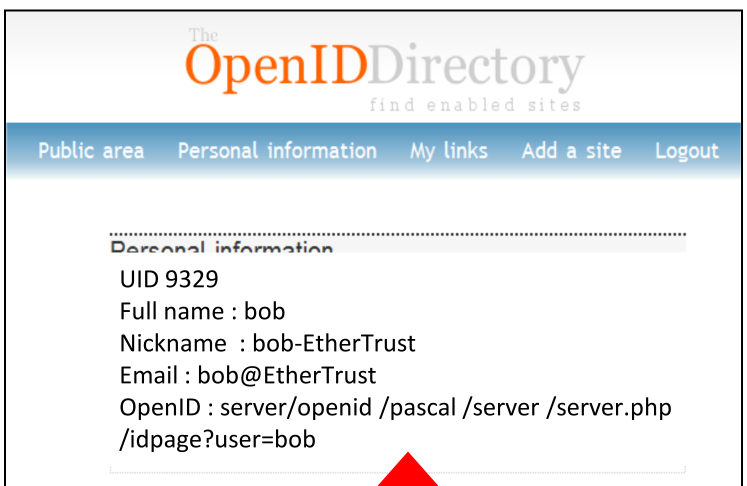
http://server.com/idpage?bob

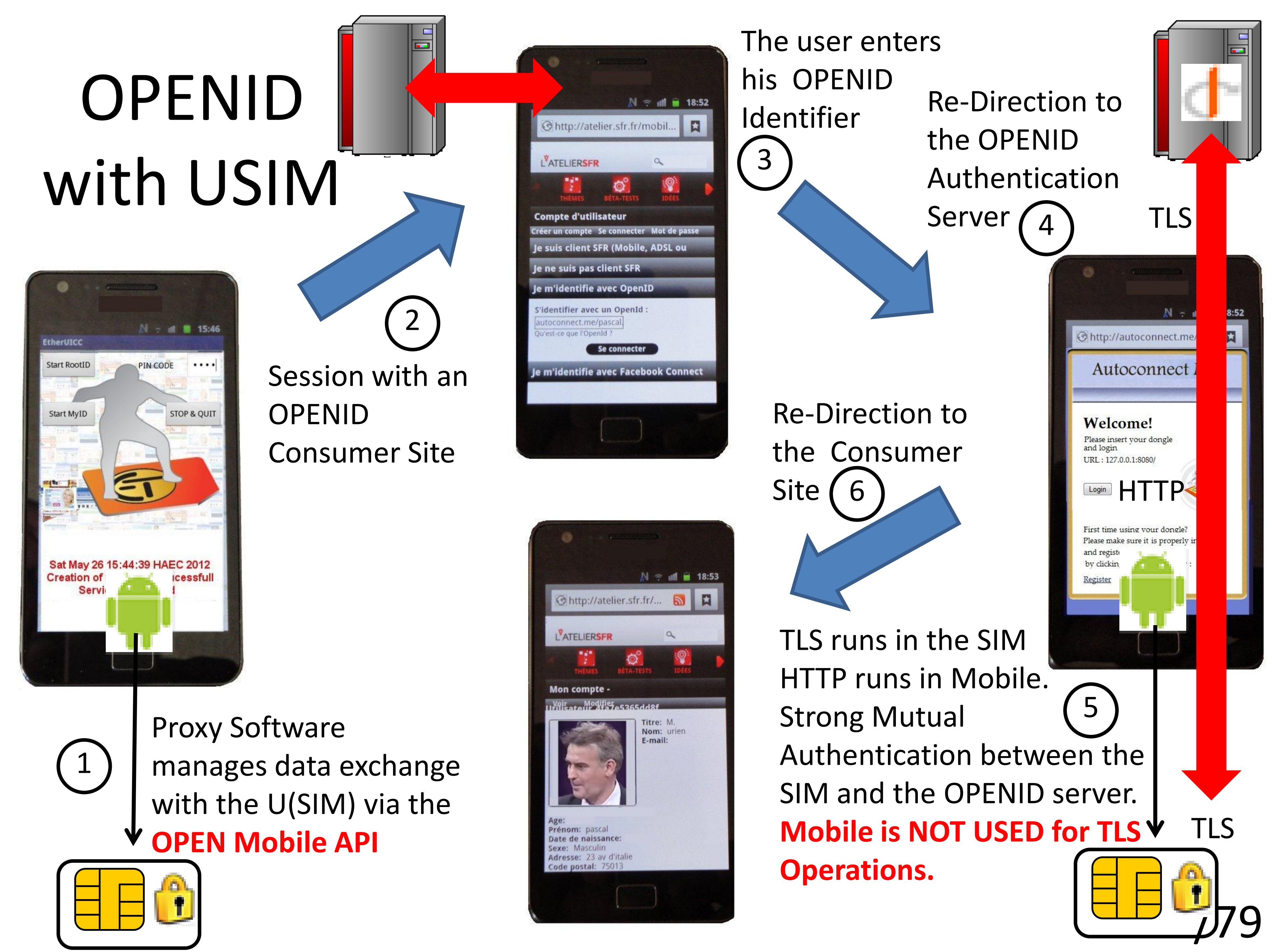


http://127.0.0.:8080/~https=server.com/login?sid



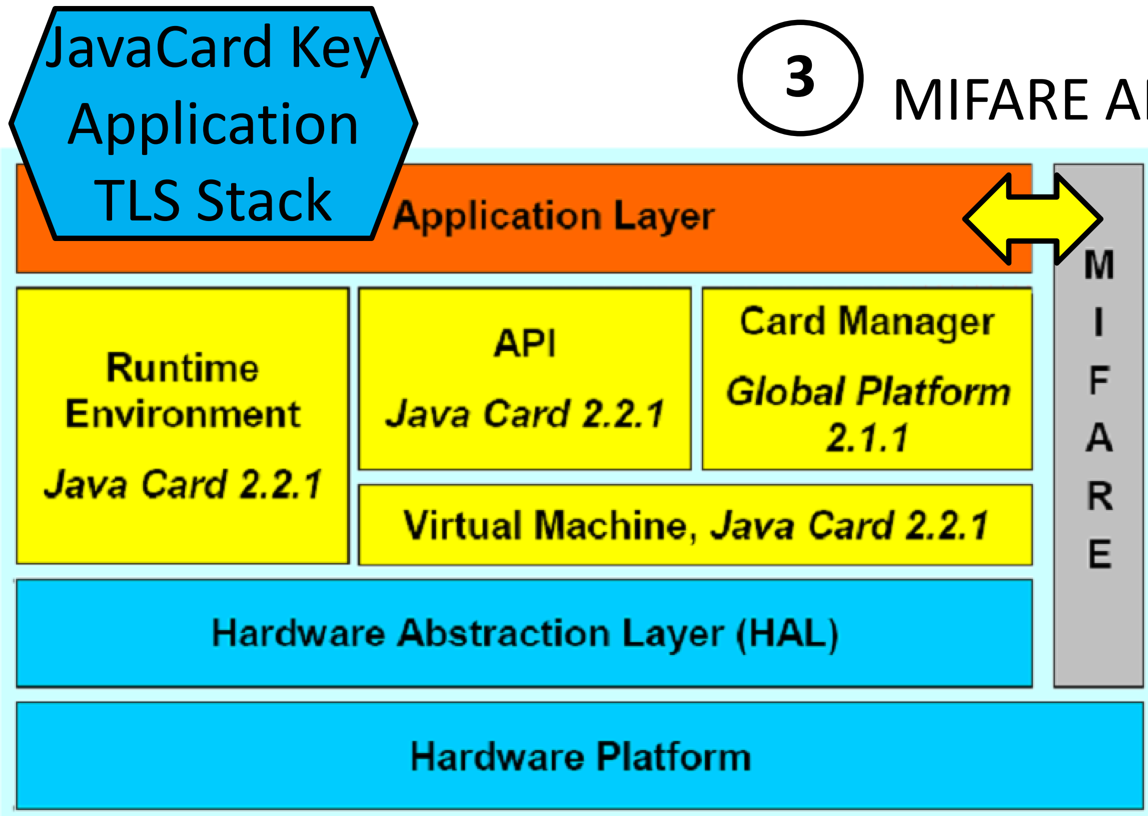
http://127.0.0.:8080/~https=server.com/trust

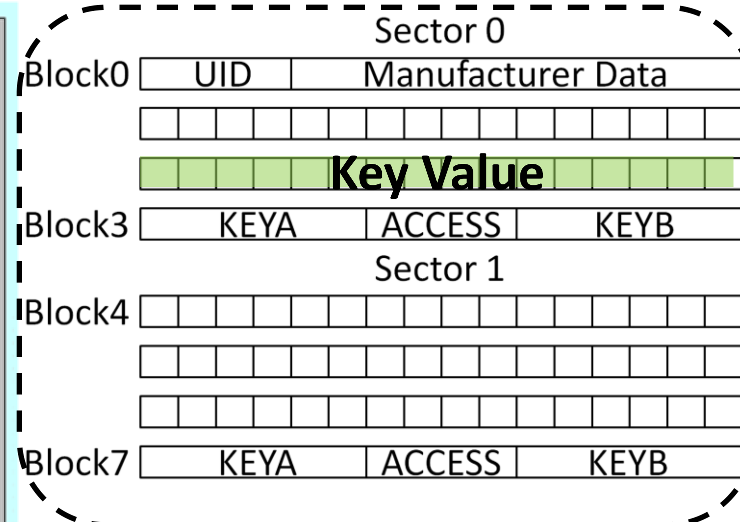



OPENID
with USIM
Session with an
OPENID
Consumer Site
2
The user enters
his OPENID
Identifier
3
Re-Direction to
the OPENID
Authentication
Server
4
TLS
HTTP
Re-Direction to
the Consumer
Site
6
Proxy Software
manages data exchange
with the U(SIM) via the
OPEN Mobile API
1
TLS runs in the SIM
HTTP runs in Mobile.
Strong Mutual
Authentication between the
SIM and the OPENID server.
5
Mobile is NOT USED for TLS
Operations.
TLS

Secure RFID (Dual Interface)

3 MIFARE API





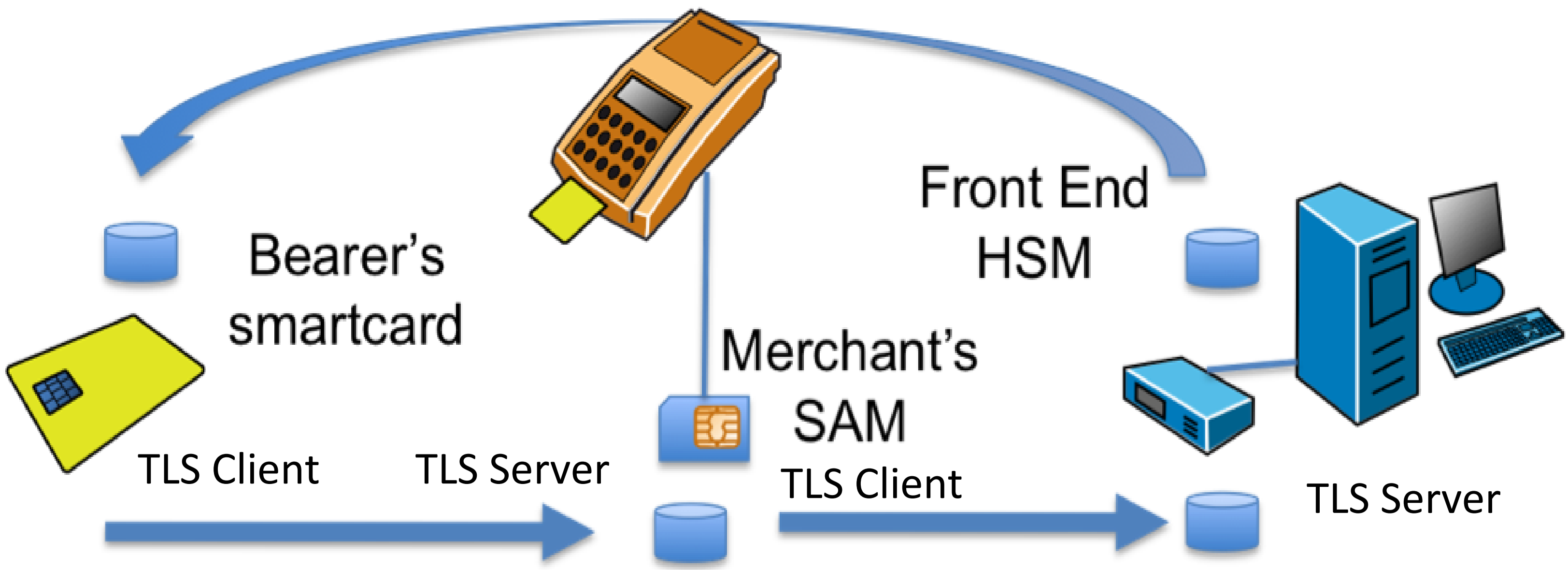
2 Mifare Emulation

1 Secure Microcontroller with NFC resources equipped with a Java Virtual Machine

NFC Mobile

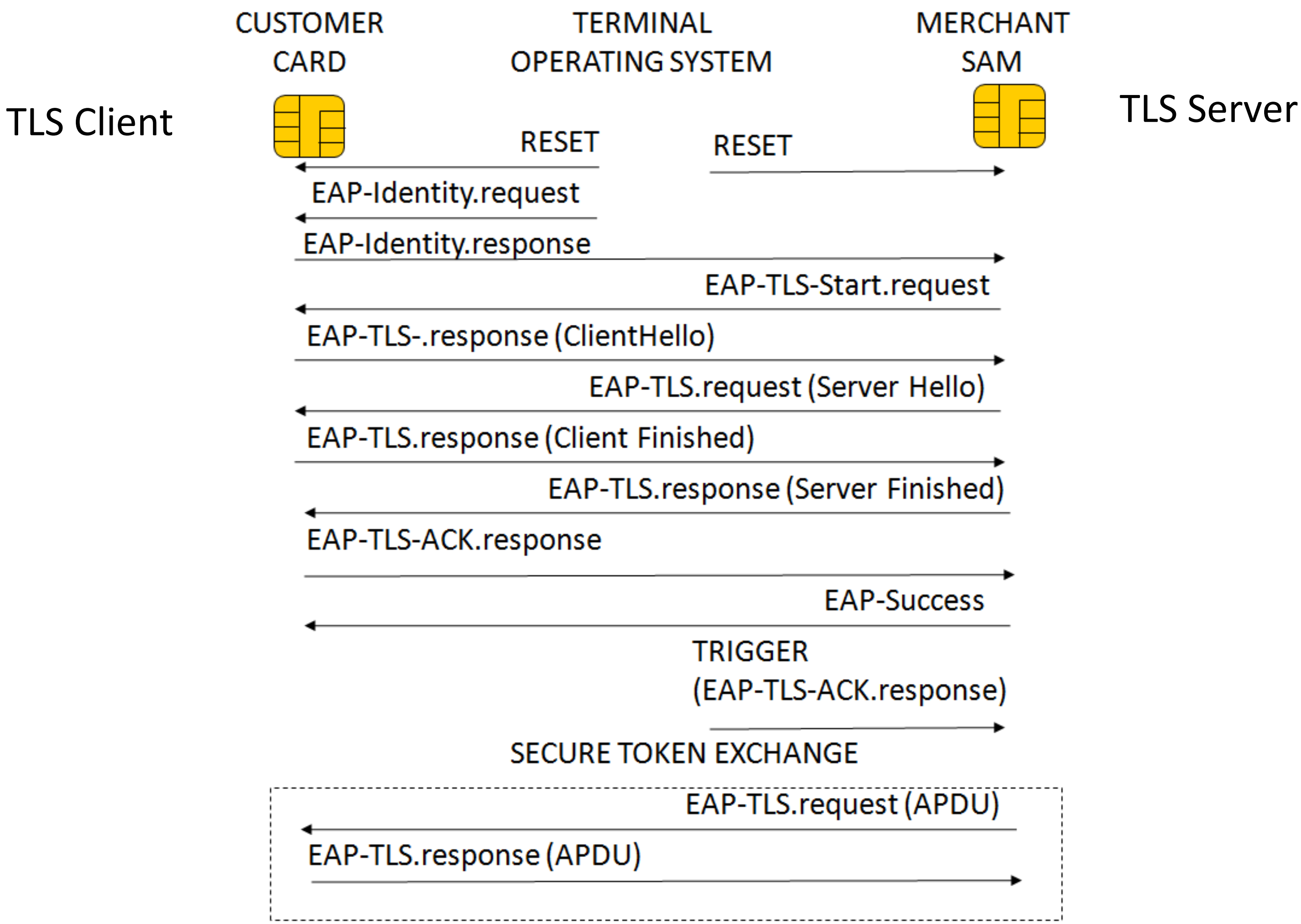
Legacy Mifare Lock

Pre-Payment System

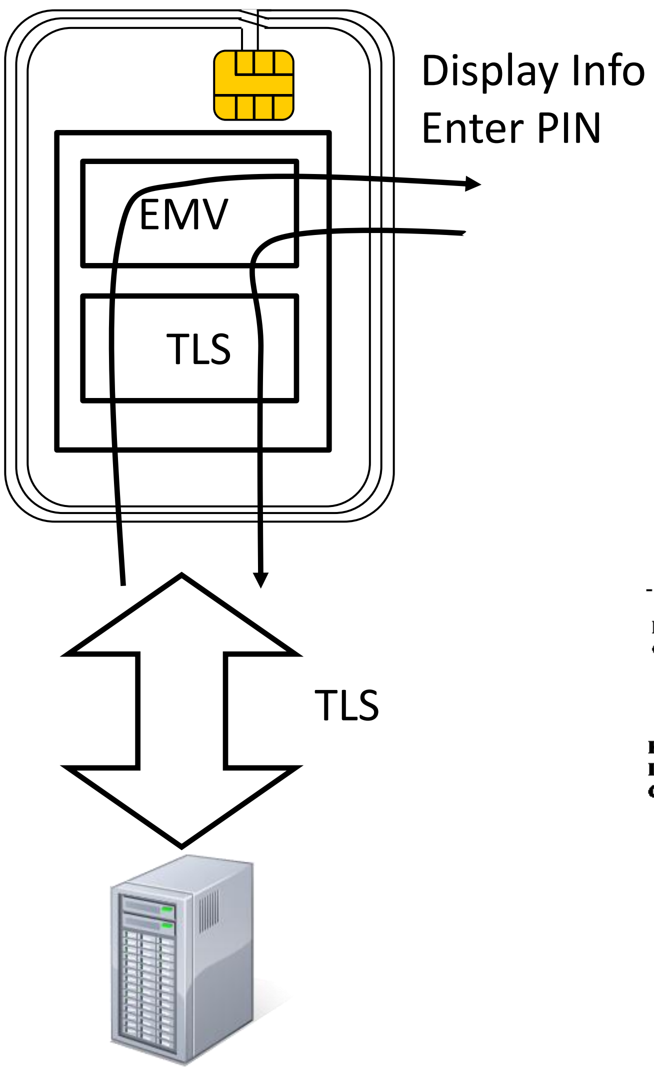


RSA Pub 1024	RSA Priv 1024	MD5 /bloc 64B	SHA1 /bloc 64B	3xDES /bloc 8B	AES /bloc 16B	RC4 /byte
25	550	0,50	0,90	2,10	2,60	0,50

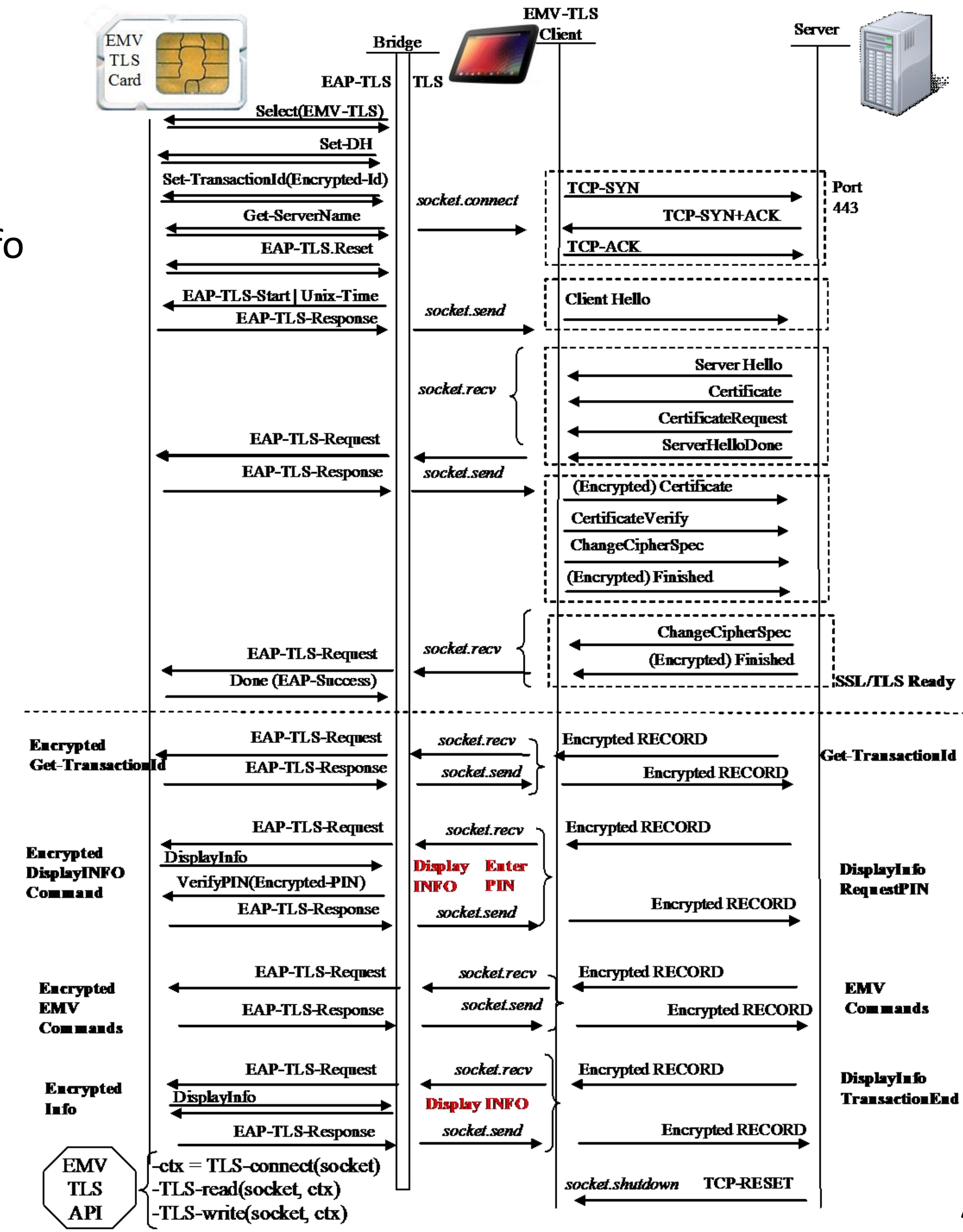
A Transaction Scenario



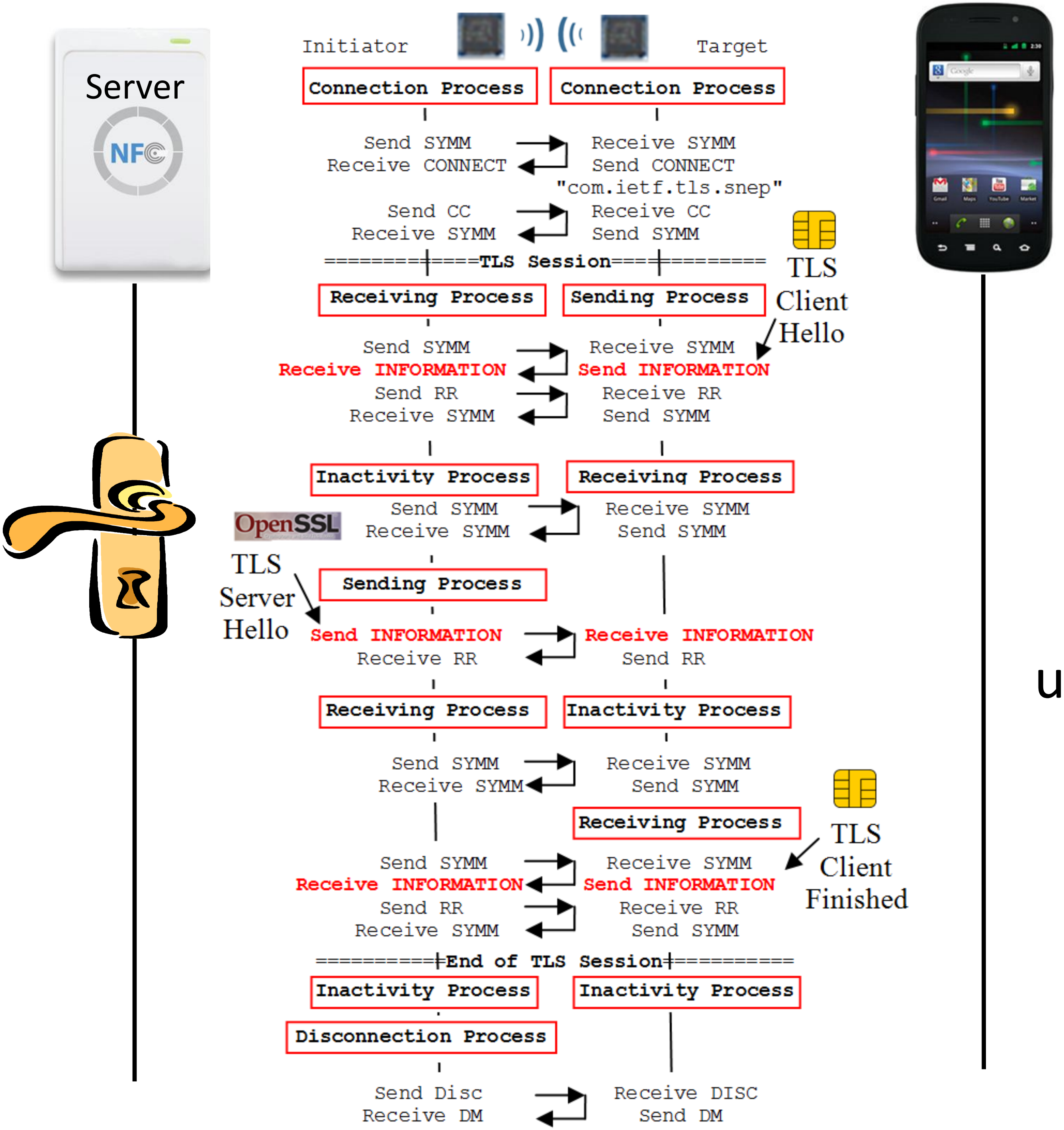
EMV-TLS



Remote EMV Use



LLCPS



Secure NFC P2P

TLS over LLCP:

LLCPS

ServiceName:

urn:nfc:sn:tls:snep

See

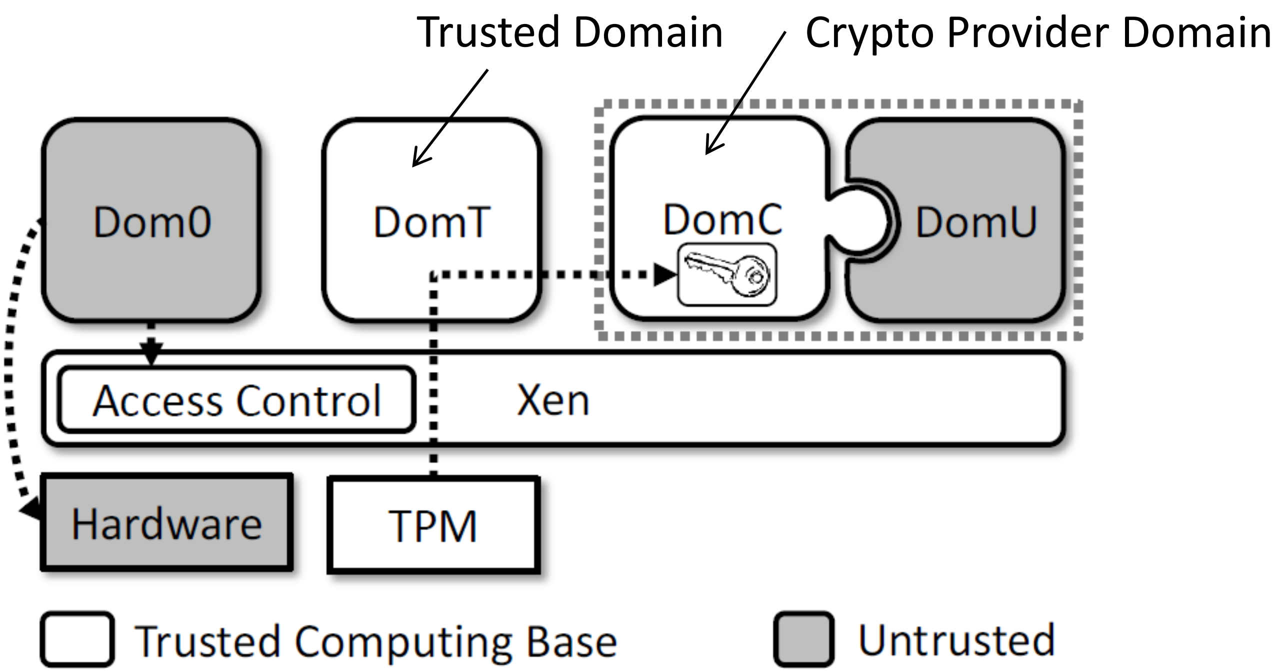
draft-urien-tls-llcp-00

For details

Ideas About Secure Elements in The Cloud

Cryptography as a Service, Tcloud

TClouds http://www.tclouds-project.eu/



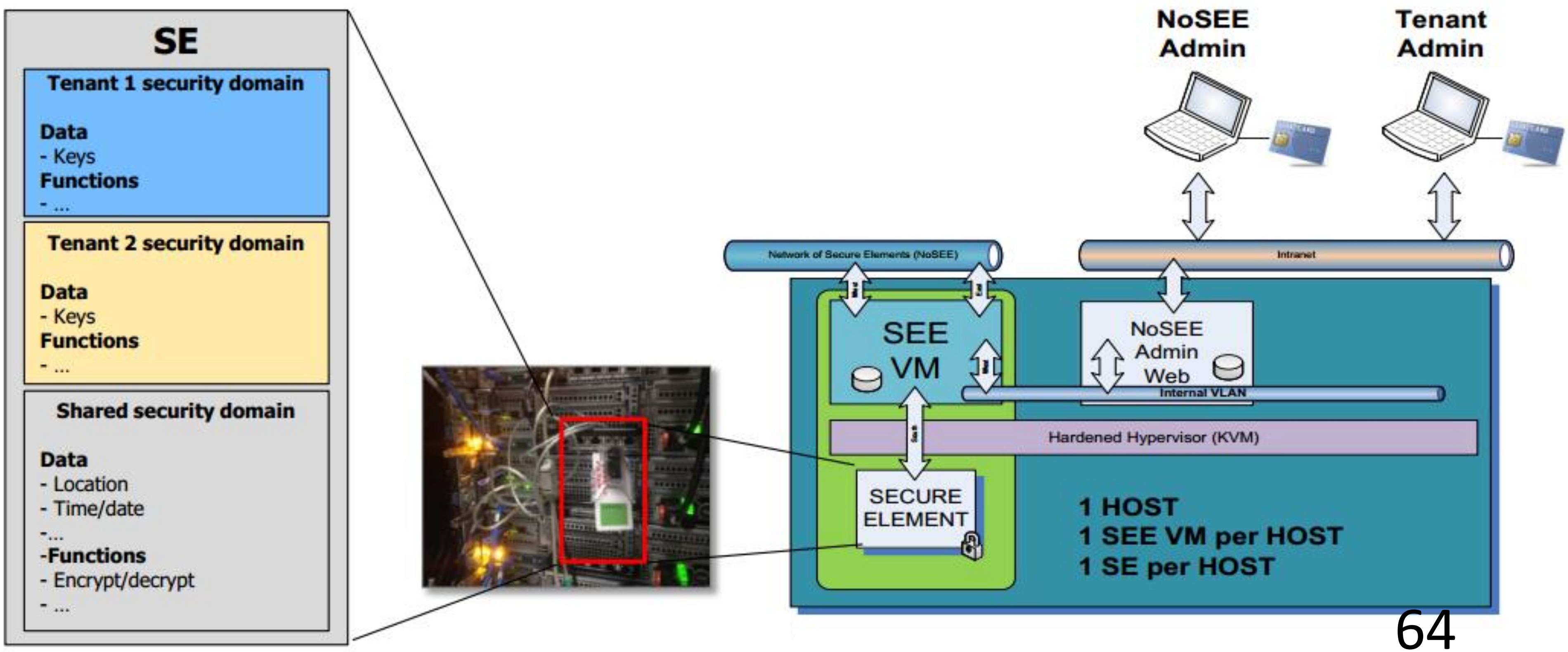
"Client-controlled Cryptography-as-a-Service in the Cloud"
Sören Bleikertz, Sven Bugiel, Hugo Ideler, Stefan Nürnberger, Ahmad-Reza Sadeghi

SEED4C, Secured Embedded Element for Cloud

SEE. Secure Element Extended

NoSEE. Network of Secure Element Extended

http://www.celticplus-seed4c.org/



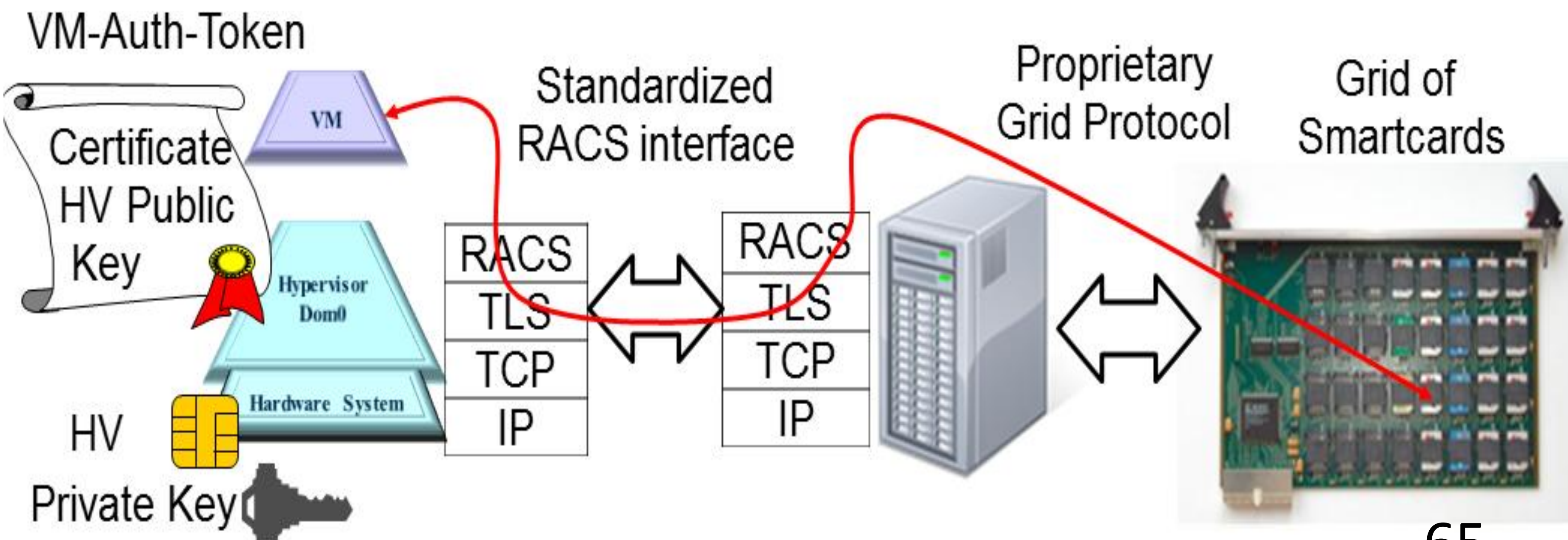
SecFuNet: Identity for VM

https://www.youtube.com/watch?v=Qxk84xdDDBk

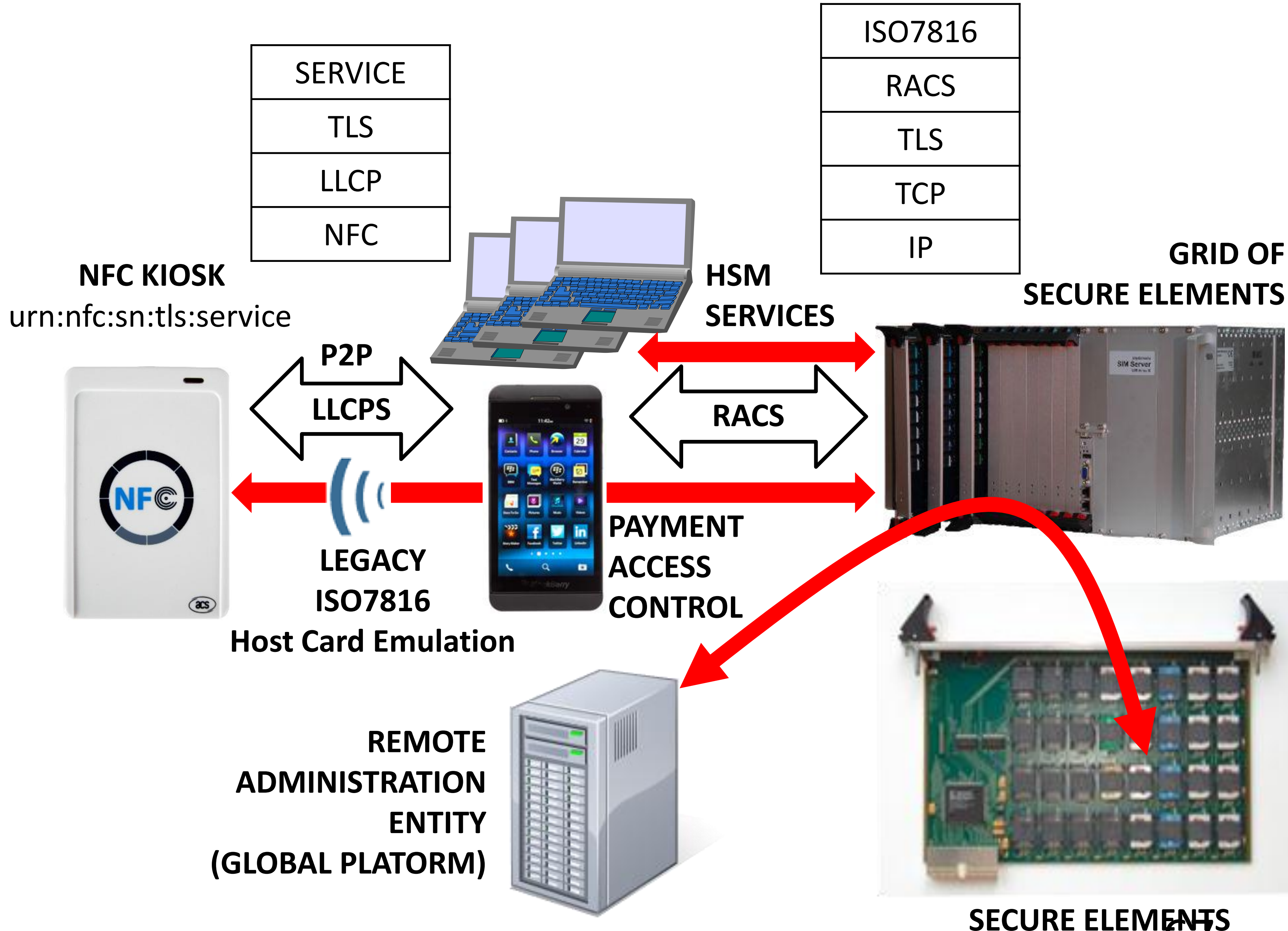
http://www.secfunet.eu

European Brazilian Project

Remote APDU Call Secure (RACS): http://tools.ietf.org/html/draft-urien-core-racs-00



Cloud of Secure Elements


SERVICE
TLS
LLCP
NFC
ISO7816
RACS
TLS
TCP
IP
NFC KIOSK
urn:nfc:sn:tls:service
HSM
SERVICES
GRID OF
SECURE ELEMENTS
P2P
LLCPS
RACS
NFC
LEGACY
ISO7816
Host Card Emulation
PAYMENT
ACCESS
CONTROL
REMOTE
ADMINISTRATION
ENTITY
(GLOBAL PLATORM)
SECURE ELEMENTS

Four Components

- NFC kiosk
  - Connected to various information systems
- Mobile / Virtual Machines
  - HCE (or P2P) NFC mode
  - Dedicated Application(s)
  - Secure Element
  - RACS Servers
  - Host Secure Elements
- Administration Centers
  - Manage embedded software lifetime
  - Global Platform over RACS

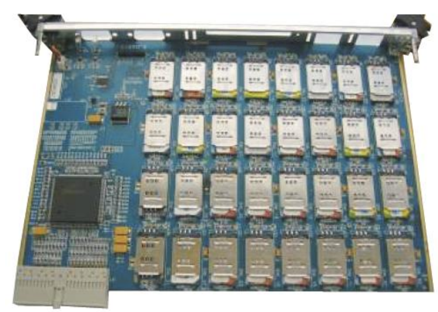
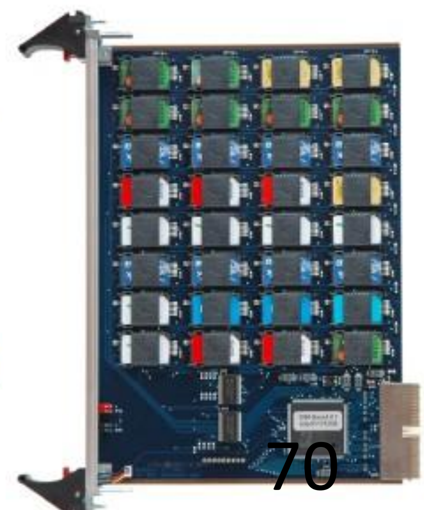
About RACS
Remote APDU Call Secure

http://tools.ietf.org/html/draft-urien-core-racs-00

Grid of Secure Elements (GoSE)

- Today mostly used as SIM-Server
- Up to 100,000 SIMs
  - Typical server capacity 512 SIMs
  - Typical daughter board capacity 32 SIMS
- Proprietary Protocols
- What about security ?

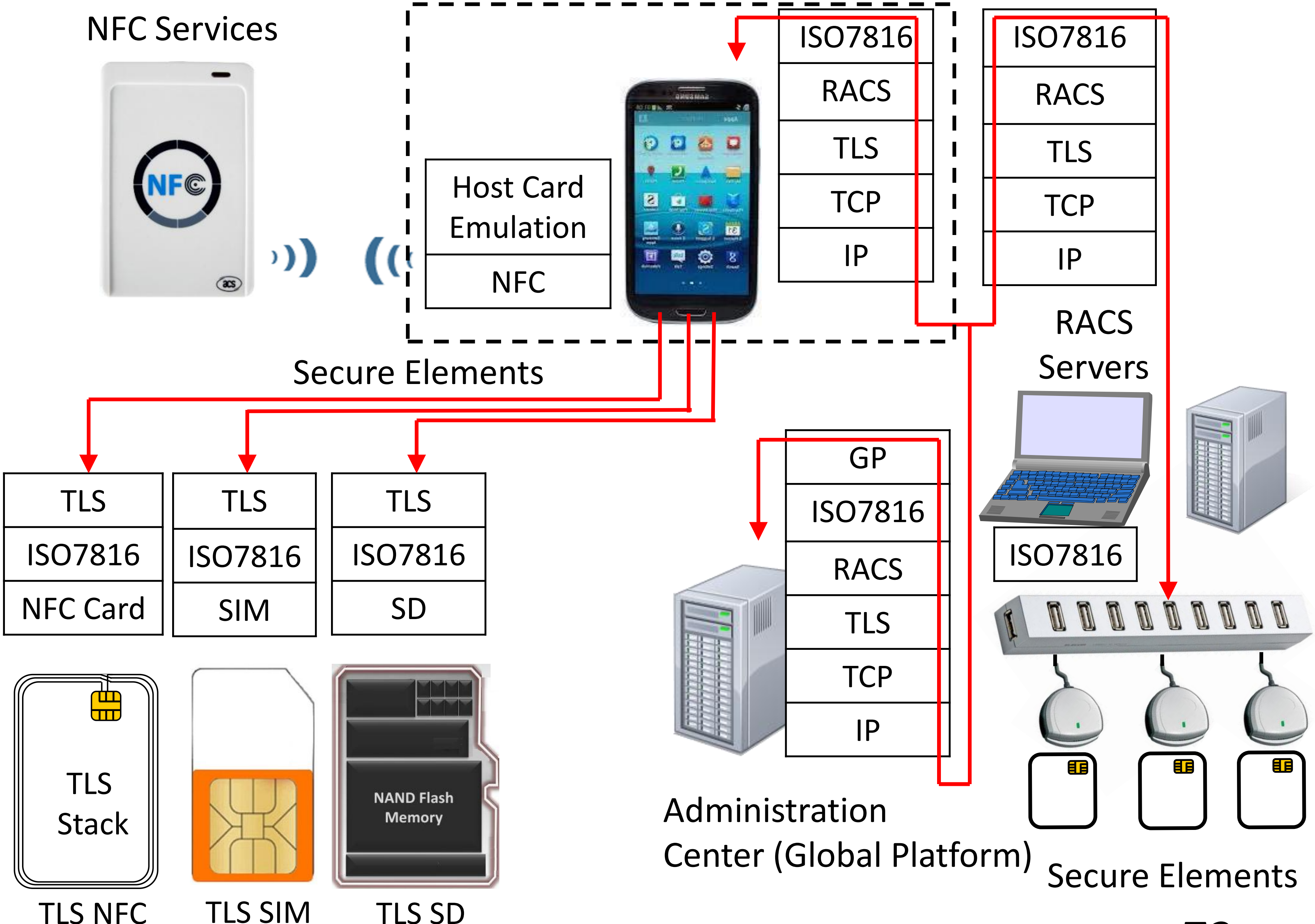
- What is needed for the CoSE ?
  - Very high scalability
  - Security
  - Openness
  - REST architecture

What is RACS ?

- Remote APDU Call Secure, the core of the CoSE
  - IETF draft, draft-urien-core-racs-00/03
- The RACS protocol provides all the features needed for the remote use of secure elements, i.e.
  - Inventory of Secure Elements
  - Information exchange (APDU) with the secure elements
- RACS works over the stack TLS/TCP/IP
- RACS is designed according to the representational State Transfer (REST) architecture, which encompasses the following features:
  - Client-Server architecture.
  - Stateless interaction.
  - Cache operation on the client side.
  - Uniform interface.
  - Layered system.
  - Code On Demand.

RACS
TLS
TCP
IP


NFC Services
Host Card Emulation
NFC
ISO7816
RACS
TLS
TCP
IP
ISO7816
RACS
TLS
TCP
IP
Secure Elements
RACS Servers
TLS
ISO7816
NFC Card
TLS
ISO7816
SIM
TLS
ISO7816
SD
GP
ISO7816
RACS
TLS
TCP
IP
ISO7816
TLS Stack
NAND Flash Memory
TLS NFC
TLS SIM
TLS SD
Administration Center (Global Platform)
Secure Elements

More about RACS

- A RACS request is a set of command lines
  - BEGIN
  - command command-parameter(s) CR LF
  - END
- A RACS request is a set of response lines
  - BEGIN
  - +/-status line-number parameter(s) CR LF
  - END
- Commands
  - BEGIN [label]
  - END
  - GET-VERSION [APPEND]
  - SET-VERSION [APPEND]
  - LIST   [APPEND]
  - RESET SEID [WARM] [APPEND]
  - APDU SEID ISO7816-request [CONTINUE=] [FETCH=] [MORE=] [APPEND]
  - POWERON   SEID   [APPEND]
  - SHUTDOWN   SEID   [APPEND]

The APDU Command

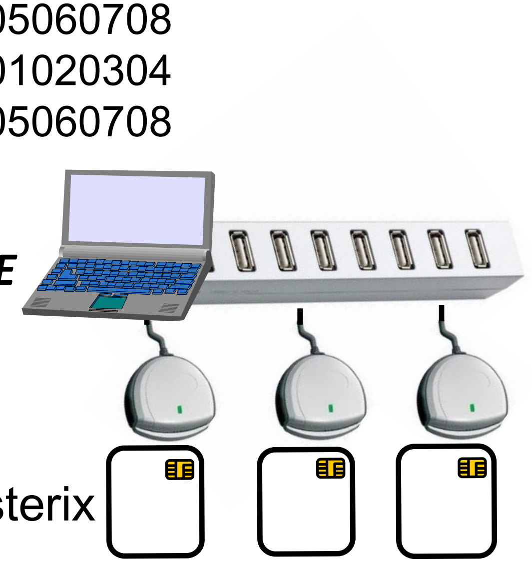
```
1. BODY = empty;
2. SW    = empty;
3. DoIt  = true;
3. Do
4. { iso7816-response = send(iso7816-request);
5.   body || sw1 || sw2 = iso7816-response;
6.   If ( (first request) && (iso7816-request.size==5)  && (body==empty) && (sw1==6C) )
8.   { iso7816-request.P3 = sw2 ; }
6.   Else
7.   { SW = sw1 || sw2
8.     BODY = BODY || body;
9.     If (sw1 == MORE)
10.    { iso7816-request = FETCH || sw2 ; }
11.    Else
12.    { DoIt=false;}
13.   }
14. }
15. While (DoIt == true) ;

16. iso7816-response = BODY || SW ;
17. If (SW != CONTINUE) Error   ;
18. Else                        No Error;
```

RACS Example

RACS REQUEST RACS://server.com:port/asterix

BEGIN TestMuscle
POWERON asterix APPEND
APDU asterix 00A4040006A00000000101 APPEND
APDU asterix B04200000830303030303030 APPEND
APDU asterix B036000105000301000 0 APPEND
APDU asterix
B0360003830100800102030405060708010203040506070801020304
0506070801020304050607080102030405060708010203040506070 8
0102030405060708010203040506070801020304050607080102030 4
0506070801020304050607080102030405060708010203040506070 8
010203040506070801020304050607080102030405060708
END

RACS RESPONSE



asterix

BEGIN TestMuscle
+008 001 SEID asterix has been powered on
+006 002 9000
+006 003 9000
+006 004 9000
+006 005
00809F2C64B365000A5CE0B3D235CDCD4610D75FFAEE50FF6EB2803573C
2B8B940DDB75B3A2E08AF933BFDFDC180DBF09A9D191A9FBD5C46731AF
7B810E5A76B79D243C0EEFEDE490F5CB75370670D85CF137AAF7D89FA31
444BD466B9B828B7E0DBFAF62874A79D837369EFE9BAF055EB9601B3F70
42C424F0CDA99EFF75805582B9000
END

What is a SEID
RACS://server.com:port/seid

Grid Slot ⟷ SEID

Key Diversification Data ⟷ SEID

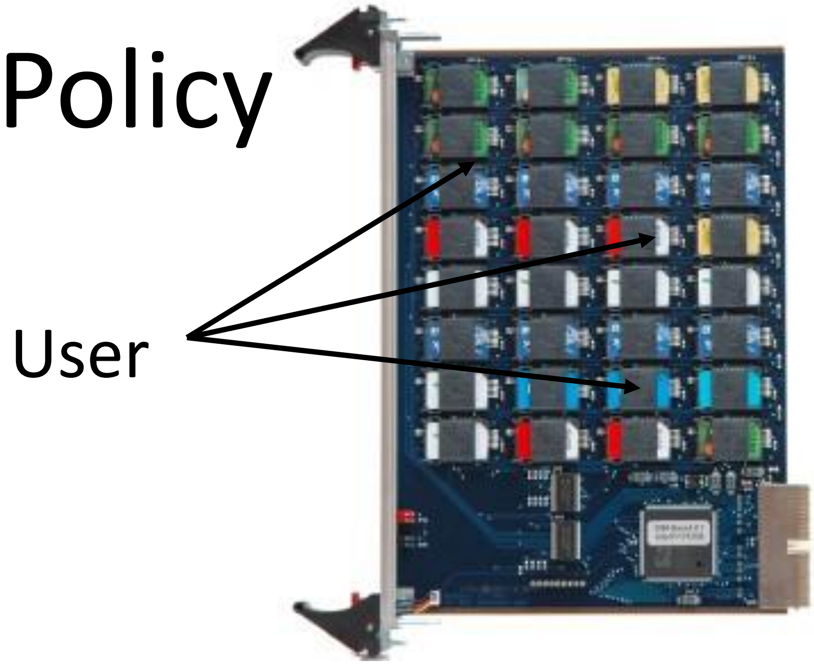
The GP command 'initialize update' is used to start a mutual authentication between the administration entity and the secure element; it collects a set of data whose first ten bytes are called the 'key diversification data'. This information is used to compute symmetric keys, and MAY comprise a serial number.

Reader Serial Number ⟷ SEID

According to the PC/SC standard a smart card reader may include a serial number. This attribute (VENDOR-IFD-SERIAL) is associated to the tag 0x0103 in the class VENDOR-INFO.

Security Policy

- 2 main tables
  - Users Table
  - SEID Table



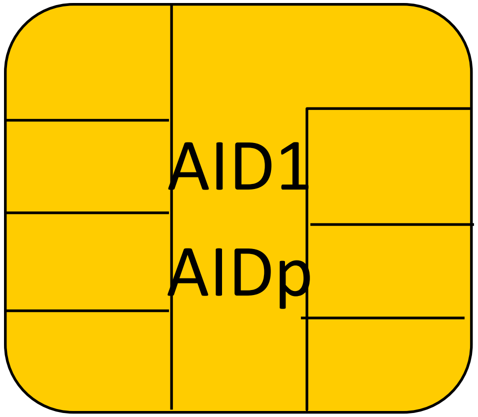
Users Table

Common Name	List of SEIDs
user1	seid1, seid2, seidp

SEID Table

AID	List of Users
aid1	user1, user2, userp

SEID



Questions

Some Papers

- Urien, P.; "LLCPS: A New Security Framework Based on TLS For NFC P2P Applications in the Internet of Things", IEEE CCNC 2013, January 11-14, Las Vegas, Nevada, USA
- LLCPS, draft-urien-tls-llcp-02.txt, IETF draft, 2013
- Remote APDU Call Secure (RACS), draft-urien-core-racs-00.txt, IETF draft 2013
- Urien, P., Piramuthu, S., "Towards a Secure Cloud of Secure Elements Concepts and Experiments with NFC Mobiles", in proceeding of the CTS 2013 conference, May 2013.
- Urien, P.; "Cloud of Secure Elements, Perspectives for Mobile and Cloud Applications Security", First IEEE Conference on Communications and Network Security, October 14-16 2013 Washington D.C., USA.
- Urien, P., Piramuthu, S., "Securing NFC Mobile Services with Cloud of Secure Elements (CoSE)" – Mobicase 2013, November 2013, Paris, France
- Urien, P., "A Secure Cloud of Electronic Keys for NFC Locks Securely Controlled by NFC Smartphones", IEEE CCNC 2014
- Aissaoui-Mehrez, H. ; Urien, P. ; Pujolle, G., Global identity management of virtual machines based on remote secure elements, International Conference on Computer, Information and Telecommunication Systems (CITS), Jeju Island, South Korea, 2014
- Urien, P. "Cloud of Secure Elements: An Infrastructure For the Trust of Mobiles NFC Services", The 10th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications WiMob 2014, Larnaca, Cyprus
- Aissaoui-Mehrez, H. ; Urien, P. ; Pujolle, G., "Implementation Software To Secure Virtual Machines With Remote Grid of Secure Elements", Military Communications Conference, MILCOM 2014, Baltimore, USA