NFC Application Security

Sandeep Tamrakar

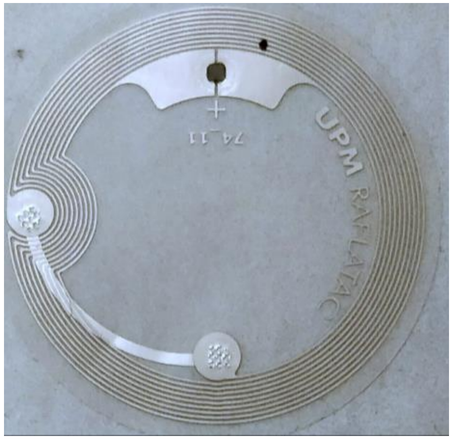
08-11-2018

NFC

- Short-range, high-frequency RFID (Radio Frequency Identity)
- Collections of Data transmission standards
  - ISO 14443, ISO 15693, ISO 18092
- Operating distance
  - 4 cm to 10 cm
- Operating Frequency
  - 13.56 MHz
- Data rates "of NFC radio"
  - 106 kbit/s, 212 kbit/s, 424 kbit/s
- Communication between two devices:
  - E.g. Reader and a Contactless card
- NFC Forum defines:
  - Interoperability
  - NFC application specification

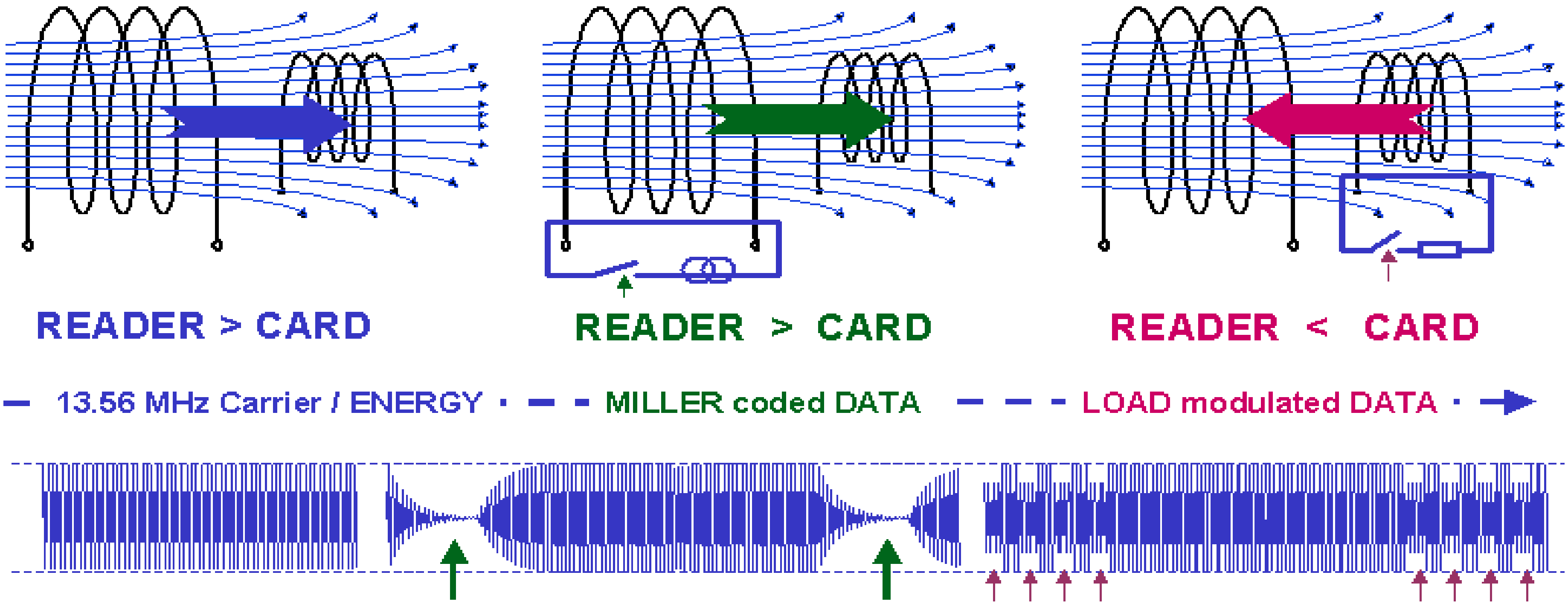
NFC devices

- Active Device (reader)
  - Proximity coupling device (PCD)
  - Connected to power source
  - Generates an electromagnetic field for data exchange



- Passive device (NFC tag)
  - Proximity integrated circuit card (PICC)
  - Harvest power from an Active device

NFC data exchange principle



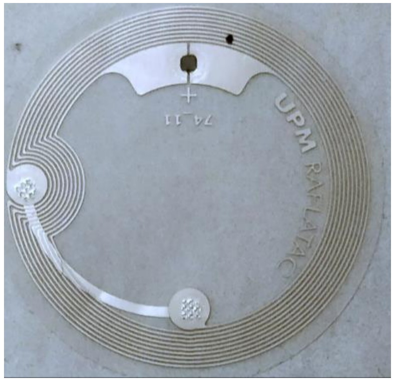
READER > CARD

READER > CARD

READER < CARD

13.56 MHz Carrier / ENERGY · MILLER coded DATA LOAD modulated DATA ·

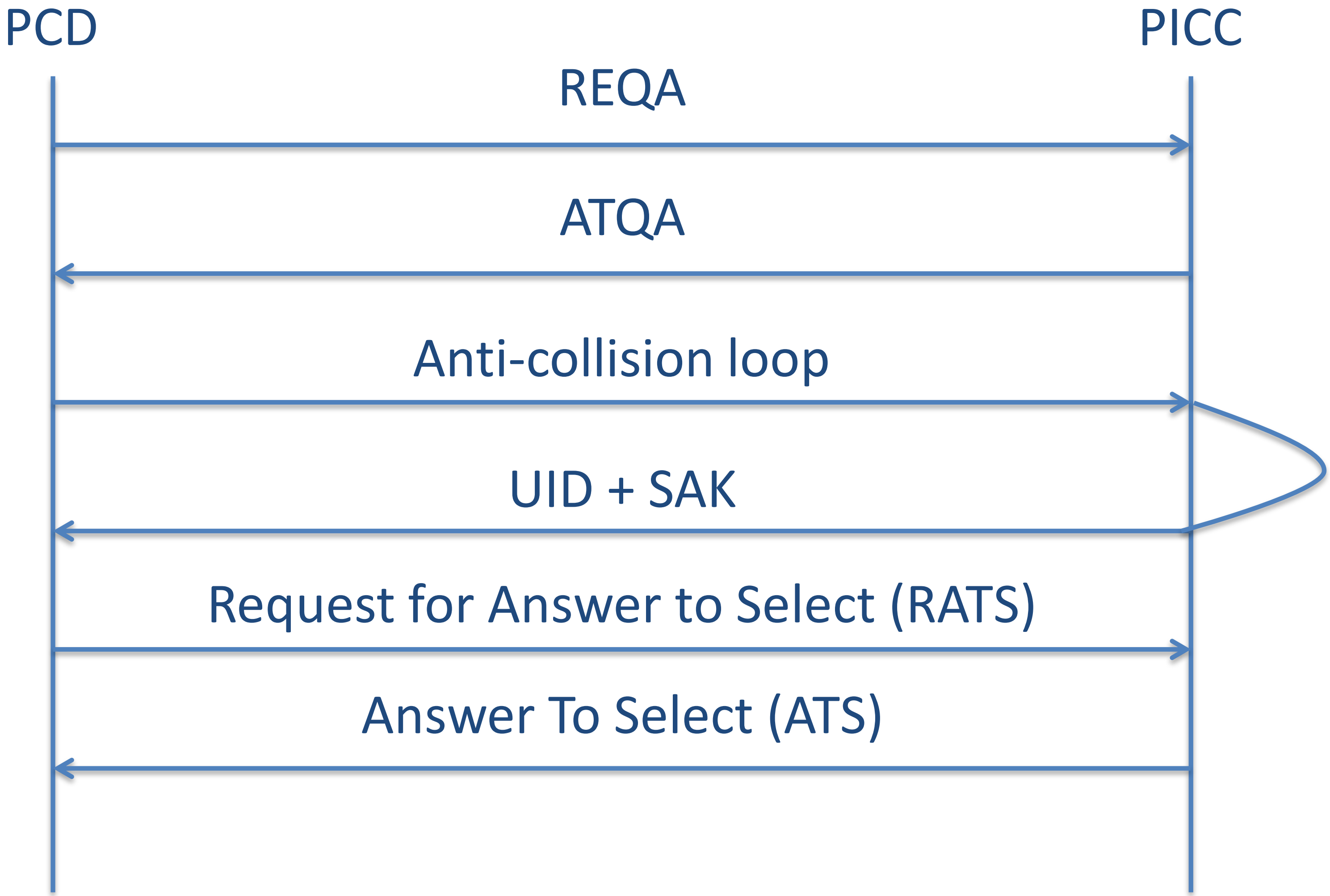
Reader



Card

Source: Mifare® (14443A) 13.56 MHz RFID Proximity Antennas www.nxp.com/documents/application_note/AN78010.pdf

Example: Interaction between a PCD and a PICC



PCD

PICC

REQA

ATQA

Anti-collision loop

UID + SAK

Request for Answer to Select (RATS)

Answer To Select (ATS)

Active NFC device modes of operation

- Reader / Writer Mode (PCD, ISO 14443)
  - Active device that transmits power
  - Reads and modifies data stored in passive tag
  - E.g. Mobile phone reading smart poster
- Card Emulation Mode (PICC, ISO 14443)
  - Acts like a passive target
  - Interacts with external active readers
  - E.g. Mobile phone used as transport ticket, Google Wallet
- Peer-to-peer Mode (ISO 18092)
  - Both initiator and target transmit power
  - Bi-directional data channel
  - E.g. Transferring files between Android phones via NFC, Android Beam

Passive NFC devices

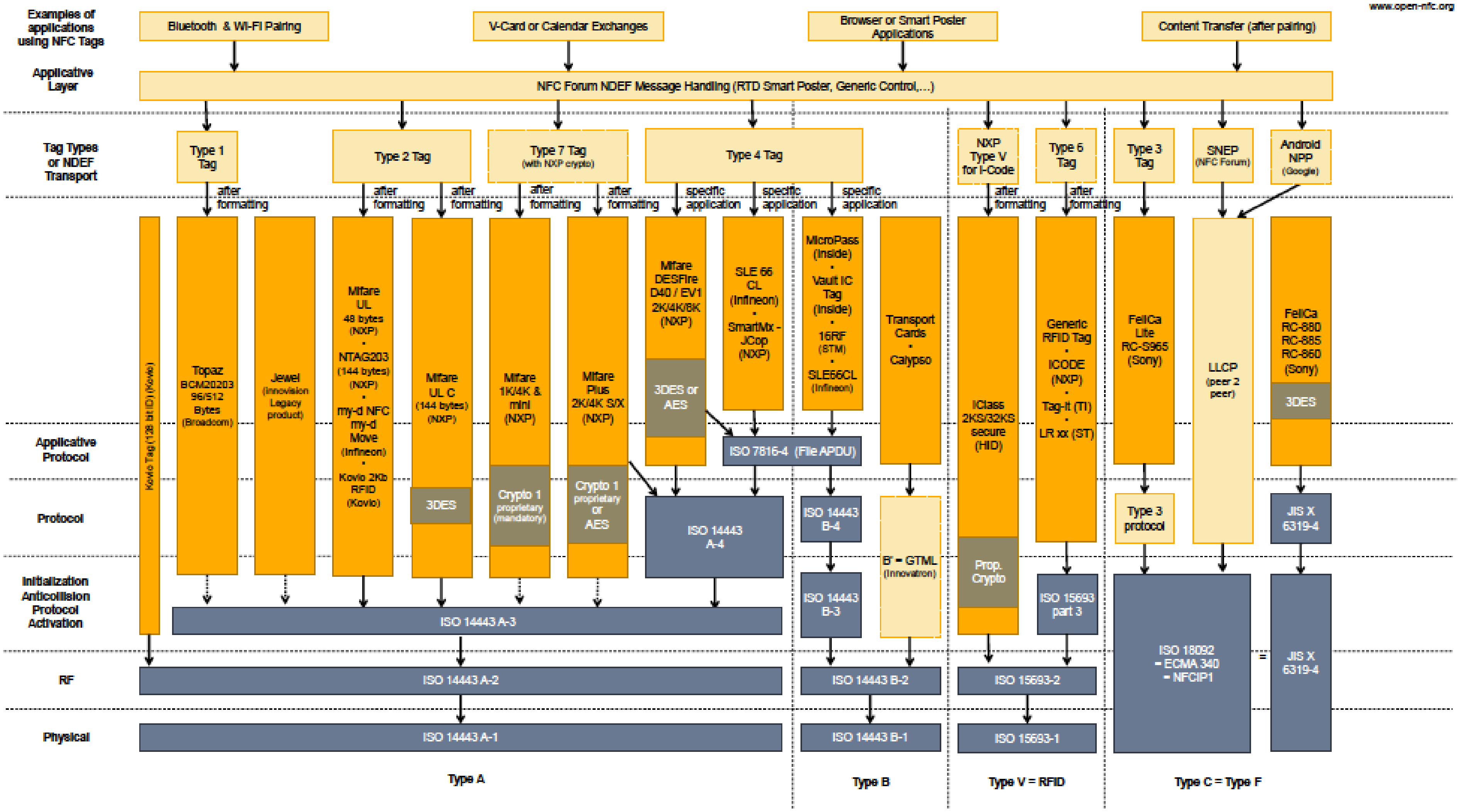
NFC tags and smart cards

- Memory tags (Type 1 tags)
- Memory tags with access control (Type 2 tags)
- Tags with cryptographic hardware (Type 4, Type 7 tags)
- Programmable contactless smart cards (Type 4: JCop tags)

NFC Standards, Products and Specifications

Version 1.9
jbblanchet@insidefr.com

www.open-nfc.org



Examples of applications using NFC Tags: Bluetooth & Wi-Fi Pairing; V-Card or Calendar Exchanges; Browser or Smart Poster Applications; Content Transfer (after pairing)

Applicative Layer: NFC Forum NDEF Message Handling (RTD Smart Poster, Generic Control,…)

Tag Types or NDEF Transport: Type 1 Tag; Type 2 Tag; Type 7 Tag (with NXP crypto); Type 4 Tag; NXP Type V for I-Code; Type 6 Tag; Type 3 Tag; SNEP (NFC Forum); Android NPP (Google)

after formatting · after formatting · after formatting · after formatting · after formatting · specific application · specific application · specific application · after formatting · after formatting

Kovio Tag (128 bit ID) (Kovio)

Topaz BCM20203 96/512 Bytes (Broadcom)

Jewel (Innovision Legacy product)

Mifare UL 48 bytes (NXP) • NTAG203 (144 bytes) (NXP) • my-d NFC my-d Move (Infineon) • Kovio 2Kb RFID (Kovio)

Mifare UL C (144 bytes) (NXP) — 3DES

Mifare 1K/4K & mini (NXP) — Crypto 1 proprietary (mandatory)

Mifare Plus 2K/4K S/X (NXP) — Crypto 1 proprietary or AES

Mifare DESFire D40 / EV1 2K/4K/8K (NXP) — 3DES or AES

SLE 66 CL (Infineon) • SmartMx - JCop (NXP)

MicroPass (Inside) • Vault IC Tag (Inside) • 16RF (STM) • SLE66CL (Infineon)

Transport Cards • Calypso

iClass 2KS/32KS secure (HID) — Prop. Crypto

Generic RFID Tag • ICODE (NXP) • Tag-it (TI) • LR xx (ST)

FeliCa Lite RC-S965 (Sony)

LLCP (peer 2 peer)

FeliCa RC-880 RC-885 RC-860 (Sony) — 3DES

Applicative Protocol: ISO 7816-4 (File APDU)

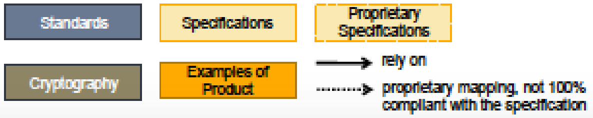
Protocol: ISO 14443 A-4; ISO 14443 B-4; B' = GTML (Innovatron); Type 3 protocol; JIS X 6319-4

Initialization Anticollision Protocol Activation: ISO 14443 A-3; ISO 14443 B-3; ISO 15693 part 3; ISO 18092 = ECMA 340 = NFCIP1 = JIS X 6319-4

RF: ISO 14443 A-2; ISO 14443 B-2; ISO 15693-2

Physical: ISO 14443 A-1; ISO 14443 B-1; ISO 15693-1

Type A · Type B · Type V = RFID · Type C = Type F



Standards · Specifications · Proprietary Specifications · Cryptography · Examples of Product

rely on

proprietary mapping, not 100% compliant with the specification

Example: Type 2 tag (MIFARE Ultralight) memory layout

Page address	Byte Numbers			
00h	UID0	UID1	UID2	BCC0
01h	UID3	UID4	UID5	UID6
02h	BICC1	INT	LOCK0	LOCK1
03h	OTP0	OTP1	OTP2	OTP3
04h	Data0	Data1	Data2	Data3
05h	Data4	Data5	Data6	Data7
….	….	….	….	….
0Fh	Data44	Data45	Data46	Data47

Bits	7	6	5	4	3	2	1	0
LOCK0	L7	L6	L5	L4	L3	BL 15-10	BL 9-4	BL OTP
LOCK1	L15	L14	L13	L12	L11	L10	L9	L8

- Byte 0 – 9 : read only
- Byte 10 – 15: One time programmable (OTP) bytes; Once a bit in an OTP byte is set, it cannot be reset back.
- L : Lock page
- BL: Block lock, Once a BL bit is set the locking configuration for the corresponding page is unchangeable.

MIFARE Ultralight with NDEF data

```
# Memory content:
[00] * 04 E4 91 F9 (UID0-UID2, BCC0)
[01] * CA 1A 26 80 (UID3-UID6)
[02] . 76 48 00 00 (BCC1, INT, LOCK0-LOCK1)
[03] . E1 10 06 00 (OTP0-OTP3)
[04] . 03 0D D1 01 |....|
[05] . 09 55 01 61 |.U.a|
[06] . 61 6C 74 6F |alto|
[07] . 2E 66 69 FE |.fi.|
[08] . 00 00 00 00 |....|
[09] . 00 00 00 00 |....|
[0A] . 00 00 00 00 |....|
[0B] . 00 00 00 00 |....|
[0C] . 00 00 00 00 |....|
[0D] . 00 00 00 00 |....|
[0E] . 00 00 00 00 |....|
[0F] . 00 00 00 03 |....|
  *:locked & blocked, .:un(b)locked
```

Manufacture Defined bytes

Lock bytes

OTP bytes

NDEF data

□ UID

OTP data **E1:10:06:00** defines NDEF application

For detail on NDEF format, see NFC forum NFC Data Exchange Format (NDEF) Technical specification

One-Time Programmable bits

- Writing values on OTP bits
  - ORs current value with new value
- E.g.
  - 00000000 00000000 00000000 00000000
  - Write: 0x000011AE (10001 10101110)
  - 00000000 00000000 00010001 10101110
  - Write: 0x00001523 (10101 100011)
  - 00000000 00000000 00010101 10101111

Features of NFC tag types

	Type 1 Tags	Type 2 Tags	Type 3 Tags	Type 4 Tags
Unique Identity	4 or 7 bytes	4 or 7 bytes	8 bytes	7 bytes
Transmission Protocol	ISO 14443A	ISO 14443A	ISO 18092	ISO 14443A
Memory Size	96 bytes (up to 2 KB)	64 bytes (up to 2 KB)	Variable sizes (up to 1 MB)	Variable sizes (up to 32KB)
Memory Organization	12 blocks, each of 8 bytes	16 pages, each of 4 bytes	Blocks, each of 16 bytes	Smart card based.
OTP bits	48 bits	32 bits		
Lock bits	16 bits	16/32 bits		
Re-writable	Until locked	Until locked	Pre-defined	Issuer-defined
Data collision protection	No	Yes	Yes	Yes
Transmission speed	106 kbits/s	106 kbits/s	212 kbits/s or 424 kbits/s	106 kbits/s, 212 kbits/s, 424 kbits/s
Examples	Topaz	Ultralight	FeliCa Lite	Java cards, DESFire

Contactless smart cards

- Memory tags with some security functionality
  - MIFARE Ultralight: UID, lock bytes, OTP
  - Ultralight C: triple-DES authentication
  - DESFire EV1: Triple-DES / AES mutual authentication, file system with access control lists
- Smart cards with contactless interface
  - CPU and operating system
  - Tamper-resistant processing environment
  - Secure crypto-processor
  - Secure file system
  - E.g. JavaCard, EMV contactless debit and credit cards
- *Distinction between memory cards and smart cards is not always clear cut*

Threats on memory tags

- Tag cloning
  - E.g. Location check-in tags can be cloned to falsely claim that you have been at the location (to claim loyalty discounts)
  - Prevented to some degree by calculating MAC that includes UID.
- Modification of tag data
  - Prevented by locking tag re-write
- Swapping / replacing valid tags
  - E.g. Tags used to help purchase items from vending machines can be swapped so that when a customer tries to purchase an item from a vending machine, the immoral person waiting at the other vending machine gets the purchased item.

NFC tag security

- Security mechanisms:
  - 7-byte Unique Identity (UID)
  - One-time programmable bits: bits that can be set to one but not reset to zero
    - can be used as counter
  - Pages can be locked to prevent modification
- Security assumptions:
  - The UID cannot be cloned or spoofed !!!?
  - When reading the tag, the UID and card content cannot be modified by the attacker (physical session integrity) !!!?

Cloning Ultralight tags

- Clonable cards are available
  - Rewrite the entire memory area including UID, OTP and Lock bytes
- Demo: Ultralight Tag cloning

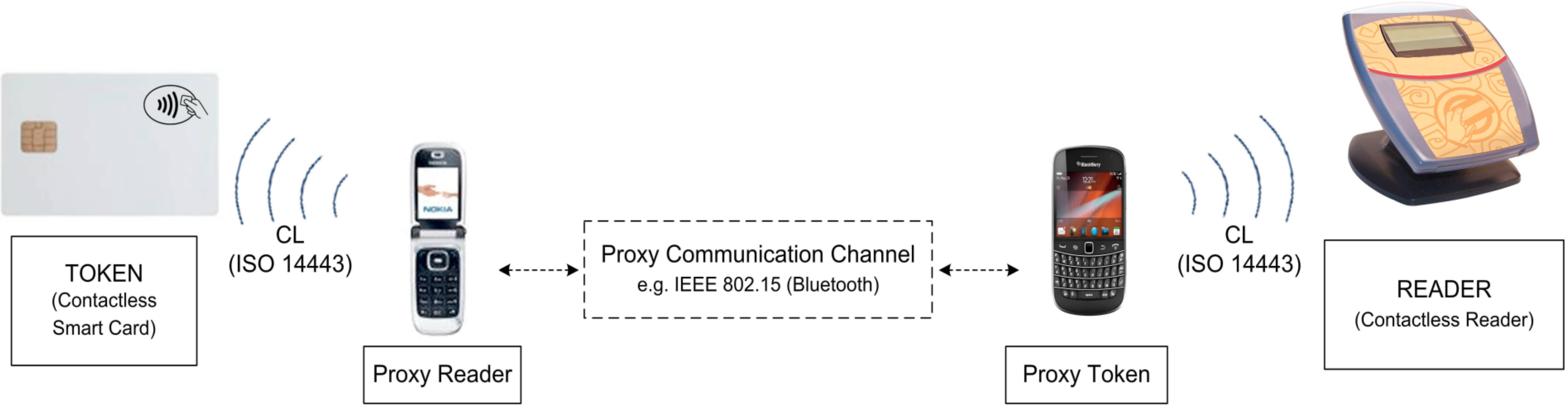
Ultralight C

- Memory organization is similar to Ultralight
- More memory 192 bytes
- Additional 32-bit one way counter
- Access control using an authentication key
    - Write protected or both read/write protected

No Cryptographic security included in the NFC Specs

- NFC transmission protocols do not define any specific encryption or security mechanism
  - ISO 14443 : Read/Write and Card Emulation mode
  - ISO 18092: Peer-to-peer mode
- NDEF specification defines signature scheme for integrity protection
  - Does not prevents content cloning (signature does not cover the card UID)
  - Does not include reader authentication for access control
- Therefore, cryptographic security must be defined by the NFC application.

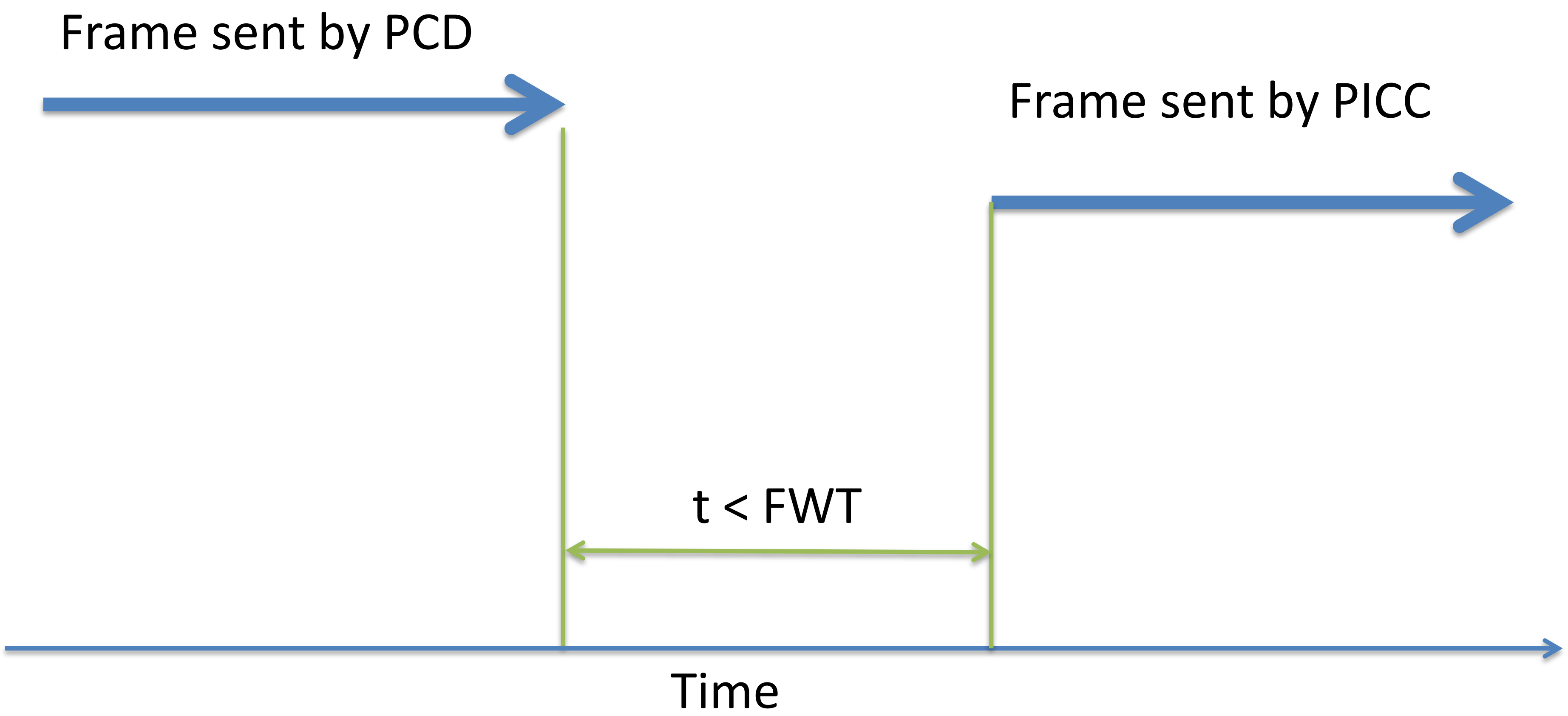
Relay Attack on NFC



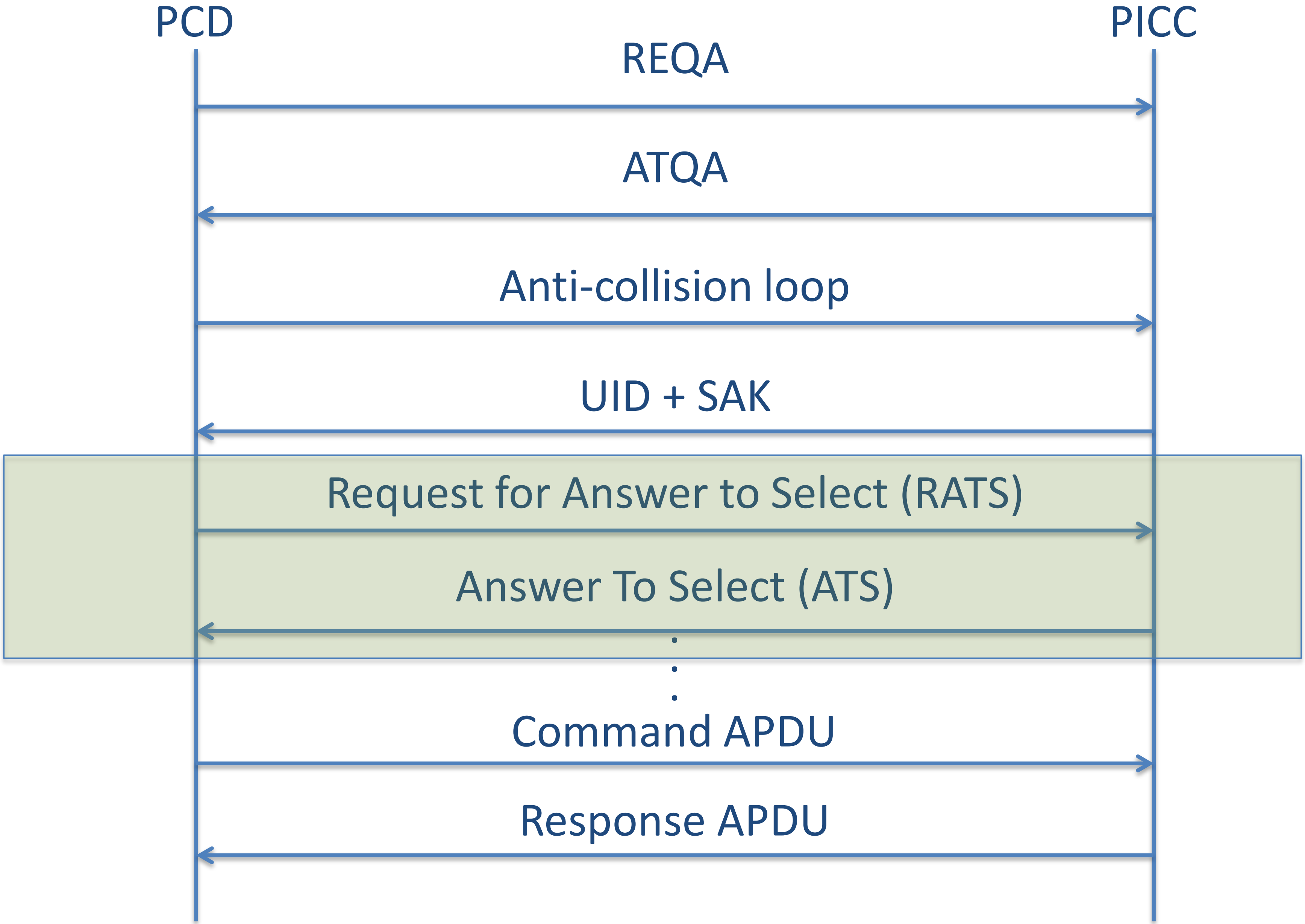
- Relaying e.g. contactless EMV payments from your pocket to a faraway shop
  - Requires card emulation on the proxy token
  - Does not require UID spoofing because EMV does not use the UID

Source: L. Francis, G. P. Hancke, K. E. Mayes, and K. Markantonakis. Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones. Cryptology ePrint Archive, Report 2011/618, 2011. http://eprint.iacr.org/2011/618.

Frame Waiting Time (FWT)



NFC reader and tag interaction



FWT parameter

Answer To Select (ATS)	
TL	Length Byte
T0	Format Byte
TA	Interface Bytes
TB	
TC	
T1	Historical Bytes (ISO/IEC 7816-4)
Tk	
CRC 1	
CRC 2	

TB	
FWI	SFGI

FWI: Frame Waiting Time Integer

SFGI: Start-up Frame Guard Time

$$FWT = (256 \text{ X } 16 / fc) \text{ X } 2^{FWI}$$

$FWT_{Min} = 0\text{: } (256 \text{ X } 16/ 13.56 \text{ X } 10^6) \text{ X } 1 \approx 303\ \mu s$

$FWT = 4\text{: } (256 \text{ X } 16/ 13.56 \text{ X } 10^6) \text{ X } 2^4 \approx 4833\ \mu s$

$FWT = 8\text{: } (256 \text{ X } 16/ 13.56 \text{ X } 10^6) \text{ X } 2^8 \approx 77\ ms$

$FWT_{Max} = 14\text{: } (256 \text{ X } 16/ 13.56 \text{ X } 10^6) \text{ X } 2^{14} \approx 4949\ ms$

Observation on Android Jelly Bean (4.1.2)

- MIFARE DESFire:
  - default FWI = 0x8
  - FWT = 77 ms
- Nexus S responded well beyond 77 ms (≈430ms)
- Changing FWI parameter doesn’t affect response time
- We assume fixed FWT implementation
- Readers often ignores FWT configurations

NFC on mobile phones

- Integration of NFC in mobile phone has grown significantly
  - Almost all new Android phones
  - iPhone 6 and above
  - BlackBerry 10
  - Windows phone (Obsolete)

NFC support on phones

- Mostly Read/write and P2P mode
- Some phone platform include Secure Element necessary for Card Emulation
- Host Card Emulation API is available on Android 4.4 and above
- Currently used applications
  - NDEF tag read/write
    - FourSquare check-ins
    - Samsung TectTiles
- Potential NFC applications:
  - Public transit tickets
  - Mobile payment (Apple pay, Google wallet)
  - Loyalty card
  - Access control to premises

Threats on NFC phones

- Denial of Service attacks
  - Mobile phone NFC stack reacts to any tag within its NFC range
  - Some mal-formatted tags can jam the stack
  - Also, most of the card manager in SE blocks itself after 10 successive authentication failures
- Malware delivery via NFC
  - Mobile OSs reads NDEF message and opens corresponding application
    - E.g. NDEF with URL causes phone to open the URL in its default web browser
  - NDEF with URL to malware download page
  - NFC message with malicious content
    - Malicious file over NFC to exploit android document viewer vulnerability [1]
    - NFC to execute Unstructured Supplementary Service Data (USSD) codes [2]

[1] https://www.hkcert.org/my_url/en/blog/12092801

[2] http://www.zdnet.com/exploit-beamed-via-nfc-to-hack-samsung-galaxy-s3-android-4-0-4-7000004510/

NFC based financial application on phone

- Rely on security offered by the mobile phone platform
  - Sandboxes
  - Permission based access control
- Protection against mobile malwares
- Protection against Ill intent of the phone user
  - E.g. User may gain root access to modify ticket value
- Protection against remote and local attacker
- Protection even when the OS is compromised

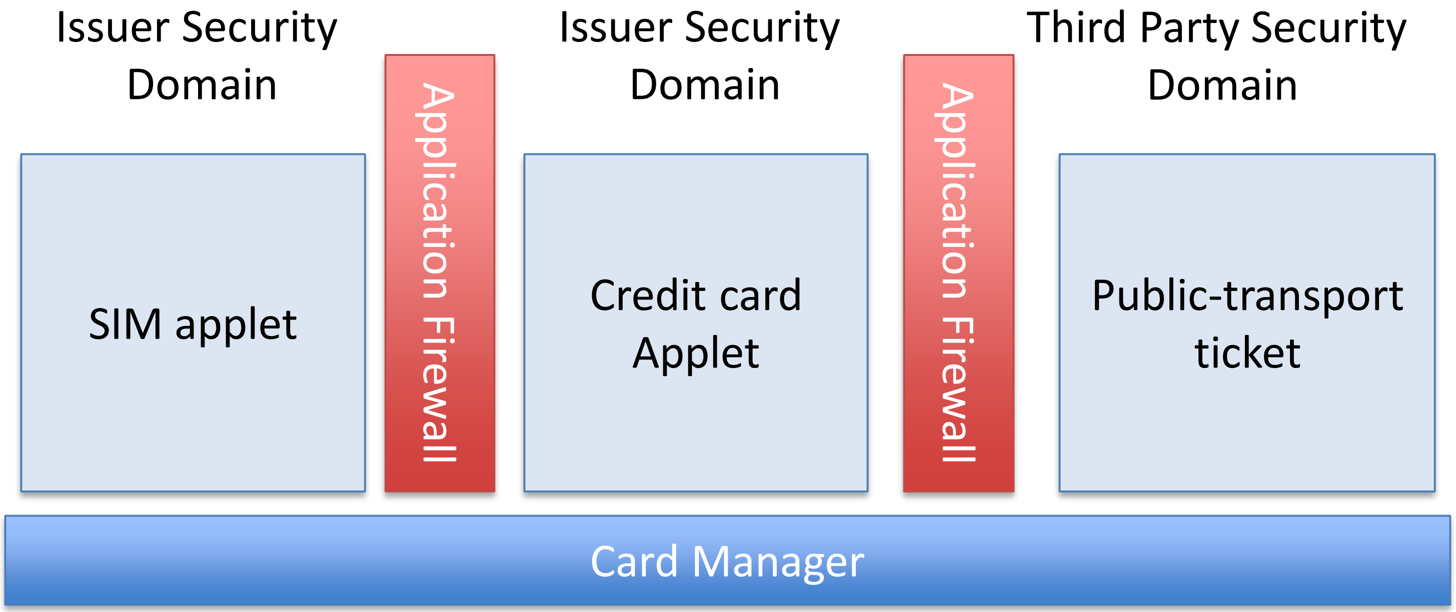
Secure execution on mobile phone

- Isolate Execution
  - Execution of a security-sensitive code in complete isolation from other codes
  - Ensures integrity and run-time secrecy of application data
- Secure Storage
  - Protects stored data from unauthorized access
    - Passwords, secret keys, credentials etc.
- Remote Attestation
  - Remotely verify authenticity of any particular application before interaction
  - Root of trust measurement
  - E.g. Key attestation in Android Keystore
- Secure Provisioning
  - Securely deploying application module or cryptographic keys to a specific user device from a remote server over the air
  - Application migration from one device to another
  - E.g. Key wrapper feature in Android Keystore
- Trusted path
  - Ensures unaltered communication channel between the end points
  - Direct physical connection between NFC front end controller and the isolated execution environment

Available Secure Execution

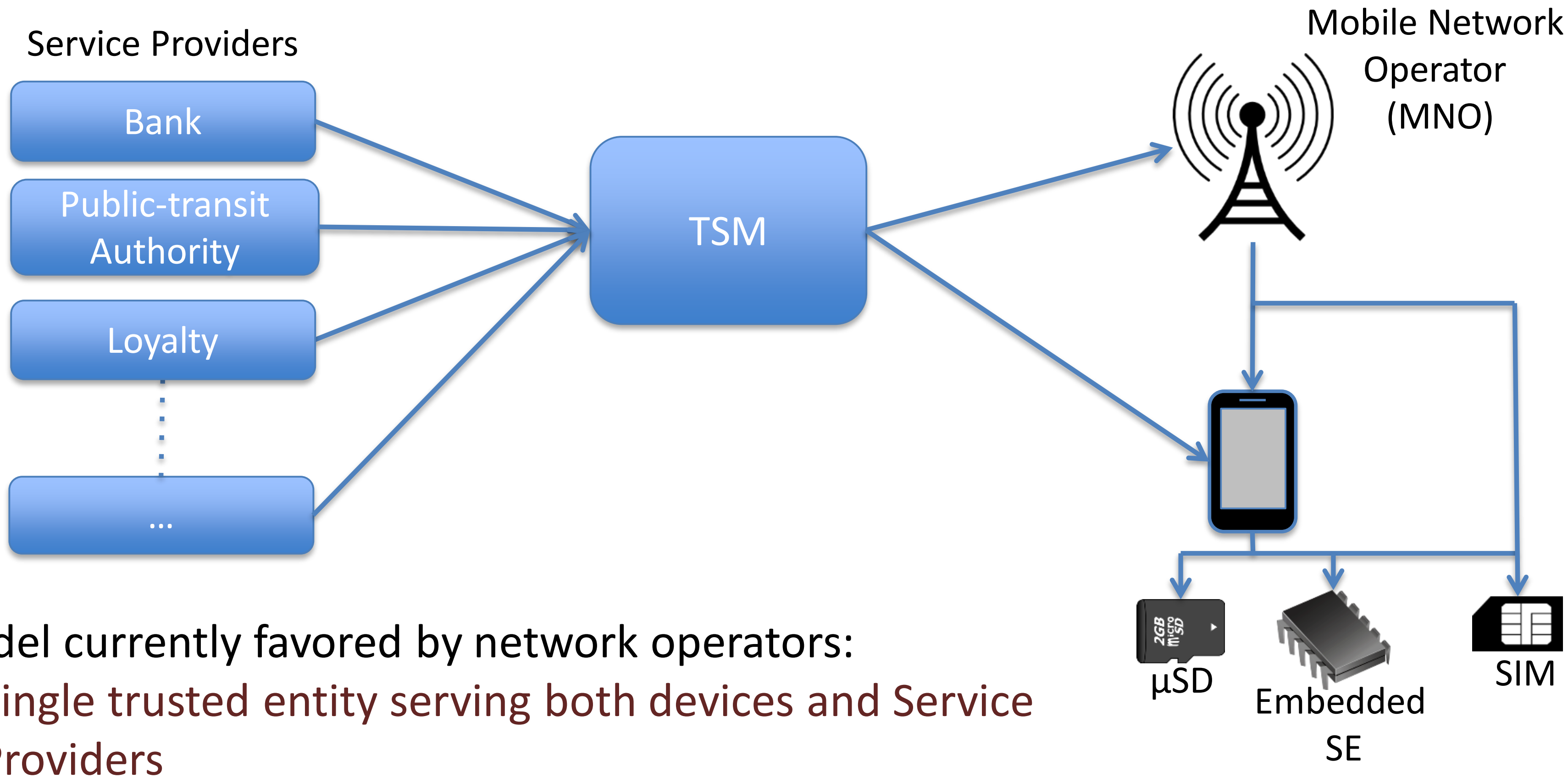
- Contactless stickers
  - Independent of the mobile phone OS
    - Elisa Lompakko (Earlier version)
- Universal Integrated Circuit Card (UICC)
  - Preferred by Mobile Network Operators
    - Orange Quick Tap (2011 - 2013)
- Secure MicroSD
  - Used by some banks in Taiwan
- Embedded Secure Element
  - Google Wallet
- Programmable Trusted Execution Environment
  - Kinibi
  - OPTEE
  - On-board Credential (ObC)

Security Element Architecture, e.g. SIM



- SIM is multi-application smart card
- Each service provider creates a separate Security Domain on the card
- Problems: increases the complexity of card manager; over-the-air installation of new applets is challenging

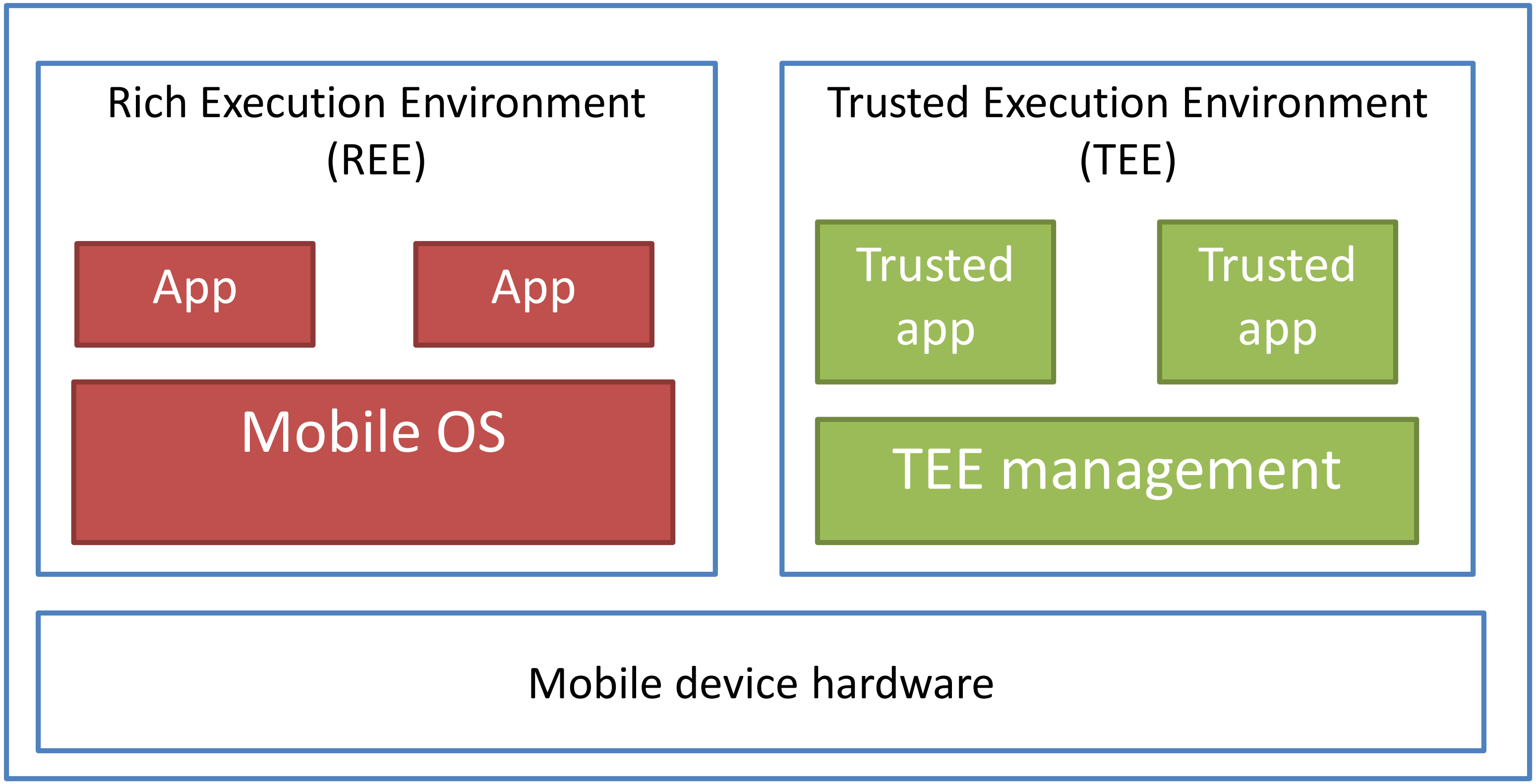
Trusted Service Manager (TSM)



Model currently favored by network operators:

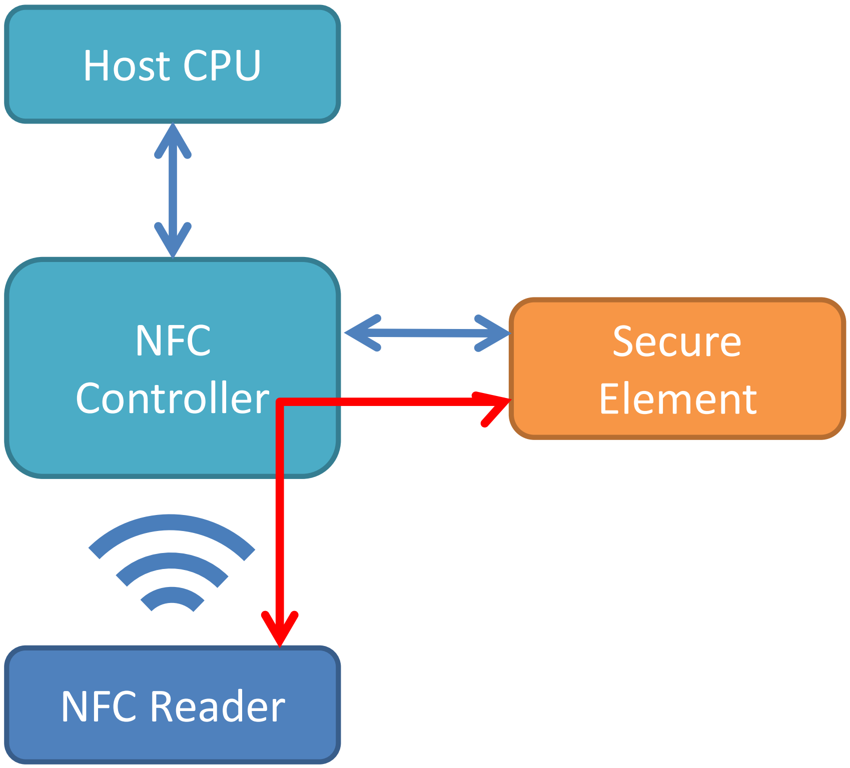
- Single trusted entity serving both devices and Service Providers
- Securely distributes and personalize the SP application to the customer's SE over the air (OTA personalization).
- Verify user's device and SE capabilities and resources
- Manage life cycle of the applications

TEE Architecture

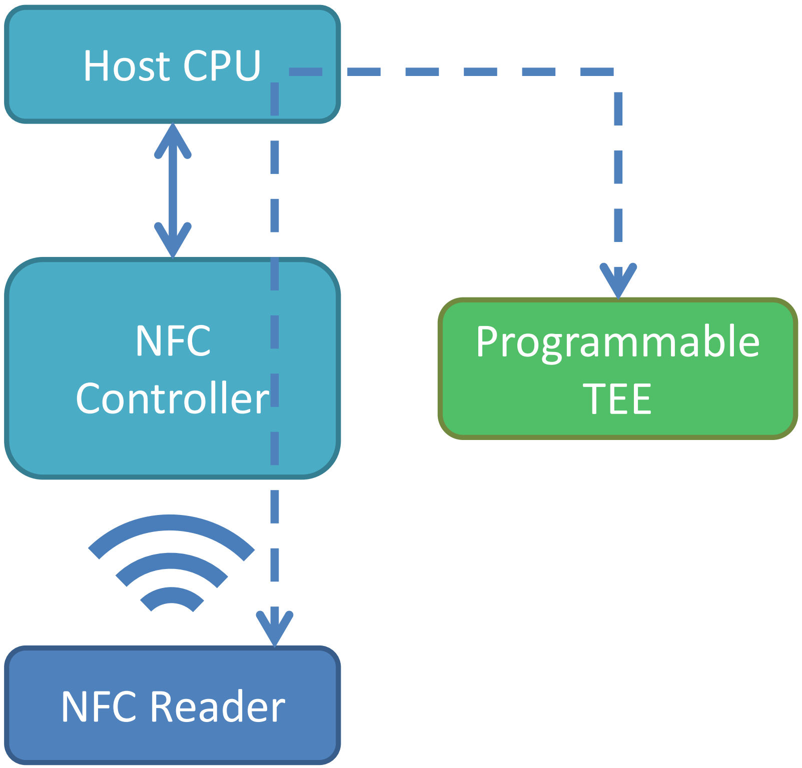


- Executes trusted apps in isolation using Hardware-enforced isolation
- Secure storage
- Protects confidentiality and integrity of a Trusted apps runtime states

Host Card Emulation



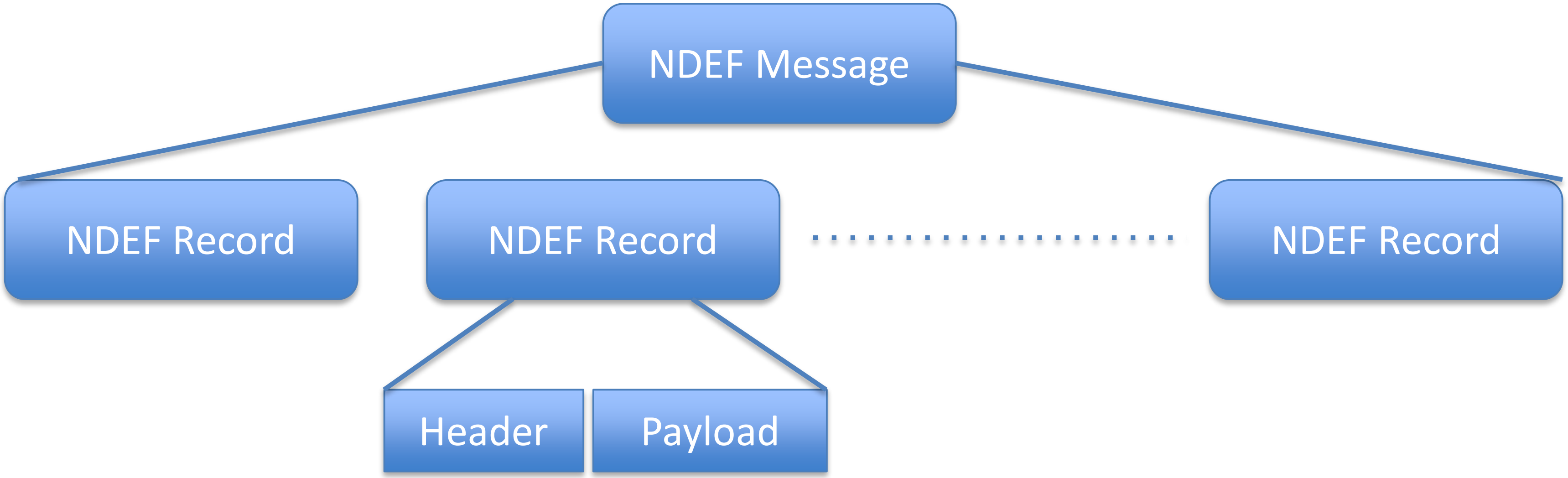
NFC card emulation with a secure element



NFC card emulation with a Programmable TEE

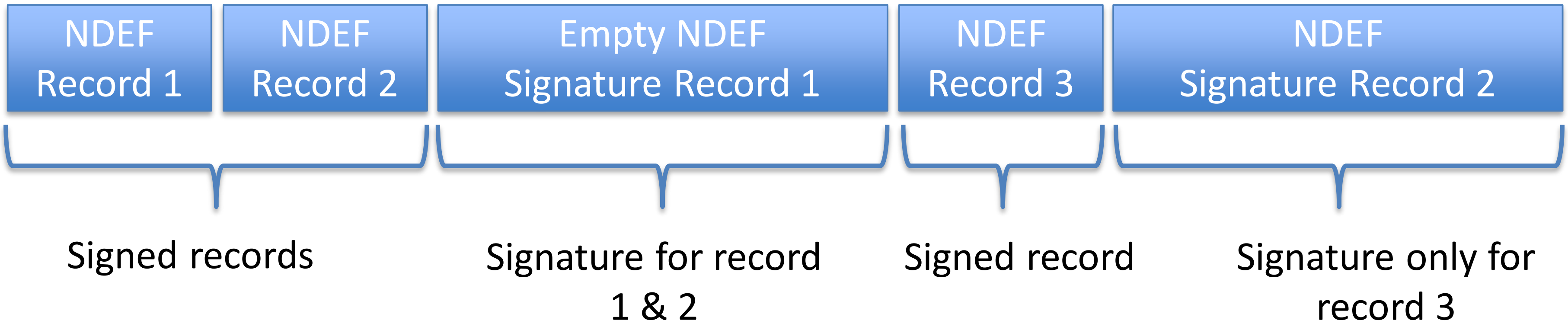
NFC applications

NFC Data Exchange Format (NDEF)



- NDEF is Message encapsulation format.
- Used to exchange messages between:
  - NFC devices or
  - An NFC device and a tag.
- Contains one or more NFC application data as NDEF Record
- Header defines the properties of the Payload.
  - Start and end of NDEF records
  - Record type definition (RTD): payload data type
  - Length of the payload etc.

Signature Record Type Definition



NDEF Record 1

NDEF Record 2

Empty NDEF Signature Record 1

NDEF Record 3

NDEF Signature Record 2

Signed records

Signature for record 1 & 2

Signed record

Signature only for record 3

- Provides integrity and authenticity
- Signature RTD contains:
  - Signature,
    - RSA (1024) with SHA-1 and PKCS#1 v 1.5 padding or PSS
    - ECDSA (P-192) with SHA-1 with no padding.
  - Certificate chain.
  - Or, reference location to the signature
- Signature Record apply for
  - all preceding records, (from record 1) or,
  - Record following the preceding Signature record.
- Vulnerability
  - Cloning, replacing a tag with another valid tag

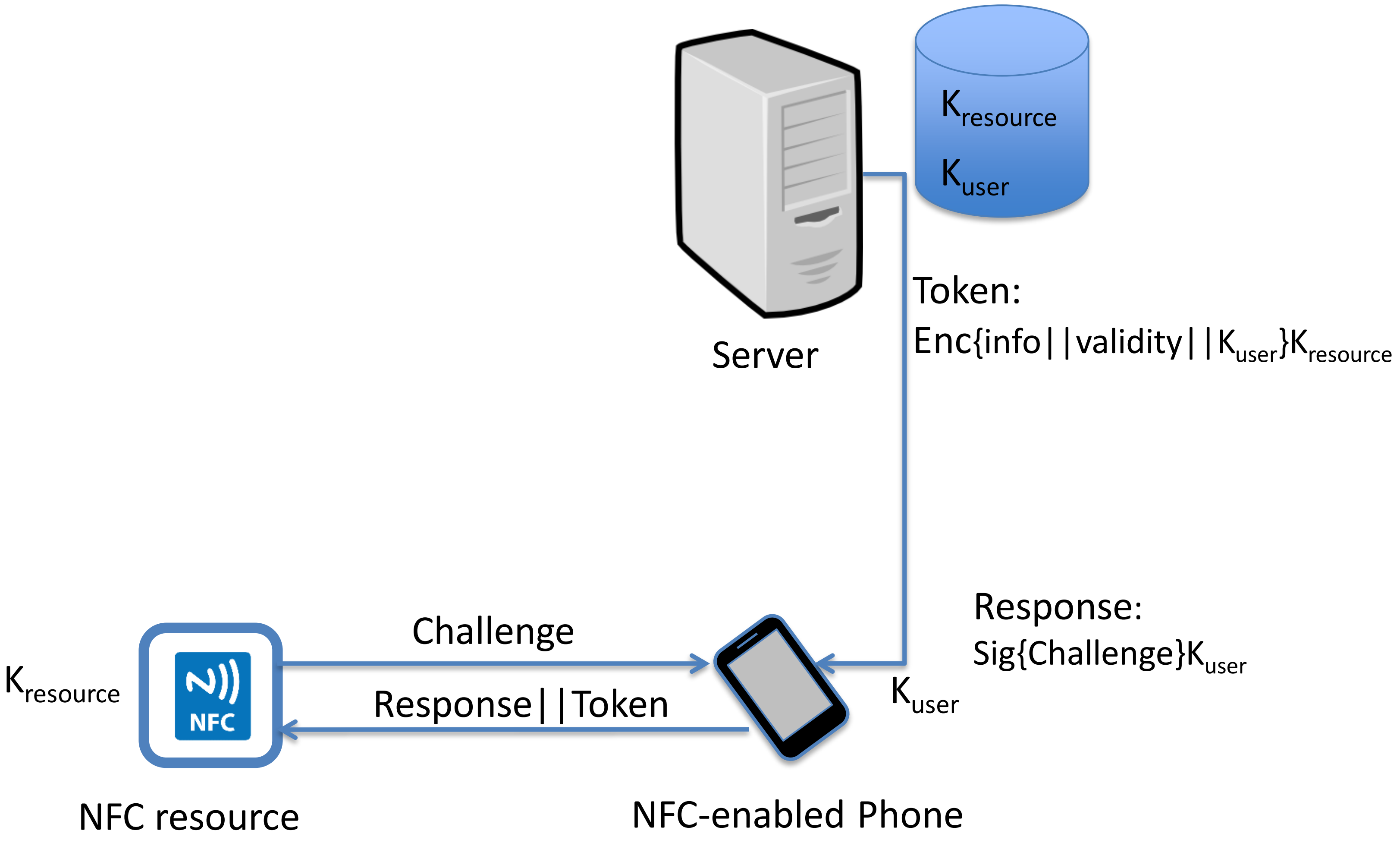
Example 1. Tag UID based NFC applications

- Simple Access control application based on tag UID
  - NFC tags is used as credential to identify the user
  - Reader must be connected to a backend database
  - Backend server maintains access policies
- Pros:
  - Simple and cheap solution
    - "UID cannot be faked easily !!"
  - Suitable for small scale business
- Cons:
  - Backend complexity increases with the number of customers
  - Readers needs to be connected to the backend server all the time.

Example 2: Event Ticketing

- One time or limited use tags
  - MIFARE Ultralight / Ultralight C
- Reader implements cryptographic functionality
  - Key diversification – e.g. Hash(UID + Master key)
  - Encrypts data and store the cipher-text on tags.
  - Reads the cipher-text from tags and decrypts data.
- Use of OTP bytes as incremental counter
- Use Lock bytes to prevent rewrite
- MAC for integrity protection
- Authentication keys for access control

Example 3. Access Control for Buildings



$K_{resource}$

K_{user}

Server

Token:

Enc{info||validity||K_{user}}$K_{resource}$

Response:

Sig{Challenge}K_{user}

K_{user}

Challenge

Response||Token

$K_{resource}$

NFC resource

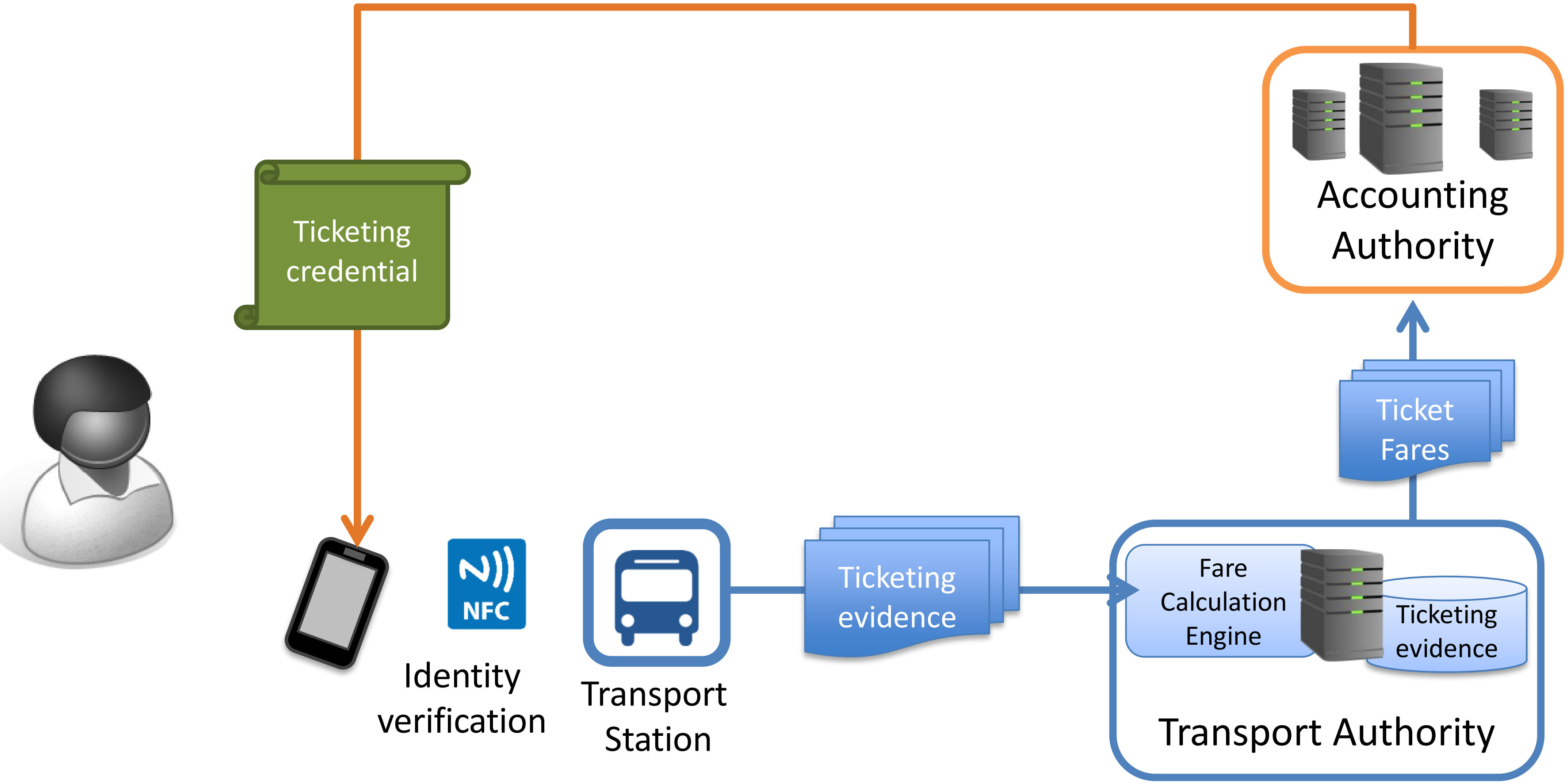
NFC-enabled Phone

Dmitrienko, Alexandra, et al. "SmartTokens: Delegable Access Control with NFC-Enabled Smartphones." Trust. 2012.

Example 4: Public transit application

- Proprietary solutions are widely used
  - MIFARE Classic
  - MIFARE Ultralight/Ultralight C
  - MIFARE DESFire EV1
  - Uses Symmetric crypto
    - Triple-DES, AES
  - Value is stored on the card
- Standards
  - ITSO: Interoperable public transport ticketing using contactless smart customer media.
  - Open Ticketing Institute: Account-based ticketing.

Account-based ticketing

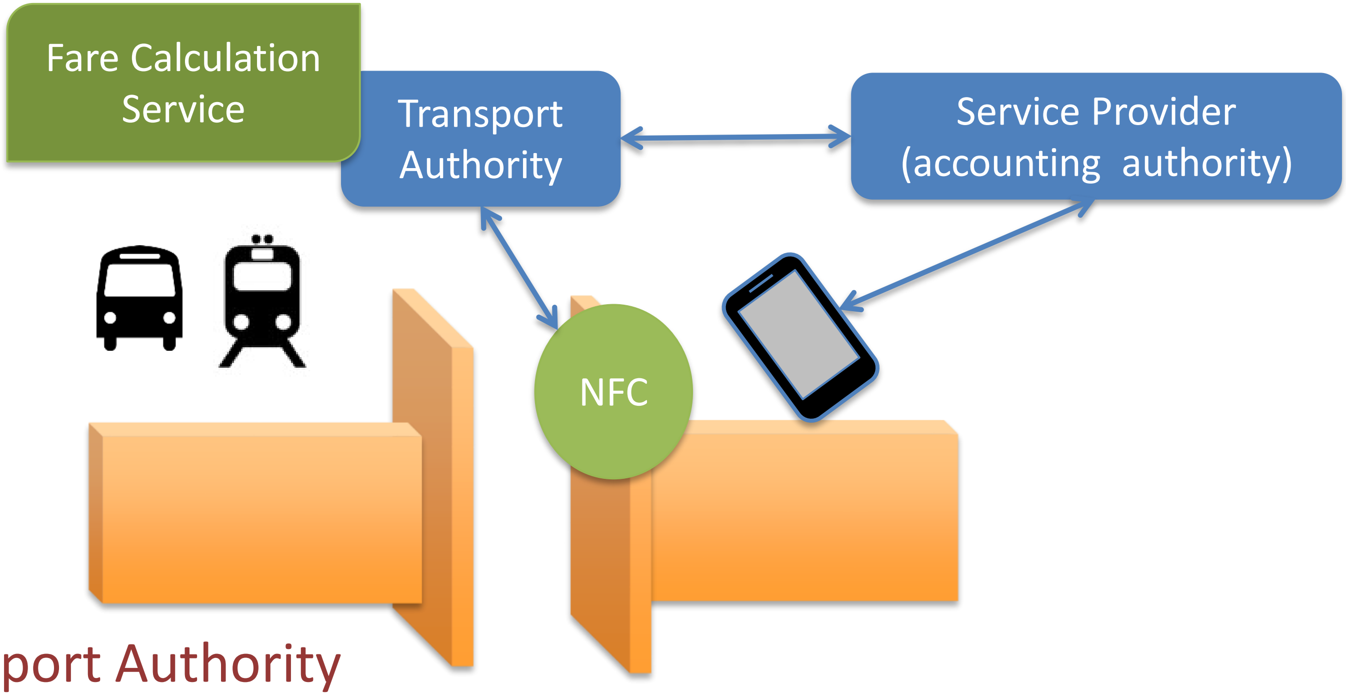


- Travel account in the cloud
- Identity verification at the transport station

Account-based ticketing

- Each traveler has a travel account in a cloud, which is operated by a service provider (SP)
- User device (travel card / NFC phones) only stores user's identity and credentials
  1. User identity is verified by a reader at the station gate
  2. Ticket identity and travel information sent to a backend server
  3. The backend server calculates the ticket fare and forwards the information to SP for payment collection
  4. Payment is collected by SP
- Allows credentials from different SPs to be used
  - E.g. Bank cards, SIM card, National ID etc.

Account-based Ticketing with Mobile Phone



- Transport Authority
  - Operates transport system
  - Calculate the fare calculation based on the ID and traveling distance
  - Collects ticketing evidence for auditing
- Service provider (e.g. bank or mobile operator)
  - Manages the customer relationship and travel account; issues the travel credentials
  - Collects evidence directly from phones and from Transport Authority for auditing
  - Collets payment from the customer (prepaid or credit)

Additional reading

- NFC Data Exchange Format (NDEF) Technical Specification
- Madlmayr, G.; Langer, J.; Kantner, C.; Scharinger, J.; , "NFC Devices: Security and Privacy," *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on* , vol., no., pp.642-647, 4-7 March 2008
  doi: 10.1109/ARES.2008.105
- L. Francis, G. P. Hancke, K. E. Mayes, and K. Markantonakis. Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones. Cryptology ePrint Archive, Report 2011/618, 2011. http://eprint.iacr.org/2011/618.