NFC Mobile Wallets: Where Trust Breaks Down

Max Sobell
Intrepidus Group
An NCC Group Company

AthCon, 2013

How this talk will progress

- Credit Card security features
  - Mag stripe -> Contactless -> Mobile Wallets
- Secure Element (SE)
  - What is this magical thing?
- Operating system protections (Android)
  - And how to get around them
- APDUs and communication with the SE
  - Once we can talk to it, what can we ask?
- Wallet mistakes
  - Derp

What this talk DOES NOT CONTAIN

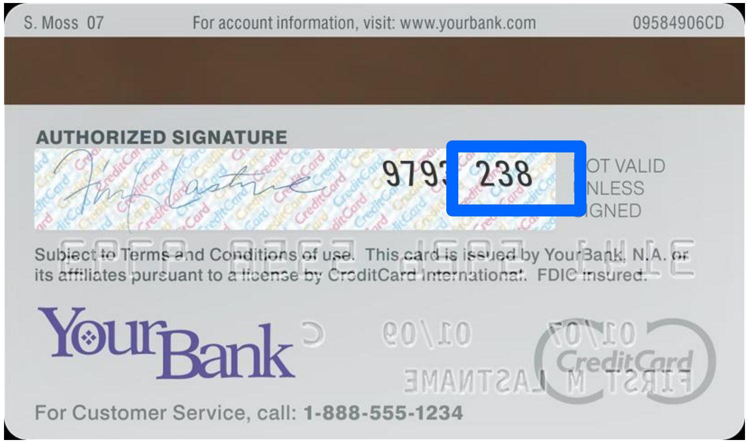
- Sweet 0days
- All of the credit card numbers
- Kiddie scriptz
- Secrets

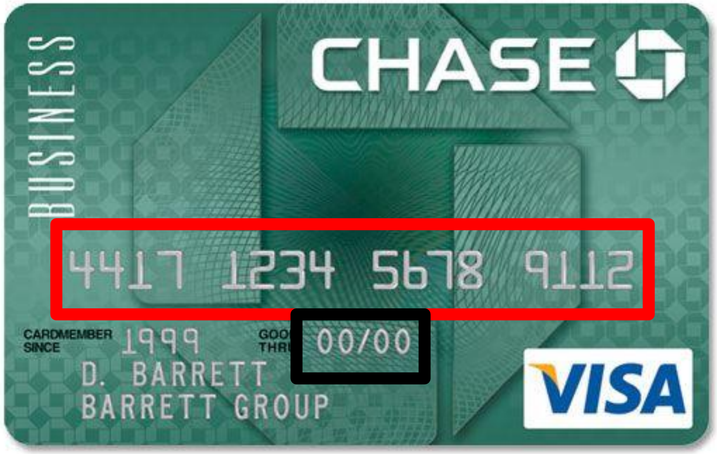
Myth vs Fact

- Myth: you can read NFC from long range with a large antenna!
  - You will probably burn a hole in someone’s pocket first. Disclaimer: not an antenna expert.
- Myth: you can eavesdrop on mobile wallet transactions from 20 meters!
  - This is extremely difficult, if not impossible. Also, what’s the point?

What’s in a credit card?

- I *knew* there would be a use for https://twitter.com/NeedADebitCard
- **PAN (Primary Account Number)**
- **CVV (MC) or CVC (Visa) or CID (Amex)**
- **Expiration**


S. Moss 07
For account information, visit: www.yourbank.com
09584906CD
AUTHORIZED SIGNATURE
9793 238
OT VALID
NLESS
GNED
Subject to Terms and Conditions of use. This card is issued by YourBank, N.A. or its affiliates pursuant to a license by CreditCard International. FDIC insured.
YourBank
CreditCard
For Customer Service, call: 1-888-555-1234



Where's the security

- CVV/CVC/CID (let's just call them all CSC for Card Security Code)
  - CSC1: Encoded in the mag strip
  - CSC2: Printed onto the actual card for CNP (card not present) transactions
- Additional verification
  - Name, Address, Expiration, and/or Billing Zipcode
- Primary Account number
  - Credit card thieves HAALP!

So… how secure are they?

- Restaurant scenario
  - U.S.: Waiter takes your card in back…
  - You sign the receipt
- Authorization for **unlimited** transactions
- All static data!

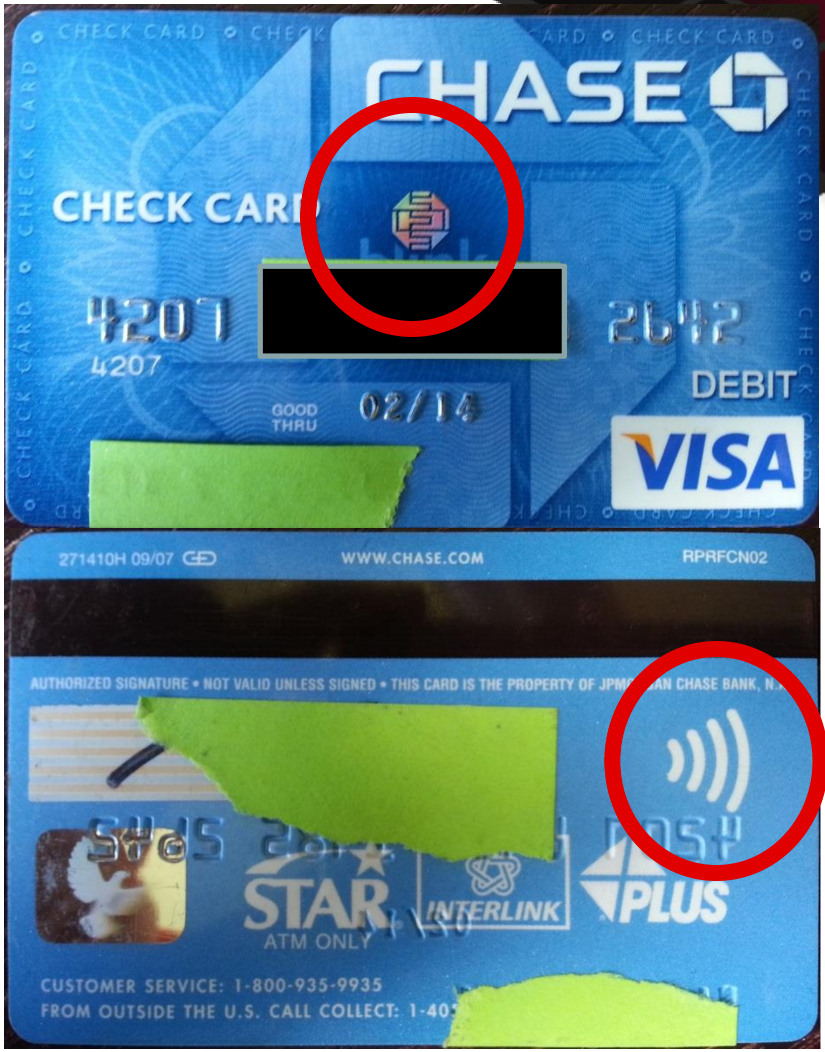
So mag stripe card cloning is easy…

… And we’re generally fine with that

How about my contactless card?

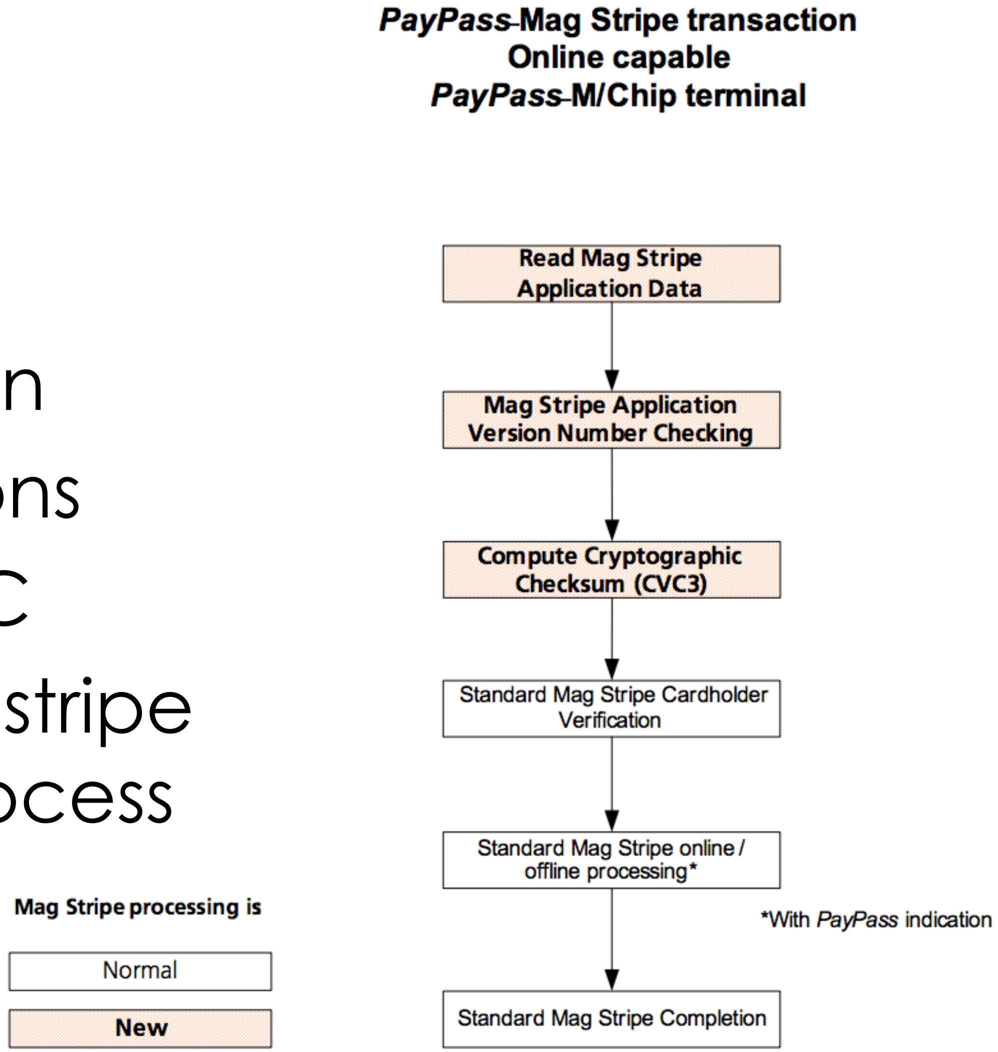
Contactless Card Overview

- Contactless cards are *sometimes* labeled
- Same PAN/CSC/Additional info protections
- **How about contactless protections?**


CHASE
CHECK CARD
4207 [redacted] 2642
4207
GOOD THRU 02/14
DEBIT
VISA
271410H 09/07
WWW.CHASE.COM
RPRFCN02
AUTHORIZED SIGNATURE • NOT VALID UNLESS SIGNED • THIS CARD IS THE PROPERTY OF JPMORGAN CHASE BANK, N.A.
STAR
ATM ONLY
INTERLINK
PLUS
CUSTOMER SERVICE: 1-800-935-9935
FROM OUTSIDE THE U.S. CALL COLLECT: 1-40

Transaction flow: CSC3

- CSC3
- From PayPass (Mastercard) documentation
- Bolt-on additions
- Calculate CVC
- Standard magstripe verification process



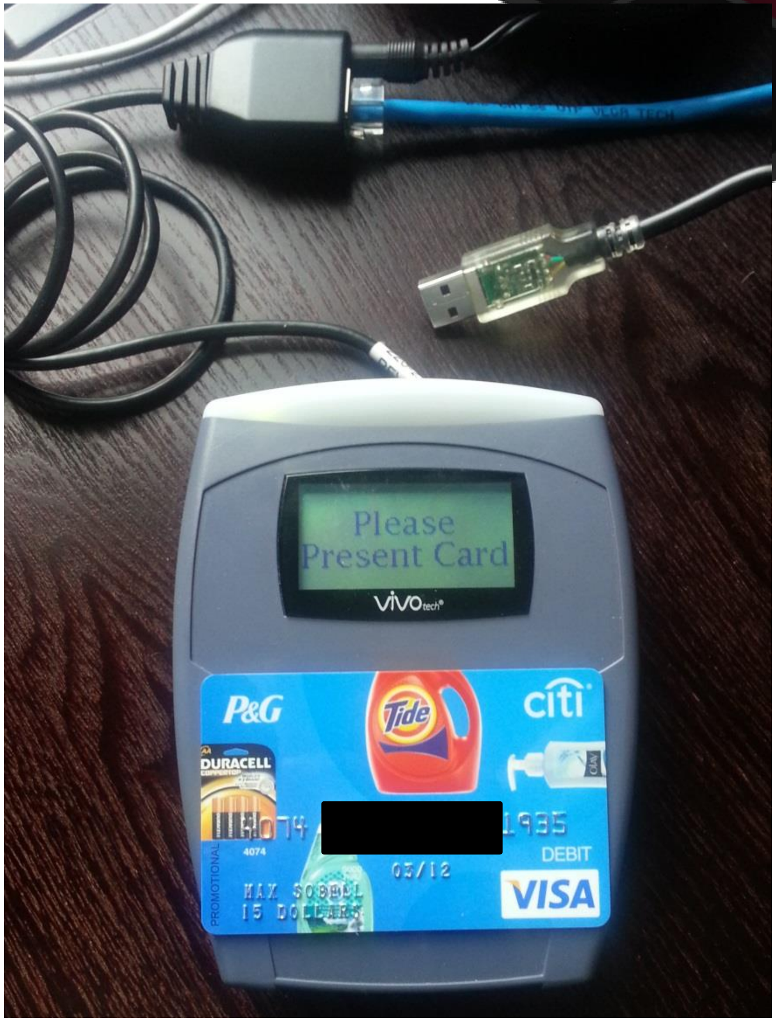
***PayPass*-Mag Stripe transaction**
Online capable
***PayPass*-M/Chip terminal**

Read Mag Stripe Application Data
Mag Stripe Application Version Number Checking
Compute Cryptographic Checksum (CVC3)
Standard Mag Stripe Cardholder Verification
Standard Mag Stripe online / offline processing*
Standard Mag Stripe Completion
Mag Stripe processing is
Normal
New

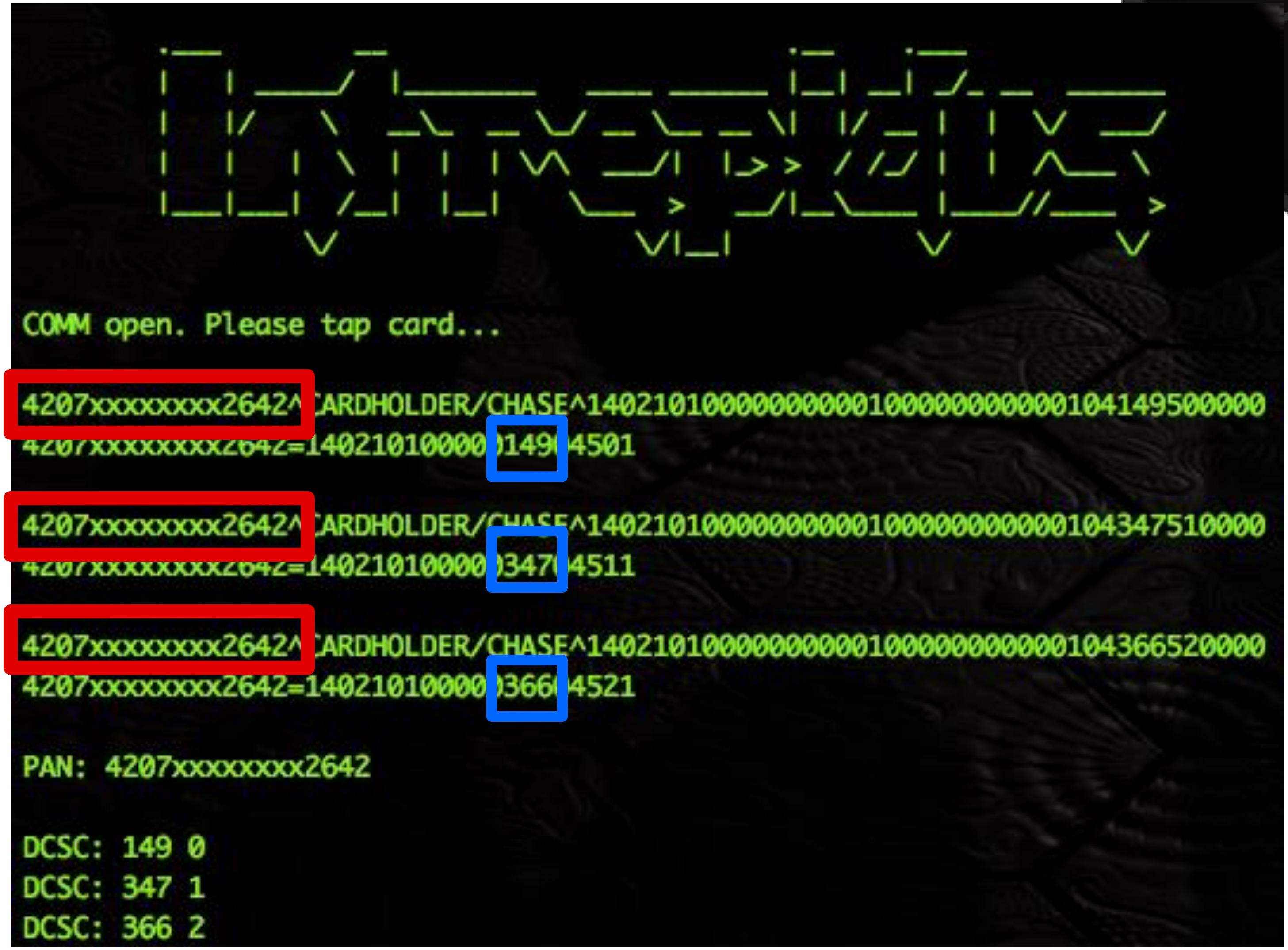
*With *PayPass* indication

Setup

- Point of Sales system (PoS)
- Jerry-rigged to a serial to USB cable
- Outputs serial data
- Script to make it pretty



DCSC in action

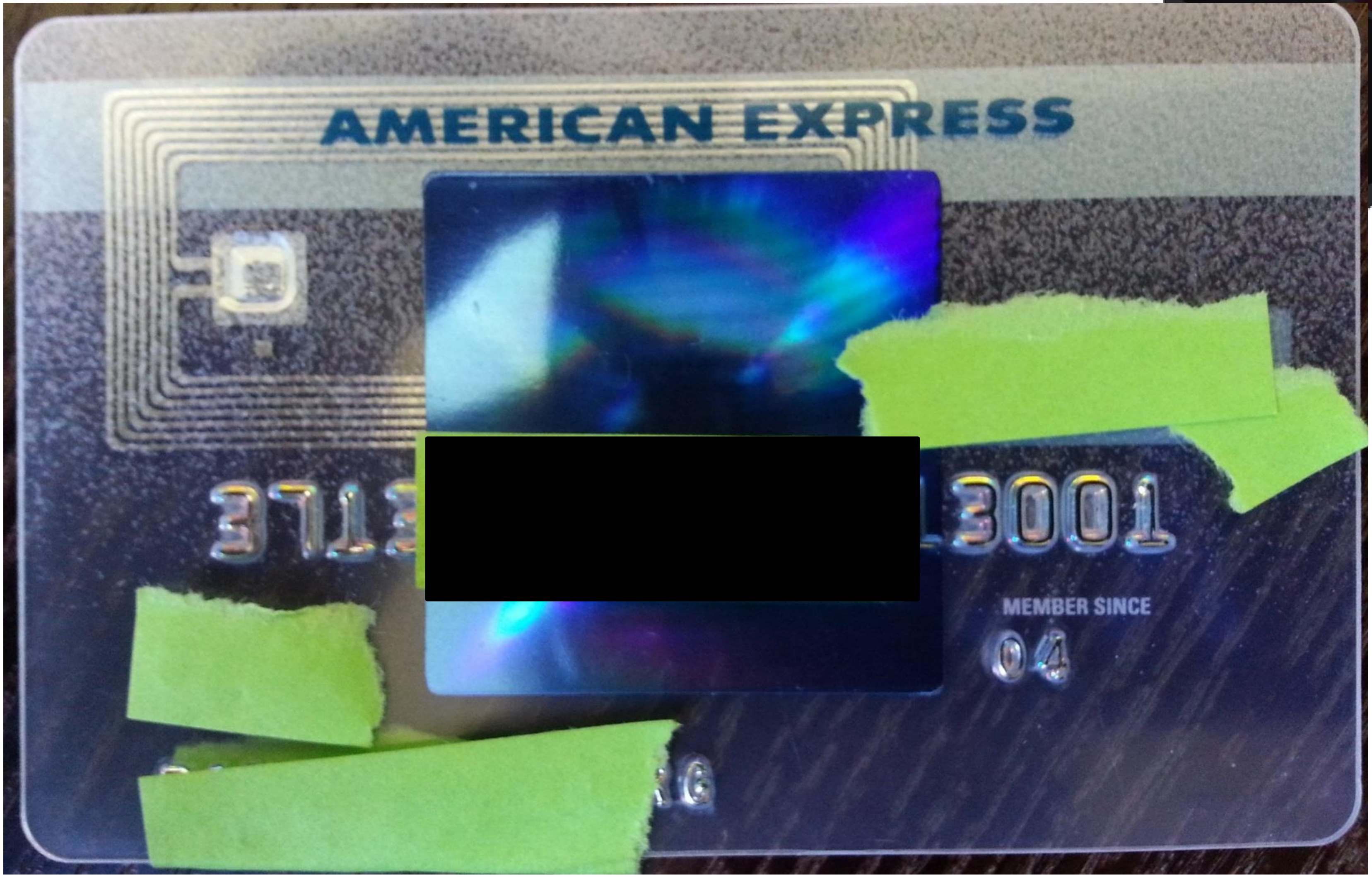


```
COMM open. Please tap card...

4207xxxxxxxx2642^CARDHOLDER/CHASE^1402101000000000000100000000000104149500000
4207xxxxxxxx2642=14021010000014904501

4207xxxxxxxx2642^CARDHOLDER/CHASE^1402101000000000000100000000000104347510000
4207xxxxxxxx2642=14021010000034714511

4207xxxxxxxx2642^CARDHOLDER/CHASE^1402101000000000000100000000000104366520000
4207xxxxxxxx2642=14021010000036624521

PAN: 4207xxxxxxxx2642

DCSC: 149 0
DCSC: 347 1
DCSC: 366 2
```

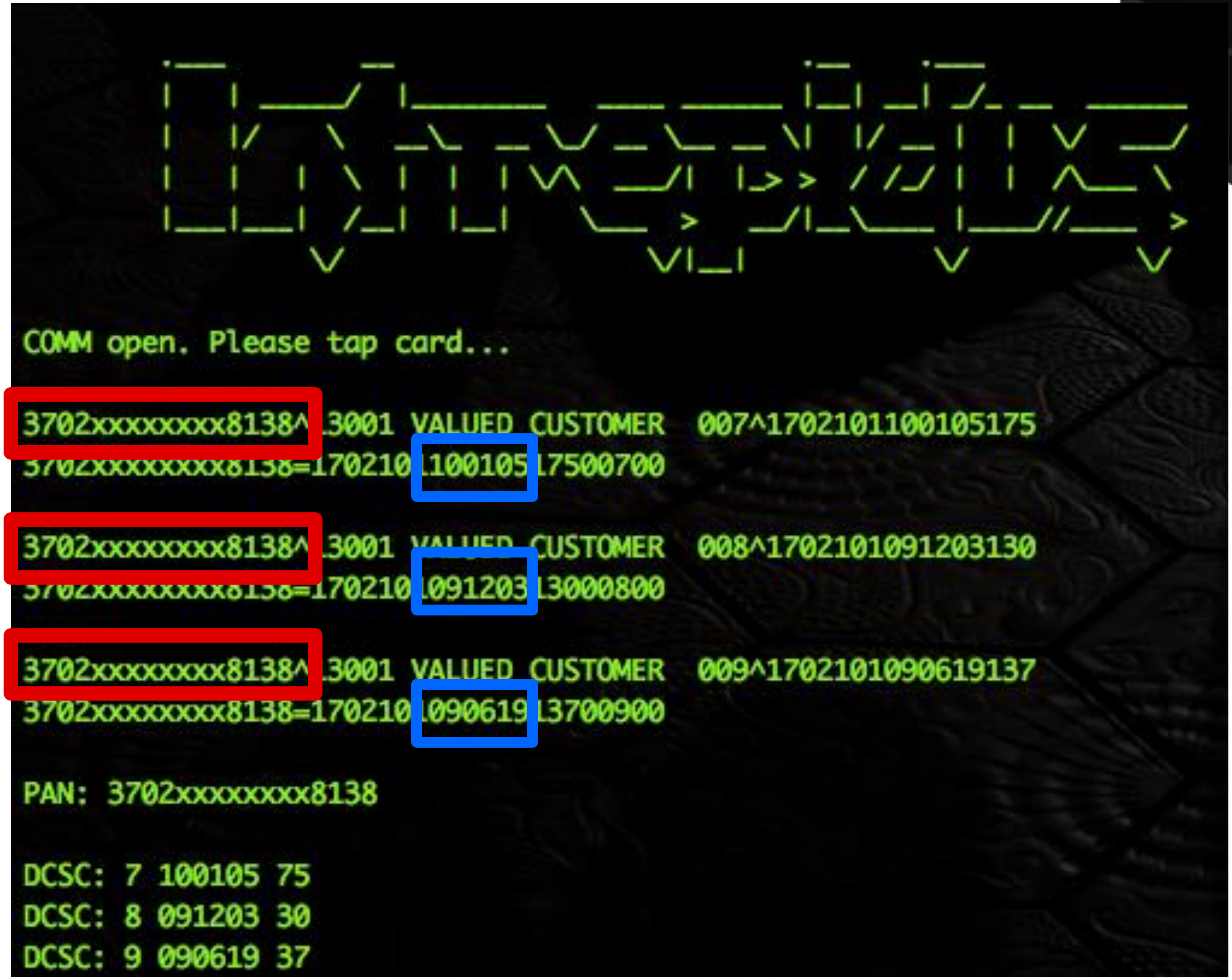
How did they do?

- PAN is the same
- DCSC is present
  - Very good protection
  - Gives one authorization at a time
  - Must be in order!
- Additional protections?
  - Can we get a different PAN?
  - Contactless PAN v standard PAN

Can we get a different PAN?



Can we get a different PAN?



```
COMM open. Please tap card...

3702xxxxxxx8138^3001 VALUED CUSTOMER   007^1702101100105175
3702xxxxxxx8138=170210110010517500700

3702xxxxxxx8138^3001 VALUED CUSTOMER   008^1702101091203130
3702xxxxxxx8138=170210109120313000800

3702xxxxxxx8138^3001 VALUED CUSTOMER   009^1702101090619137
3702xxxxxxx8138=170210109061913700900

PAN: 3702xxxxxxx8138

DCSC: 7 100105 75
DCSC: 8 091203 30
DCSC: 9 090619 37
```

How did they do?

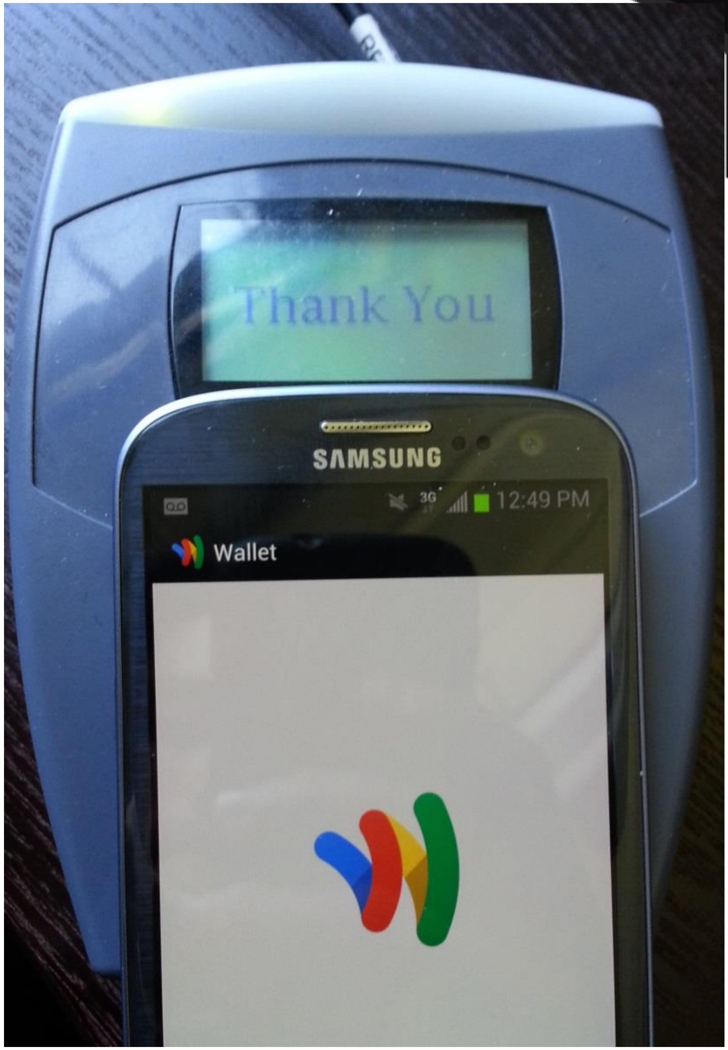
- PAN is **different**
- DCSC is present
- All additional info for back-end processing
- Helps cc processors (MC, Visa, Amex) decide which transactions are fraudulent

So…?

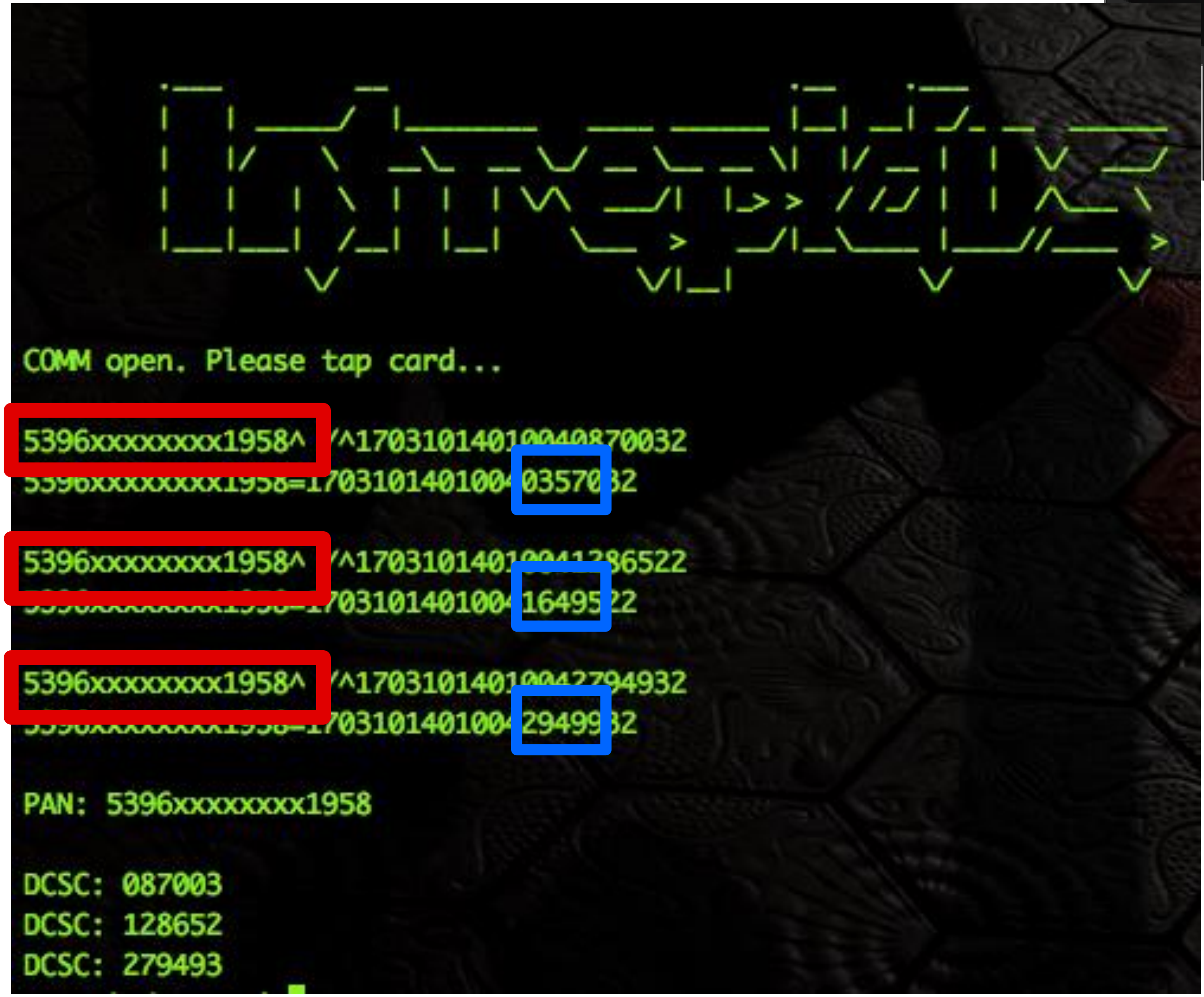
- How does this relate to mobile wallets?
  - Contactless credit card == mobile wallet to the Point-of-Sale (PoS) terminal
  - See same data
    - AID
    - PAN
    - DCSC
    - “Track 2 data”
- Contactless v Mobile Wallet Security

Mobile wallet?

- Sprint Galaxy S III
- Google Wallet
- Visa card applet



Mobile taps



```
COMM open. Please tap card...

5396xxxxxxxx1958^ /^170310140100040870032
5396xxxxxxxx1958=17031014010040357032

5396xxxxxxxx1958^ /^170310140100041286522
5396xxxxxxxx1958=17031014010041649522

5396xxxxxxxx1958^ /^170310140100042794932
5396xxxxxxxx1958=17031014010042949932

PAN: 5396xxxxxxxx1958

DCSC: 087003
DCSC: 128652
DCSC: 279493
```

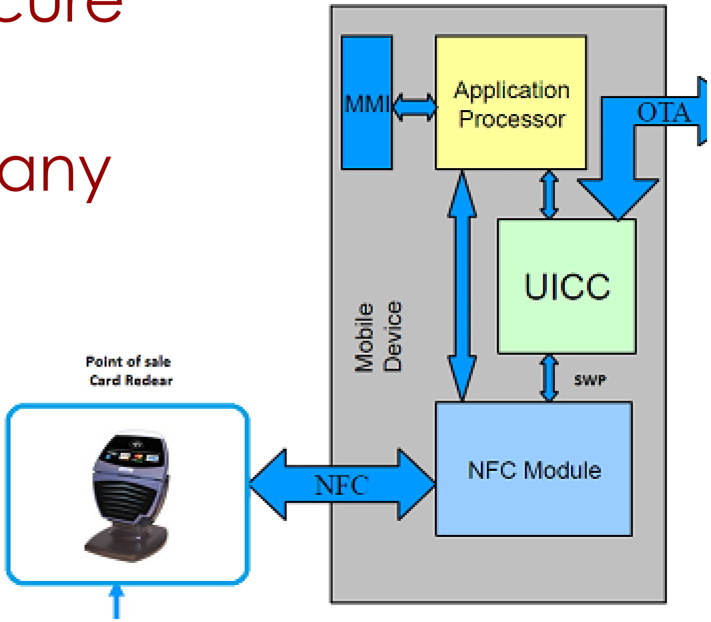
How did they do?

- All the protections we saw with the Amex
  - DCSC
  - Different PAN
- In a mobile device
- OS/device adds more protections!

One level deeper…

Architecture

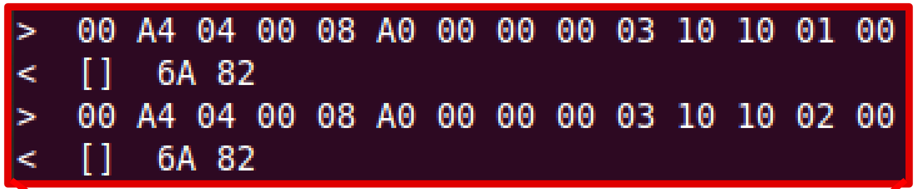
- So… my credit card is **in** the computer?
- Secure Element to the rescue!
  - Diagram shows a UICC Secure Element
  - Embedded SE built in to many devices



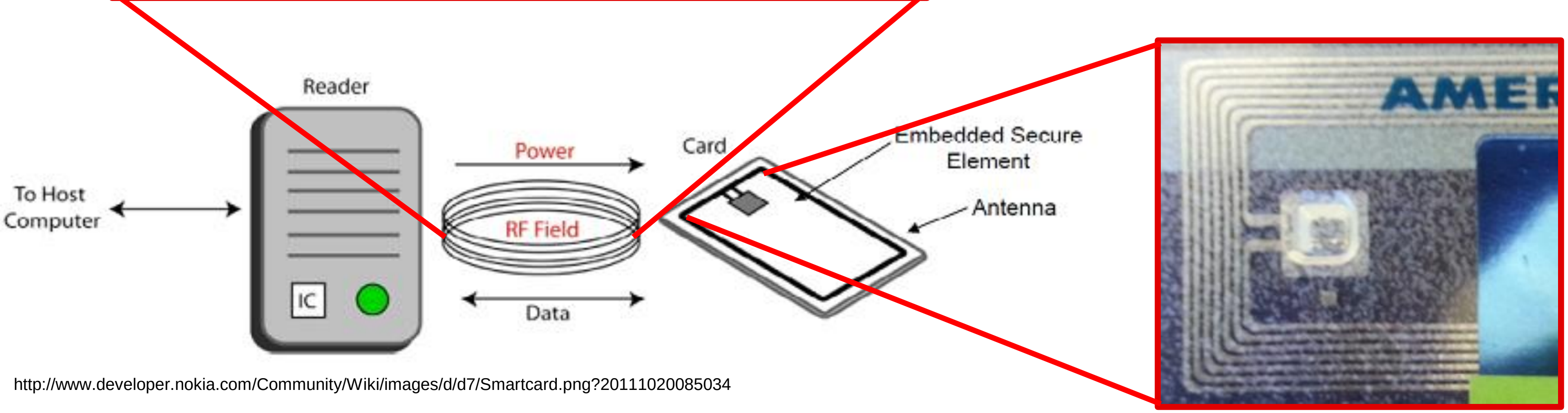
http://www.developer.nokia.com/Community/Wiki/images/e/ed/Echosystem.png?20111020085321

Communication breakdown

- Communication with SE via NFC
- APDU commands exchanged
  - Application Protocol Data Unit



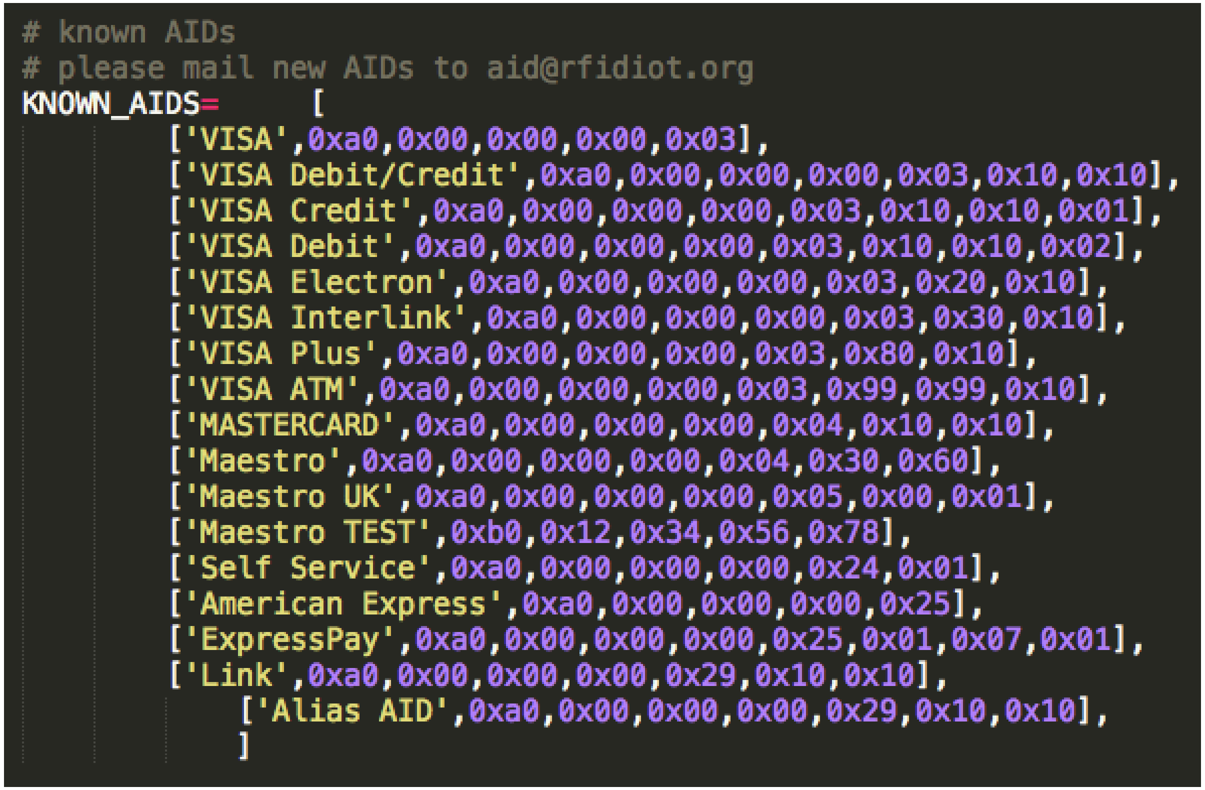
```
>  00 A4 04 00 08 A0 00 00 00 03 10 10 01 00
<  []  6A 82
>  00 A4 04 00 08 A0 00 00 00 03 10 10 02 00
<  []  6A 82
```



http://www.developer.nokia.com/Community/Wiki/images/d/d7/Smartcard.png?20111020085034

AID: Application IDentifier

- List of assigned AIDs
  - Wiki (http://en.wikipedia.org/wiki/EMV#Application_selection)
  - RFIDIOT
  - PoS/PPSE



```
# known AIDs
# please mail new AIDs to aid@rfidiot.org
KNOWN_AIDS=     [
        ['VISA',0xa0,0x00,0x00,0x00,0x03],
        ['VISA Debit/Credit',0xa0,0x00,0x00,0x00,0x03,0x10,0x10],
        ['VISA Credit',0xa0,0x00,0x00,0x00,0x03,0x10,0x10,0x01],
        ['VISA Debit',0xa0,0x00,0x00,0x00,0x03,0x10,0x10,0x02],
        ['VISA Electron',0xa0,0x00,0x00,0x00,0x03,0x20,0x10],
        ['VISA Interlink',0xa0,0x00,0x00,0x00,0x03,0x30,0x10],
        ['VISA Plus',0xa0,0x00,0x00,0x00,0x03,0x80,0x10],
        ['VISA ATM',0xa0,0x00,0x00,0x00,0x03,0x99,0x99,0x10],
        ['MASTERCARD',0xa0,0x00,0x00,0x00,0x04,0x10,0x10],
        ['Maestro',0xa0,0x00,0x00,0x00,0x04,0x30,0x60],
        ['Maestro UK',0xa0,0x00,0x00,0x00,0x05,0x00,0x01],
        ['Maestro TEST',0xb0,0x12,0x34,0x56,0x78],
        ['Self Service',0xa0,0x00,0x00,0x00,0x24,0x01],
        ['American Express',0xa0,0x00,0x00,0x00,0x25],
        ['ExpressPay',0xa0,0x00,0x00,0x00,0x25,0x01,0x07,0x01],
        ['Link',0xa0,0x00,0x00,0x00,0x29,0x10,0x10],
            ['Alias AID',0xa0,0x00,0x00,0x00,0x29,0x10,0x10],
            ]
```

Secure Element

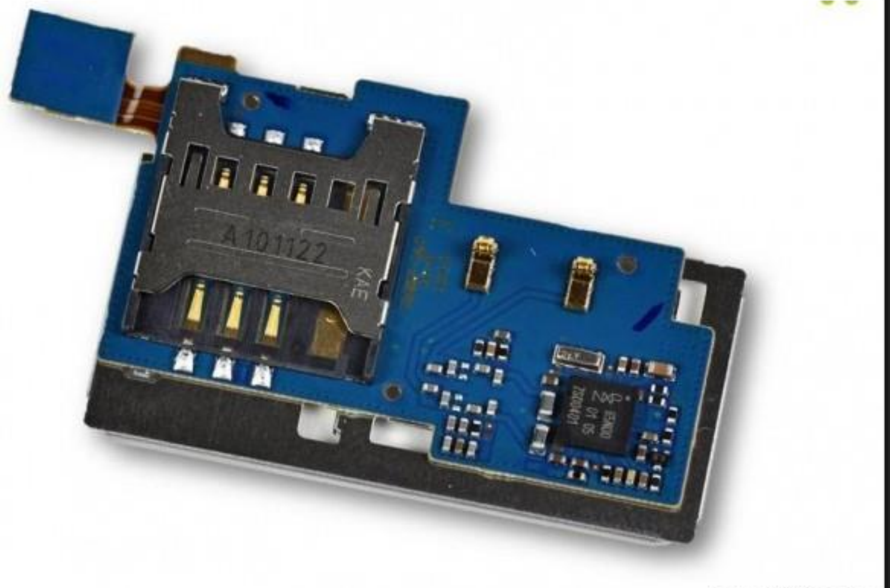
- SEs are not new!
- Tamper-resistant computer
  - Embedded
  - UICC (SIM)
- Runs JavaCard OS
  - JCOP
- Directory/File-based
  - AIDs/Records ← Your credit card!
  - Visa, Mastercard, etc.

Secure Element

- Communicates via APDUs
  - Application Protocol Data Unit
  - Contact v Contactless (Wired/Virtual)
  - ISO 7816-4
- SmartCard in a different package
  - Well-defined standards
  - JSR 177: Internal communication
    - Opens channel to application
    - Exchanges APDUs
    - Handles PIN functions

Mobile SE Form Factors

- Embedded SE
  - Embedded into NFC controller
  - Common: NXP PN 65
- UICC SE
  - Controlled by MNO
  - Shared space
  - Looks just like your standard SIM



Source: iFitIt Teardown

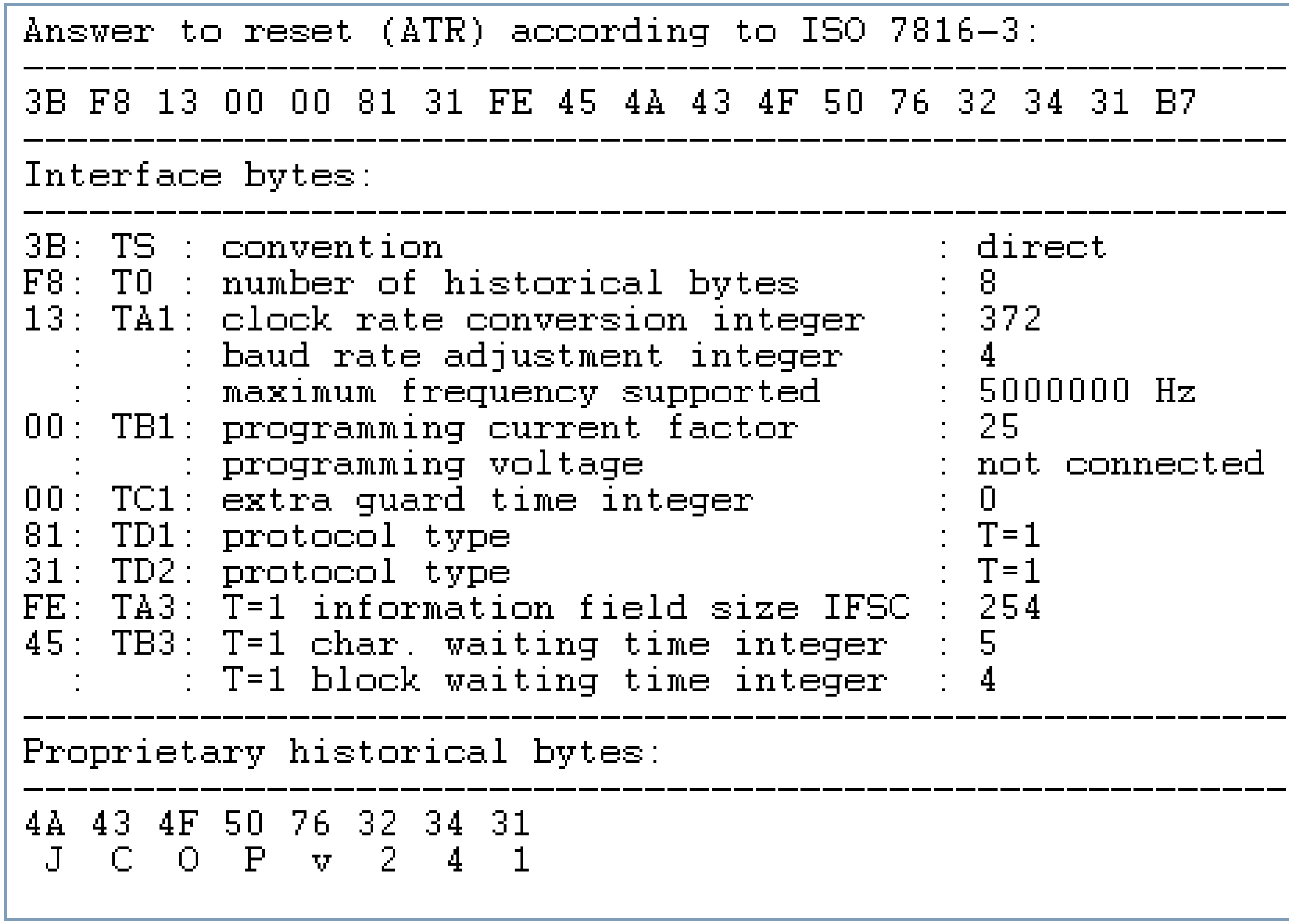
COMMUNICATION WITH THE SE

Communication with the SE

- UICC SE
- Outside of the device/smartcard:
  - Easy!
- Identive SCR3310 (and many many others)
  - USD $40
- Software
  - JLoad (Giesecke & Devrient)
  - JCShell (NXP)
  - Open Source GPShell (Global Platform)

Standard Smartcard

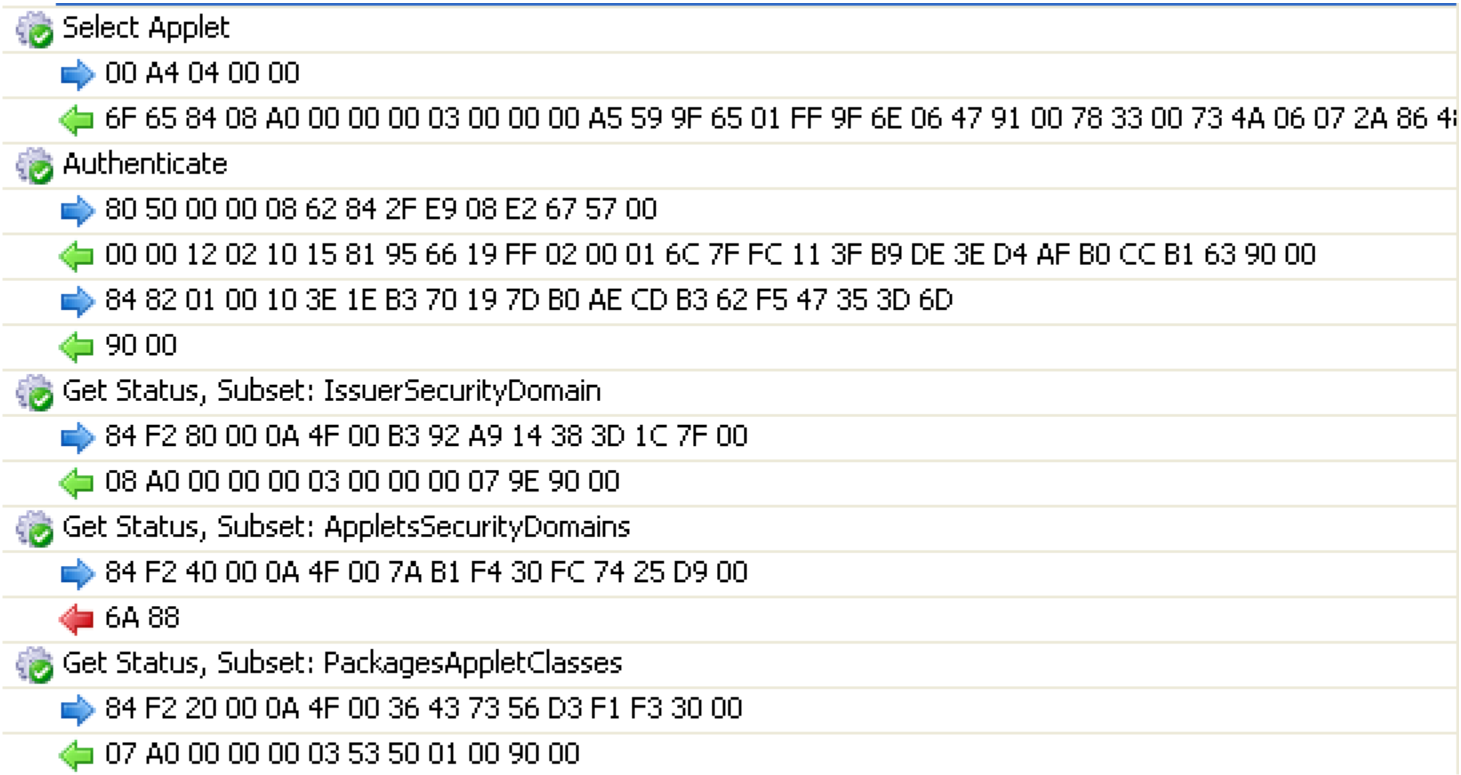
Smart Cards



```
Answer to reset (ATR) according to ISO 7816-3:
----------------------------------------------------------
3B F8 13 00 00 81 31 FE 45 4A 43 4F 50 76 32 34 31 B7
----------------------------------------------------------
Interface bytes:
----------------------------------------------------------
3B: TS : convention                          : direct
F8: T0 : number of historical bytes          : 8
13: TA1: clock rate conversion integer       : 372
  :    : baud rate adjustment integer        : 4
  :    : maximum frequency supported         : 5000000 Hz
00: TB1: programming current factor          : 25
  :    : programming voltage                 : not connected
00: TC1: extra guard time integer            : 0
81: TD1: protocol type                       : T=1
31: TD2: protocol type                       : T=1
FE: TA3: T=1 information field size IFSC     : 254
45: TB3: T=1 char. waiting time integer      : 5
  :    : T=1 block waiting time integer      : 4
----------------------------------------------------------
Proprietary historical bytes:
----------------------------------------------------------
4A 43 4F 50 76 32 34 31
 J  C  O  P  v  2  4  1
```

APDU Exchange

- Authenticate to ISD



```
Select Applet
  → 00 A4 04 00 00
  ← 6F 65 84 08 A0 00 00 00 03 00 00 00 A5 59 9F 65 01 FF 9F 6E 06 47 91 00 78 33 00 73 4A 06 07 2A 86 4
Authenticate
  → 80 50 00 00 08 62 84 2F E9 08 E2 67 57 00
  ← 00 00 12 02 10 15 81 95 66 19 FF 02 00 01 6C 7F FC 11 3F B9 DE 3E D4 AF B0 CC B1 63 90 00
  → 84 82 01 00 10 3E 1E B3 70 19 7D B0 AE CD B3 62 F5 47 35 3D 6D
  ← 90 00
Get Status, Subset: IssuerSecurityDomain
  → 84 F2 80 00 0A 4F 00 B3 92 A9 14 38 3D 1C 7F 00
  ← 08 A0 00 00 00 03 00 00 00 07 9E 90 00
Get Status, Subset: AppletsSecurityDomains
  → 84 F2 40 00 0A 4F 00 7A B1 F4 30 FC 74 25 D9 00
  ← 6A 88
Get Status, Subset: PackagesAppletClasses
  → 84 F2 20 00 0A 4F 00 36 43 73 56 D3 F1 F3 30 00
  ← 07 A0 00 00 00 03 53 50 01 00 90 00
```

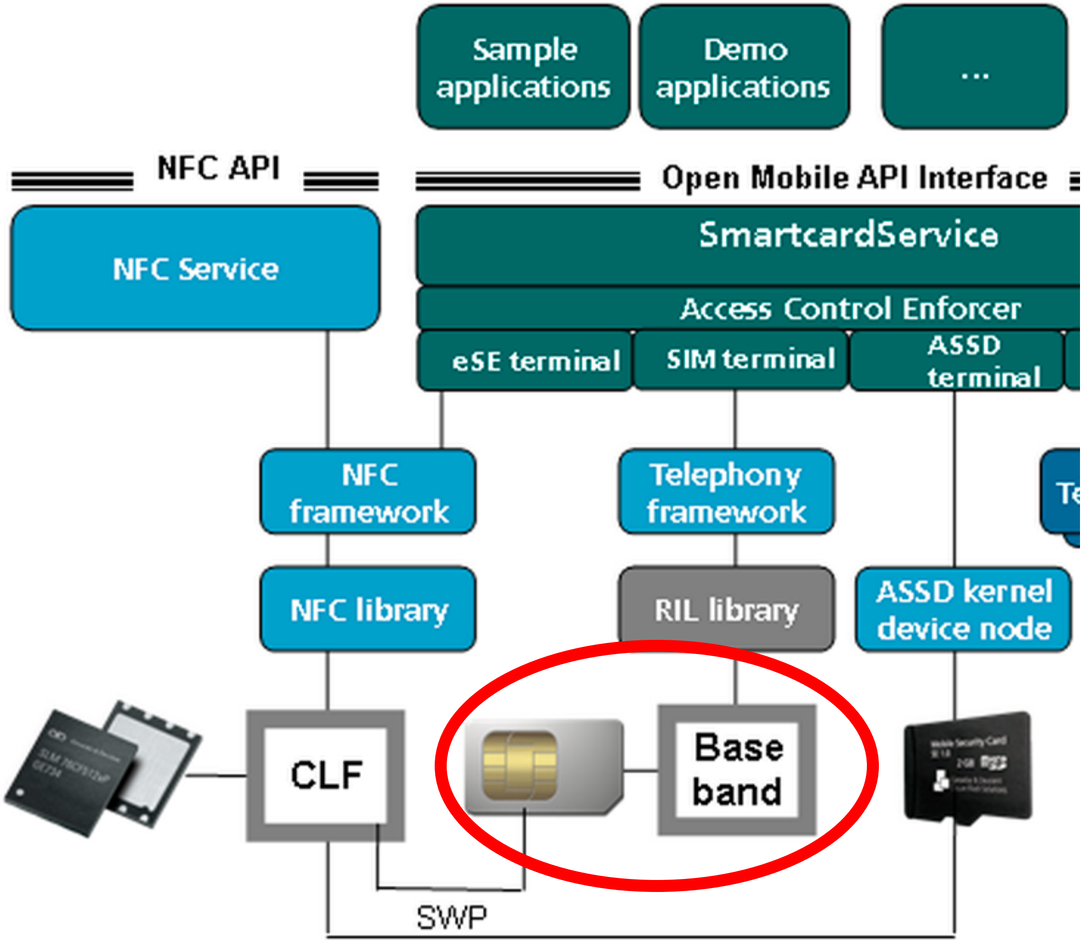
UICC SE

- UICC SE == SmartCard/Chip and PIN credit card
  - Same communication methods!

UICC SE

- How can we talk to the UICC SE from the OS?
  - UICC SE connected to baseband
  - Communication via Radio Interface Layer (RIL)
    - rild (open source) + OEM component

SEEK for Android



UICC SE

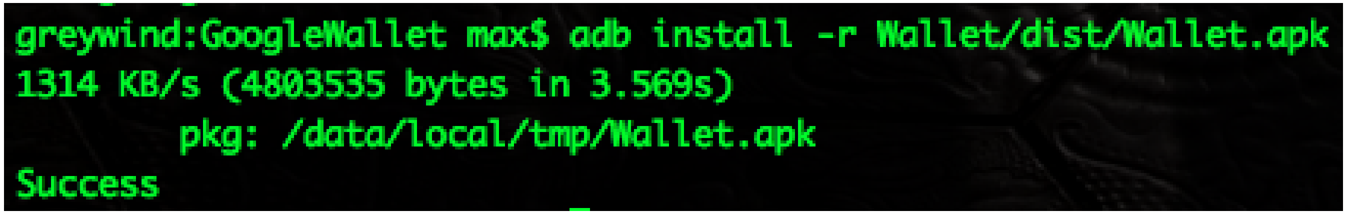
- SEEK for Android
  - Open source
  - SIMalliance Open Mobile API (SmartCard API) implementation
  - Provides patches for APDU commands via rild
  - Not required/used for Embedded SE
  - Should perform signature checks to prevent rogue application from communication with UICC SE

UICC SE

- How can we talk to the Embedded SE from the OS?
  - 2 wires: S2C SignalIn/SignalOut (NFC-Wired Interface (WI) standard)
  - 3 Modes
    - Off
    - Wired (talk to OS)
    - Virtual (talk to POS payment terminal)
  - Several OS protections
    - Need rooted device

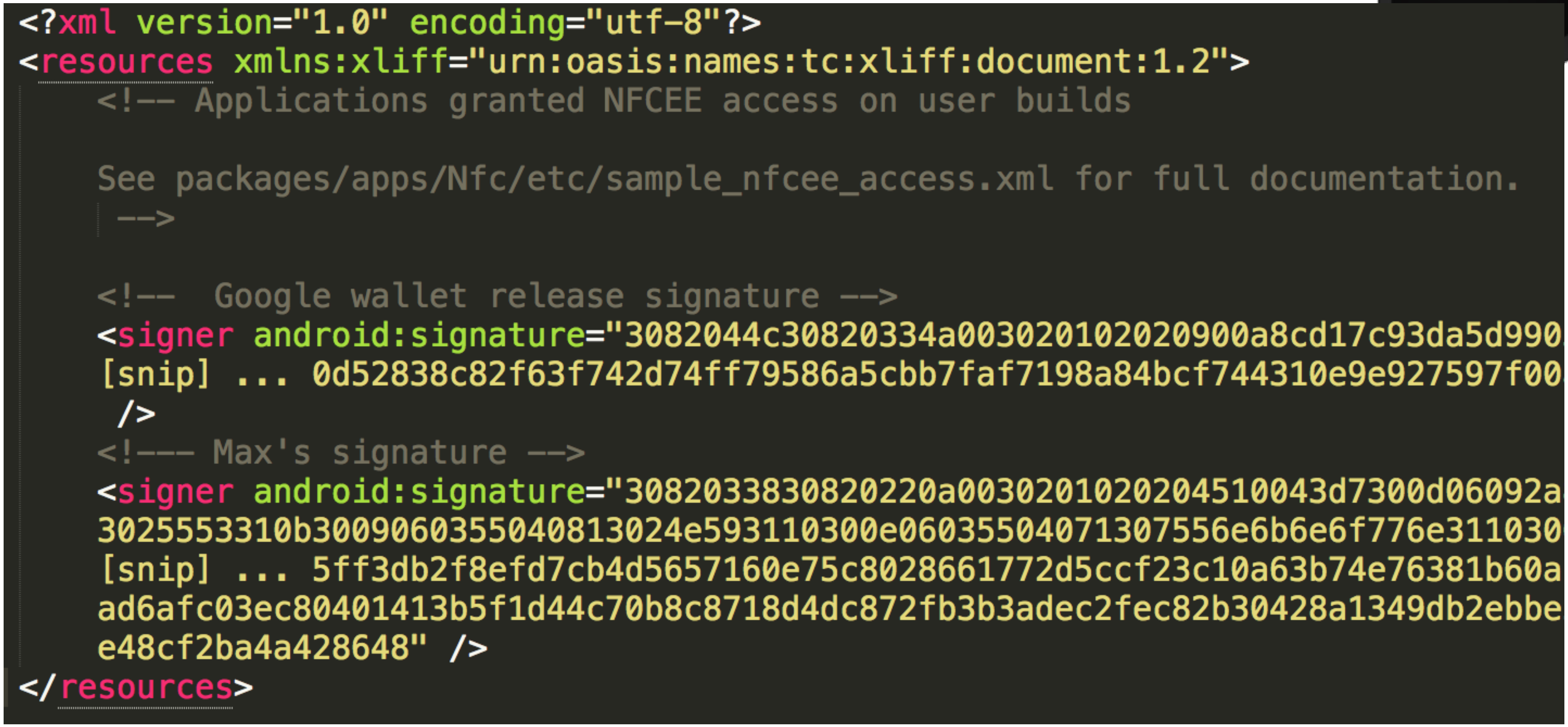
Android protections

- /system/etc/nfcee_access.xml
  - Add a new signer tag with your (public) key
  - APK requests android.permission.NFC
- (optional) To replace Google Wallet
  - /data/system/packages.xml
    - Replace cert for com.google.android.apps.walletnfcrel
    - Allows uninstall/reinstall to /system/app/Wallet.apk



```
greywind:GoogleWallet max$ adb install -r Wallet/dist/Wallet.apk
1314 KB/s (4803535 bytes in 3.569s)
        pkg: /data/local/tmp/Wallet.apk
Success
```

nfcee_access.xml



```
<?xml version="1.0" encoding="utf-8"?>
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">
    <!-- Applications granted NFCEE access on user builds

    See packages/apps/Nfc/etc/sample_nfcee_access.xml for full documentation.
     -->

    <!--  Google wallet release signature -->
    <signer android:signature="3082044c30820334a003020102020900a8cd17c93da5d990
    [snip] ... 0d52838c82f63f742d74ff79586a5cbb7faf7198a84bcf744310e9e927597f00
     />
    <!--- Max's signature -->
    <signer android:signature="3082033830820220a0030201020204510043d7300d06092a
    3025553310b3009060355040813024e593110300e06035504071307556e6b6e6f776e311030
    [snip] ... 5ff3db2f8efd7cb4d5657160e75c8028661772d5ccf23c10a63b74e76381b60a
    ad6afc03ec80401413b5f1d44c70b8c8718d4dc872fb3b3adec2fec82b30428a1349db2ebbe
    e48cf2ba4a428648" />
</resources>
```

SE Communication Overview

- We know how to talk to
  - UICC SE Wired interface
    - Outside the phone
    - Inside the phone
  - Embedded SE Wired interface
    - Inside the phone
- Next
  - Talk to the virtual interface (Wireless)

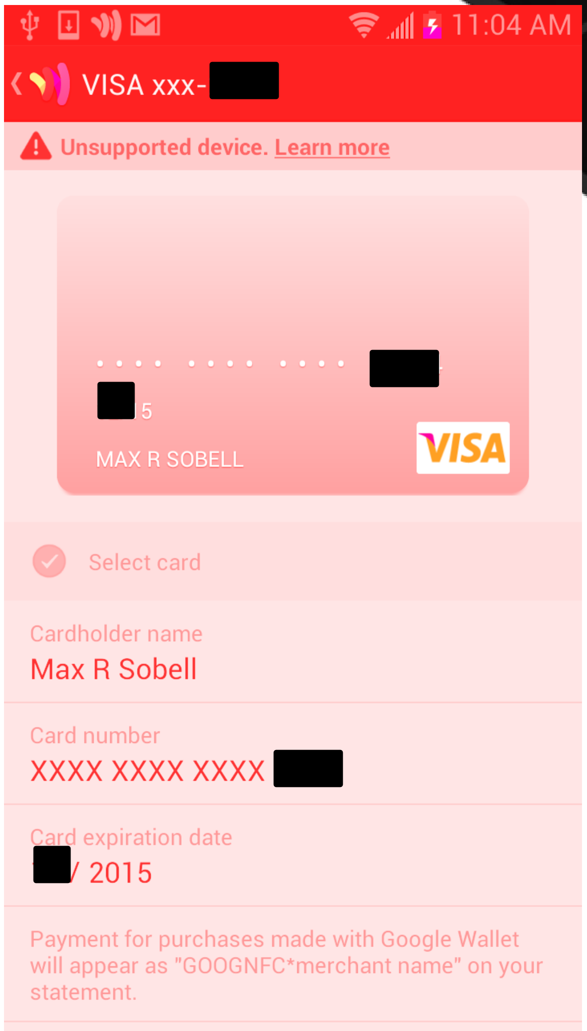
Contactless Interface

- Different attack surface
  - Range
    - 4-10 cm (in the lab…)
  - Intended functionality
    - Pay for stuff! Not change cards, etc.
  - Available data
    - Your payment authorizations!
- Functions much like standard contactless card

Google Wallet

My credit card

- It's a Visa
- No, I'm not giving you any more info
- Fine, it expires in 2015
- After messing with the contact interface, it took a looong time to get this to work :P



11:04 AM

VISA xxx-[redacted]

Unsupported device. Learn more

Select card

Cardholder name
Max R Sobell

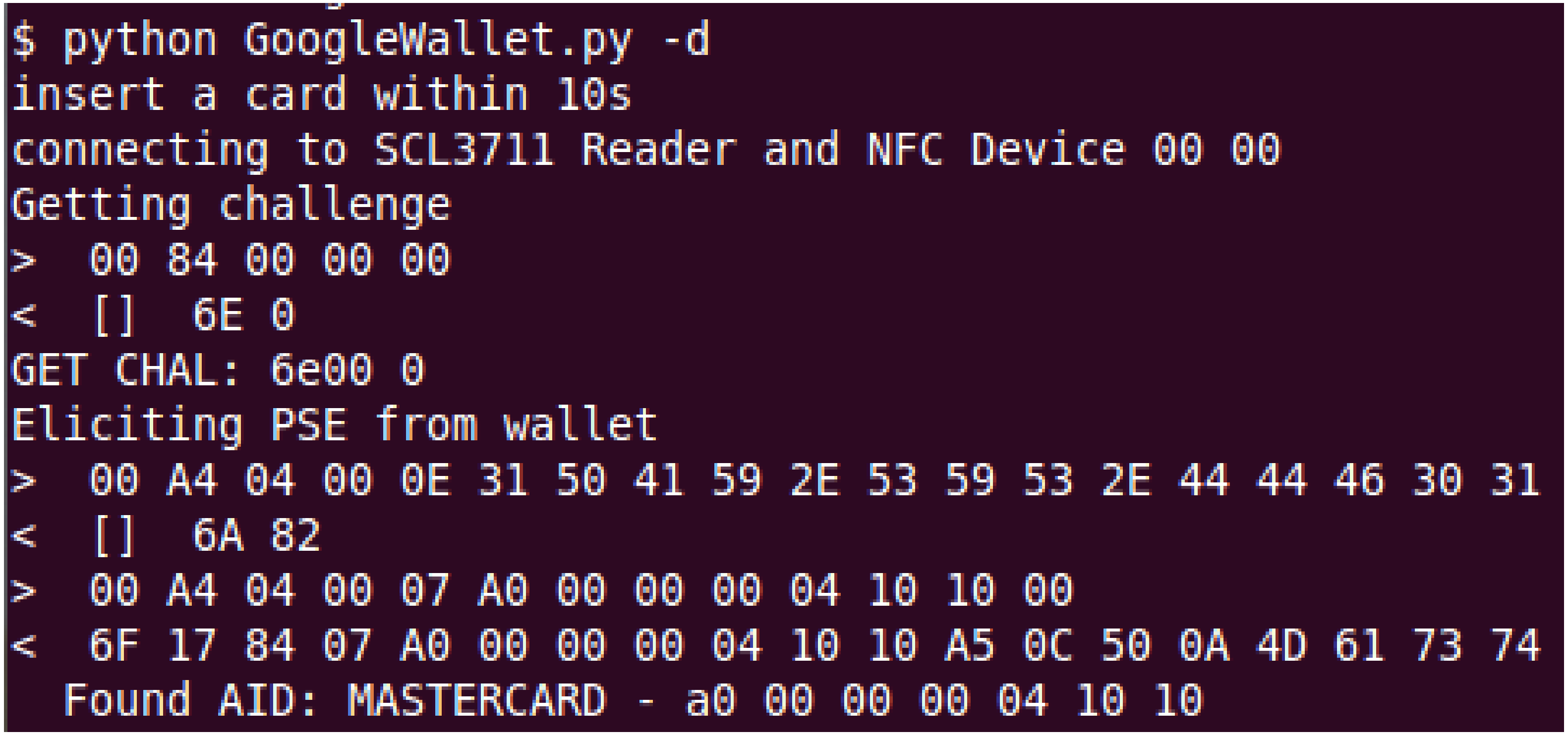
Card number
XXXX XXXX XXXX [redacted]

Card expiration date
[redacted] 2015

Payment for purchases made with Google Wallet will appear as "GOOGNFC*merchant name" on your statement.

Google Wallet Contactless

- Contactless interface
  - RFIDIOT (slight mods)



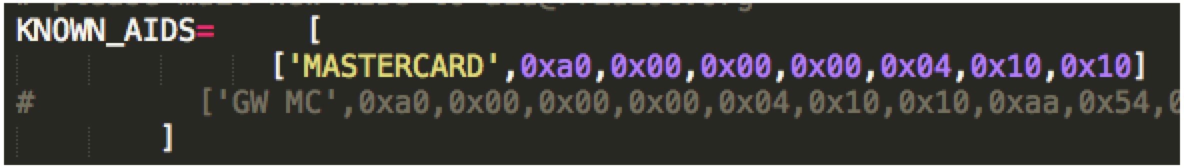
```
$ python GoogleWallet.py -d
insert a card within 10s
connecting to SCL3711 Reader and NFC Device 00 00
Getting challenge
>   00 84 00 00 00
<   []  6E 0
GET CHAL: 6e00 0
Eliciting PSE from wallet
>   00 A4 04 00 0E 31 50 41 59 2E 53 59 53 2E 44 44 46 30 31
<   []  6A 82
>   00 A4 04 00 07 A0 00 00 00 04 10 10 00
<   6F 17 84 07 A0 00 00 00 04 10 10 A5 0C 50 0A 4D 61 73 74
  Found AID: MASTERCARD - a0 00 00 00 04 10 10
```

What does the POS see?

- Wait, what? I loaded a Visa
  - Mastercard AID responds
- Google's "Cloud Wallet"
- Backend processing
- More details: http://intrepidusgroup.com/insight/2012/08/the-cloud-comes-to-your-nfc-wallet/

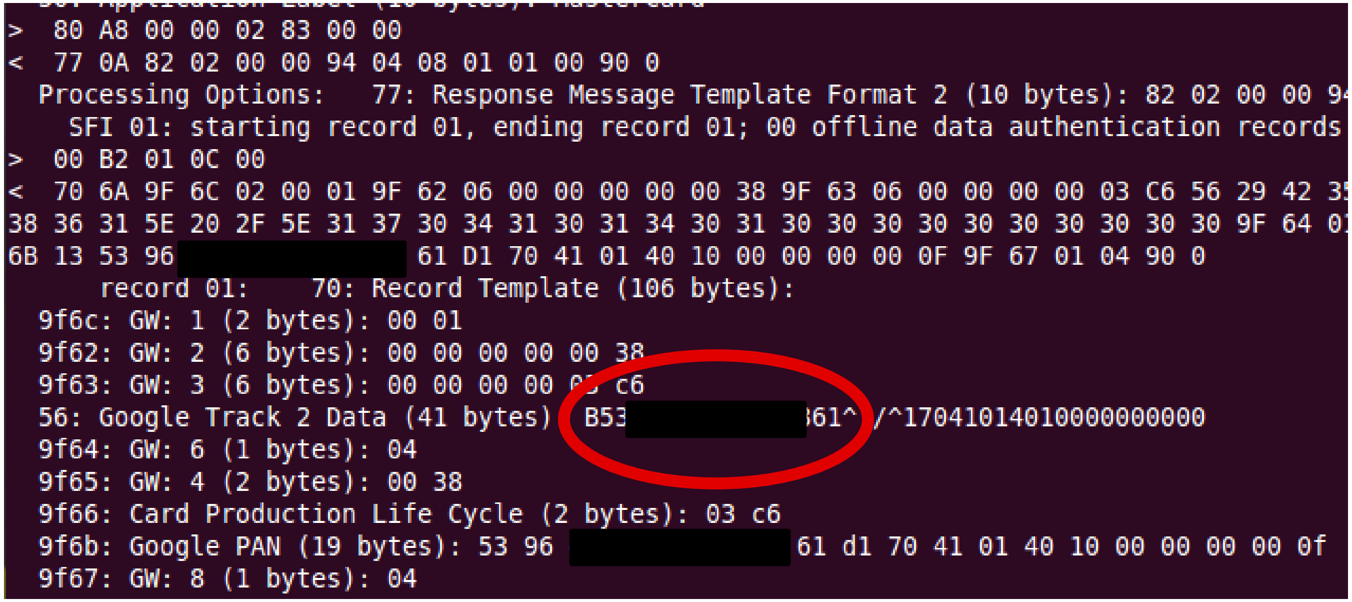
Now what?

- Google Wallet makes it easy



```
KNOWN_AIDS=      [
                ['MASTERCARD',0xa0,0x00,0x00,0x00,0x04,0x10,0x10]
#            ['GW MC',0xa0,0x00,0x00,0x00,0x04,0x10,0x10,0xaa,0x54,0
        ]
```

- AID like a directory – ask it for its files



```
>  80 A8 00 00 02 83 00 00
<  77 0A 82 02 00 00 94 04 08 01 01 00 90 0
  Processing Options:   77: Response Message Template Format 2 (10 bytes): 82 02 00 00 9
    SFI 01: starting record 01, ending record 01; 00 offline data authentication records
>  00 B2 01 0C 00
<  70 6A 9F 6C 02 00 01 9F 62 06 00 00 00 00 00 38 9F 63 06 00 00 00 00 03 C6 56 29 42 3
38 36 31 5E 20 2F 5E 31 37 30 34 31 30 31 34 30 31 30 30 30 30 30 30 30 30 30 30 9F 64 0
6B 13 53 96 [redacted] 61 D1 70 41 01 40 10 00 00 00 00 0F 9F 67 01 04 90 0
      record 01:    70: Record Template (106 bytes):
  9f6c: GW: 1 (2 bytes): 00 01
  9f62: GW: 2 (6 bytes): 00 00 00 00 00 38
  9f63: GW: 3 (6 bytes): 00 00 00 00 03 c6
  56: Google Track 2 Data (41 bytes) B53[redacted]61^ /^17041014010000000000
  9f64: GW: 6 (1 bytes): 04
  9f65: GW: 4 (2 bytes): 00 38
  9f66: Card Production Life Cycle (2 bytes): 03 c6
  9f6b: Google PAN (19 bytes): 53 96 [redacted] 61 d1 70 41 01 40 10 00 00 00 00 0f
  9f67: GW: 8 (1 bytes): 04
```

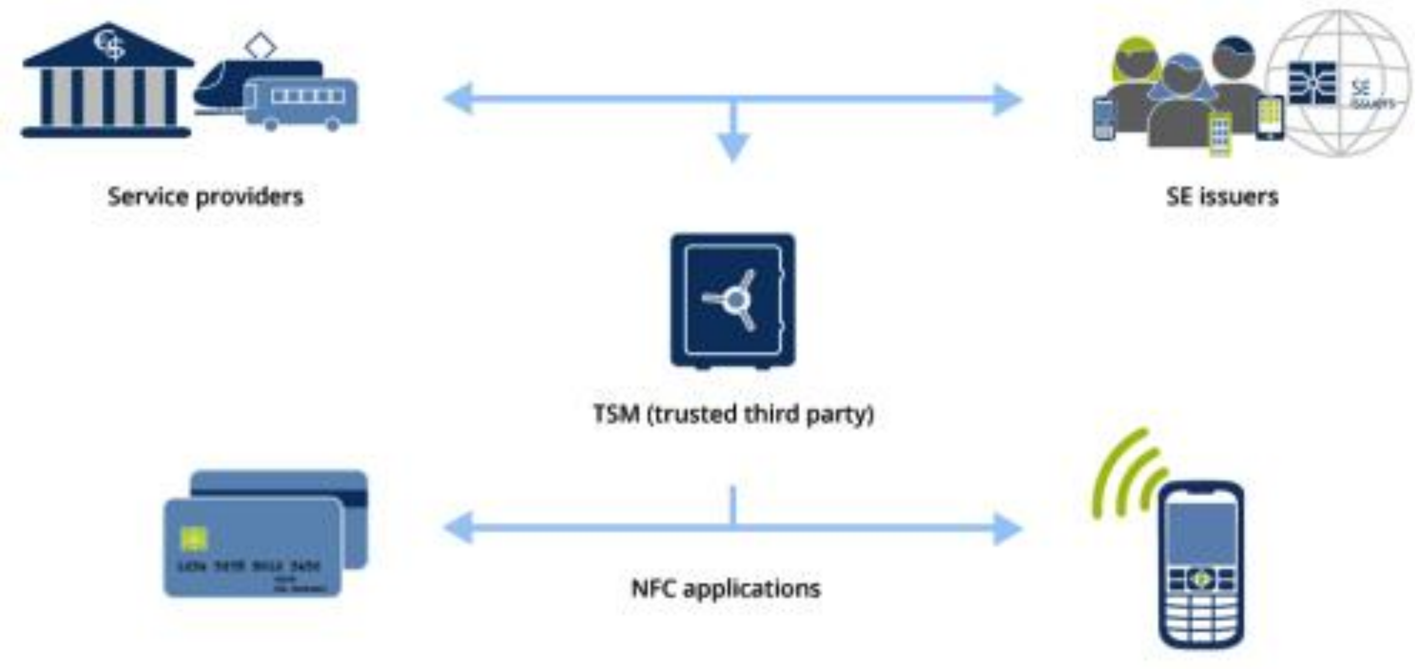
PROVISIONING

Provisioning

- How can we get the data securely from the issuer (bank) to the SE?
- Who is allowed to update that data once it is in the SE?
- Who performs de-provisioning in the case of a lost or stolen device?

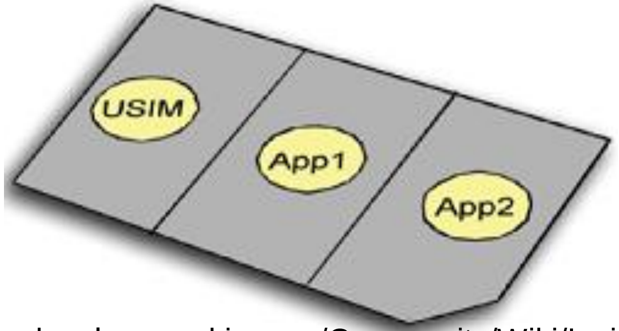
Secure Data Transport

- Trusted Service Manager
  - Global Platform (Visa's Open Platform)
    - Open specs; management of smartcards



TSM Keys

- Global Platform defines "Card Manager"
- TSM authenticates to Card Manager with Issuer Security Domain (ISD) Keys
  - List applets
  - Delete applets
  - Create new security domains
- Performs high-level management of SE environment



http://www.developer.nokia.com/Community/Wiki/Inside_NFC:_secure_payment_technology

TSM Applets

- Applets
  - Contained within ISD
  - Transit encrypted with issuer-controlled Supplementary Security Domain (SSD) Keys
  - Sandboxed areas within SE
  - Vetted before being loaded into SE
- Very difficult to attack applet to applet

Contactless interface

- Applet selection controlled by PPSE
  - Global Platform card manager
- PPSE selected first by POS, directs queries to correct (active) applet
- POS can break spec and skip PPSE
- Applet responds following EMV standards (or not)

EARLY ISSUES

Early Wallet Issues

- Google Wallet first to market
- Lots of scrutiny

PIN Issues

- Mistake #1
  - PIN outside of the SE
  - http://intrepidusgroup.com/insight/2012/02/google-wallet-pin-brute-forcing/
  - Credit to Corey and others

PIN Issues

- If the PIN is outside the SE…
  - How does the wallet get unlocked?
  - APDUs!
- SE has built-in brute force protection
  - Android filesystem does not!

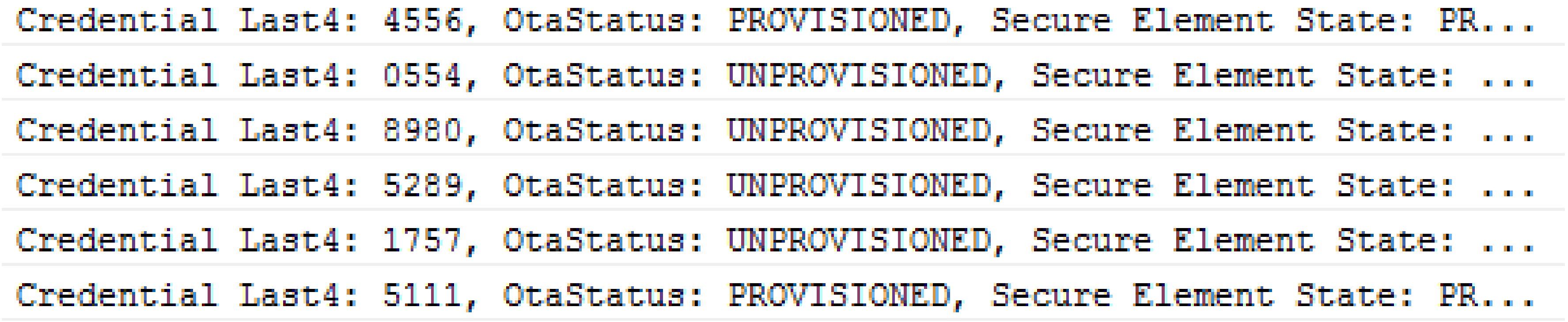
Mistake #2

- Mistake #2
  - Verbose logging (last 4 digits)
  - Trigged from Android app bug
  - Unprotected Broadcast Receiver

```
<receiver
android:name="com.google.android.apps.wallet.util.
LoggingPriorityChangeReceiver">
    <intent-filter>
        <action
android:name="com.google.android.apps.wallet.util.
CHANGE_LOG_PRIORITY_LEVEL" />
    </intent-filter>
</receiver>
```

Mistake #2

- Small privacy issue
- Bigger customer trust issue



```
Credential Last4: 4556, OtaStatus: PROVISIONED, Secure Element State: PR...
Credential Last4: 0554, OtaStatus: UNPROVISIONED, Secure Element State: ...
Credential Last4: 8980, OtaStatus: UNPROVISIONED, Secure Element State: ...
Credential Last4: 5289, OtaStatus: UNPROVISIONED, Secure Element State: ...
Credential Last4: 1757, OtaStatus: UNPROVISIONED, Secure Element State: ...
Credential Last4: 5111, OtaStatus: PROVISIONED, Secure Element State: PR...
```

Wrap Up

Thanks for listening!

@msobell

max@intrepidusgroup.com

https://github.com/msobell/MobileWallet

IG is hiring!

Thanks to Corey & the IG crew