

NFC Mobile Wallets: Why I'd Rather Swipe Your Credit Card



Max Sobell
Intrepidus Group

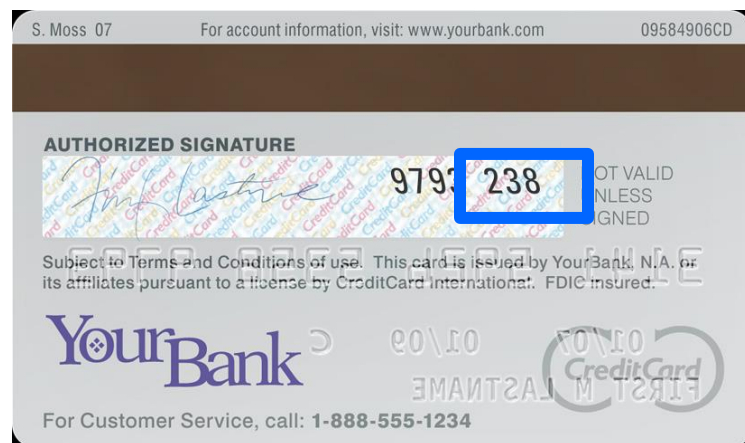
- Credit Card security features
 - As we know them now...
- Secure Element (SE)
 - What is this magical thing?
- Operating system protections (Android)
 - And how to get around them
- APDUs and communication with the SE
 - Once we can talk to it, what can we ask?
- Wallet mistakes
 - Derp

What this talk DOES NOT CONTAIN

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

What's in a credit card?

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



- 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?

- All static information
- Restaurant scenario
 - Waiter takes your card in back...
- Authorization for **unlimited** transactions
- All static data!



So plastic cards suck...

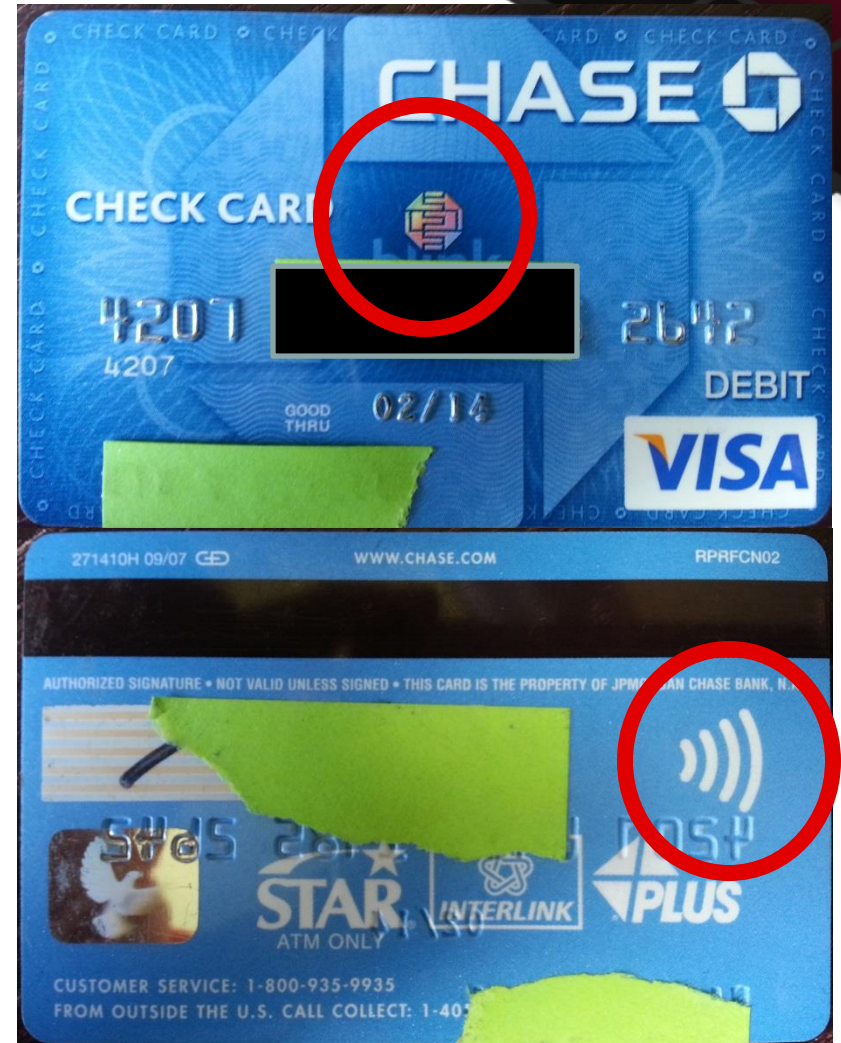


... And we're generally fine with that

How about my contactless card?

How about a contactless card?

- Contactless cards should be labeled differently
- Same PAN/CSC/Additional info protections
- How about contactless protections?



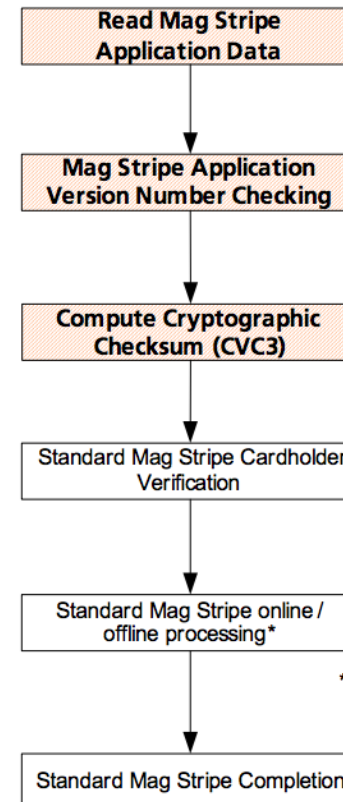
- From PayPass (Mastercard) documentation
- Bolt-on additions
- Calculate CVC
- Standard magstripe verification process

Mag Stripe processing is

Normal

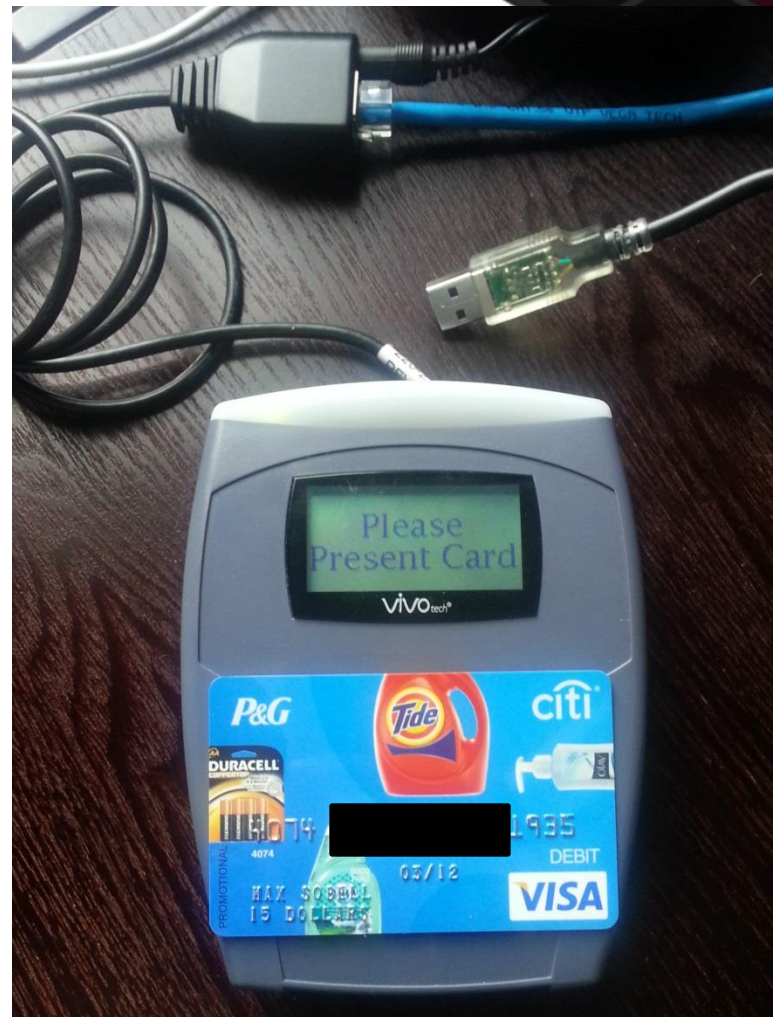
New

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



*With PayPass indication

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





COMM open. Please tap card...

4207xxxxxxxx2642^CARDHOLDER/CHASE^140210100000000001000000000000104149500000
4207xxxxxxxx2642=14021010000014904501

4207xxxxxxxx2642^CARDHOLDER/CHASE^140210100000000001000000000000104347510000
4207xxxxxxxx2642=14021010000034704511

4207xxxxxxxx2642^CARDHOLDER/CHASE^140210100000000001000000000000104366520000
4207xxxxxxxx2642=14021010000036604521

PAN: 4207xxxxxxxx2642

DCSC: 149 0

DCSC: 347 1

DCSC: 366 2

- PAN is the same
- DCSC is present
 - Gives one authorization at a time
 - Must be in order!
- 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...

3702xxxxxxxx8138^3001 VALUED CUSTOMER 007^1702101100105175
3702xxxxxxxx8138=170210110010517500700

3702xxxxxxxx8138^3001 VALUED CUSTOMER 008^1702101091203130
3702xxxxxxxx8138=170210109120313000800

3702xxxxxxxx8138^3001 VALUED CUSTOMER 009^1702101090619137
3702xxxxxxxx8138=170210109061913700900

PAN: 3702xxxxxxxx8138

DCSC: 7 100105 75

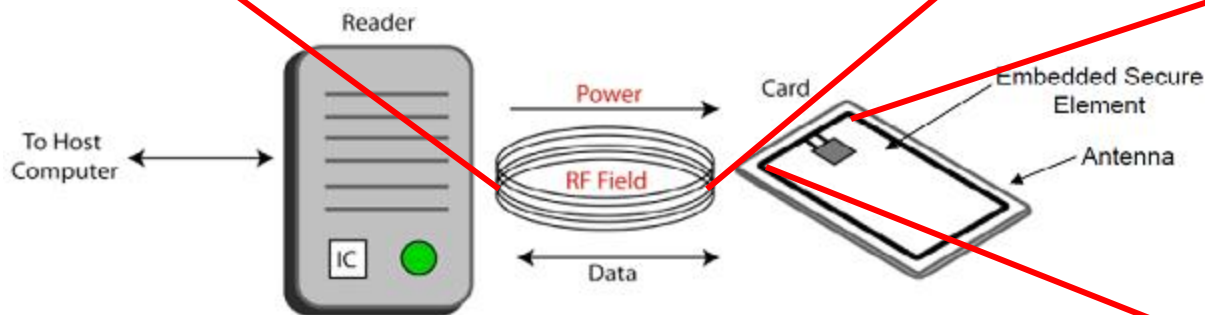
DCSC: 8 091203 30

DCSC: 9 090619 37

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

- What is actually going on?
- Communication with SE via NFC
- APDU commands exchanged

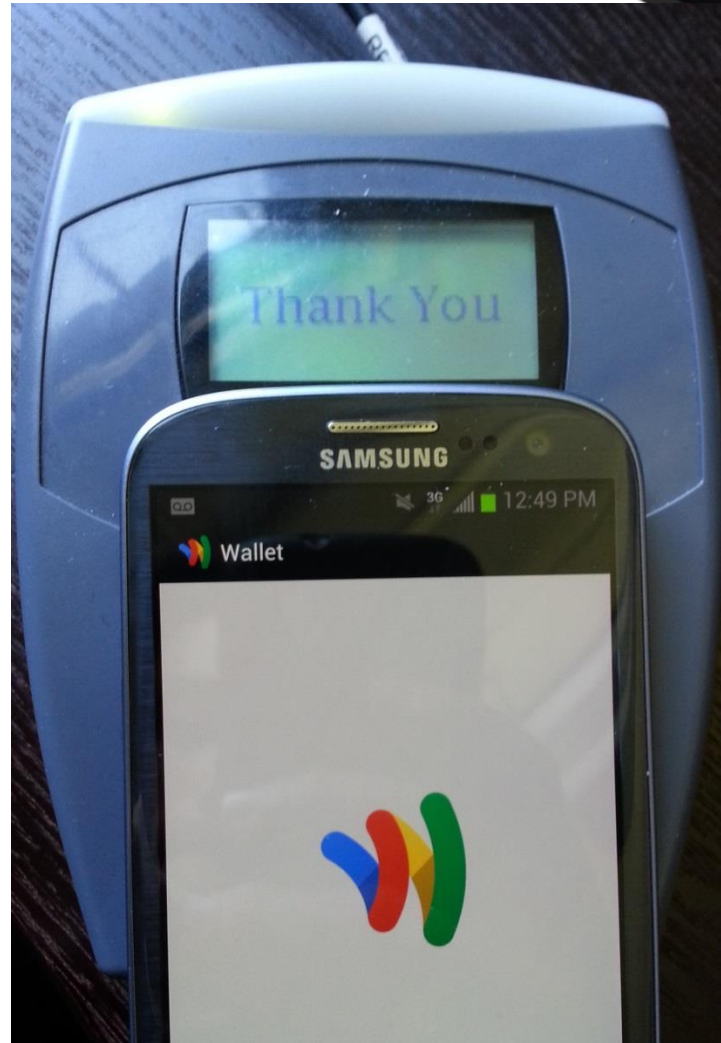
```
> 00 A4 04 00 08 A0 00 00 00 03 10 10 01 00  
< [] 6A 82  
> 00 A4 04 00 08 A0 00 00 00 03 10 10 02 00  
< [] 6A 82
```




- How does this relate to mobile wallets?
 - In the US, contactless credit card == mobile wallet to the Point-of-Sale (PoS) terminal
 - Contactless is a bolt-on accessory
- Next step: EMV
 - Europay MasterCard and Visa
 - “Chip and PIN”

Mobile wallet?

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





COMM open. Please tap card...

5396xxxxxxxx1958^ / ^17031014010040870032
5396xxxxxxxx1958=17031014010040357032

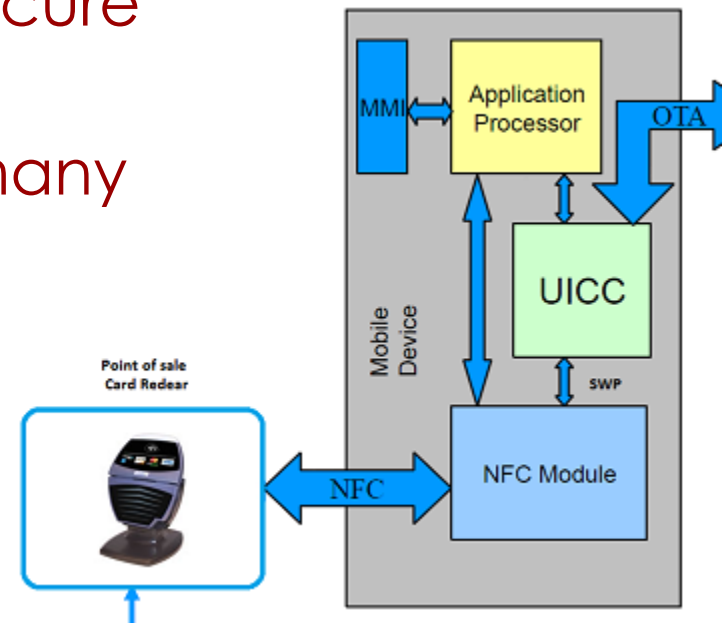
5396xxxxxxxx1958^ / ^17031014010041286522
5396xxxxxxxx1958=17031014010041649522

5396xxxxxxxx1958^ / ^17031014010042794932
5396xxxxxxxx1958=17031014010042949932

PAN: 5396xxxxxxxx1958

DCSC: 087003
DCSC: 128652
DCSC: 279493

- 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



- 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.

- 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

Standard Smartcard



- JLoad (Giesecke & Devrient)



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 41



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

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

- Easy to talk to SE outside of the phone...
- How can we talk to the Embedded SE from the OS?
 - S2C SignalIn/SignalOut (NFC-Wired Interface (WI) standard)
 - Several OS protections
 - 3 Modes
 - Off
 - Wired (talk to OS)
 - Virtual (talk to POS payment terminal)

- /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 install/reinstall to /system/app/Wallet.apk

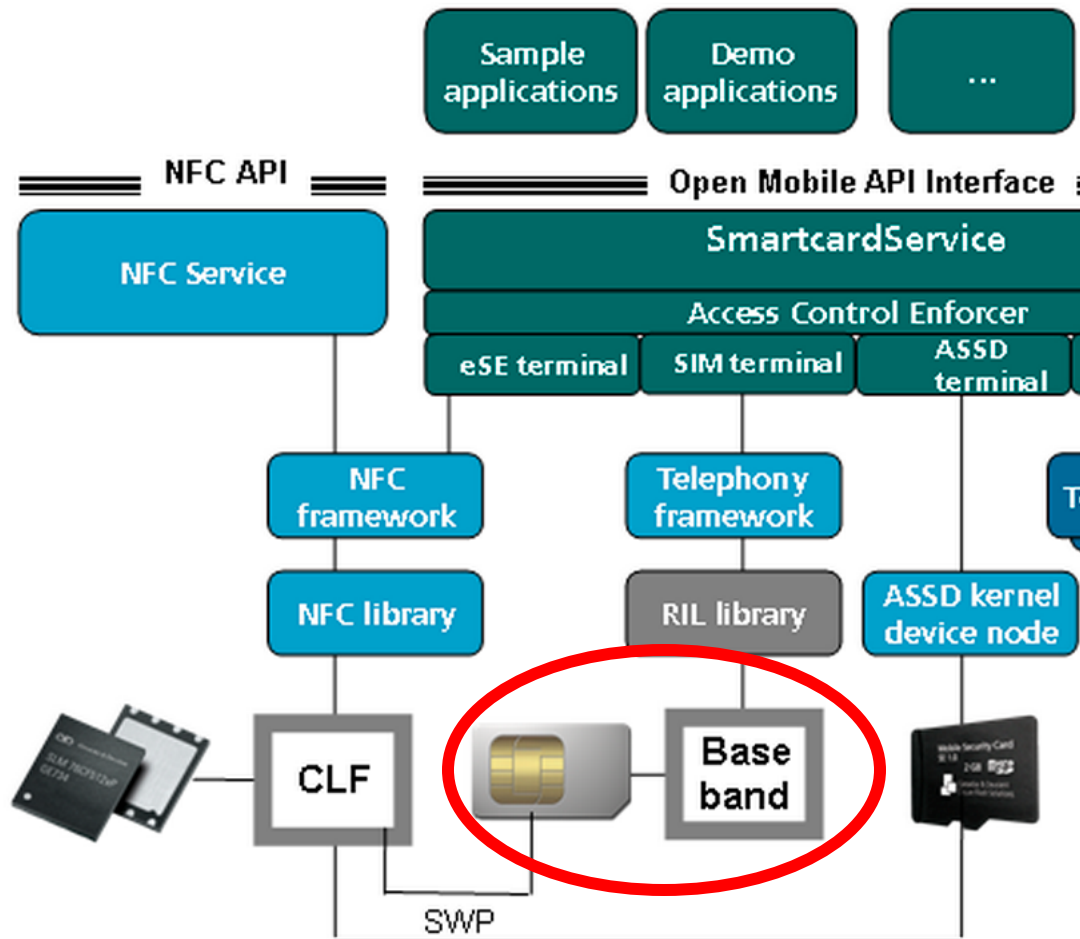
```
greywind:GoogleWallet max$ adb install -r Wallet/dist/Wallet.apk
1314 KB/s (4803535 bytes in 3.569s)
  pkg: /data/local/tmp/Wallet.apk
Success
```

```
<?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>
```

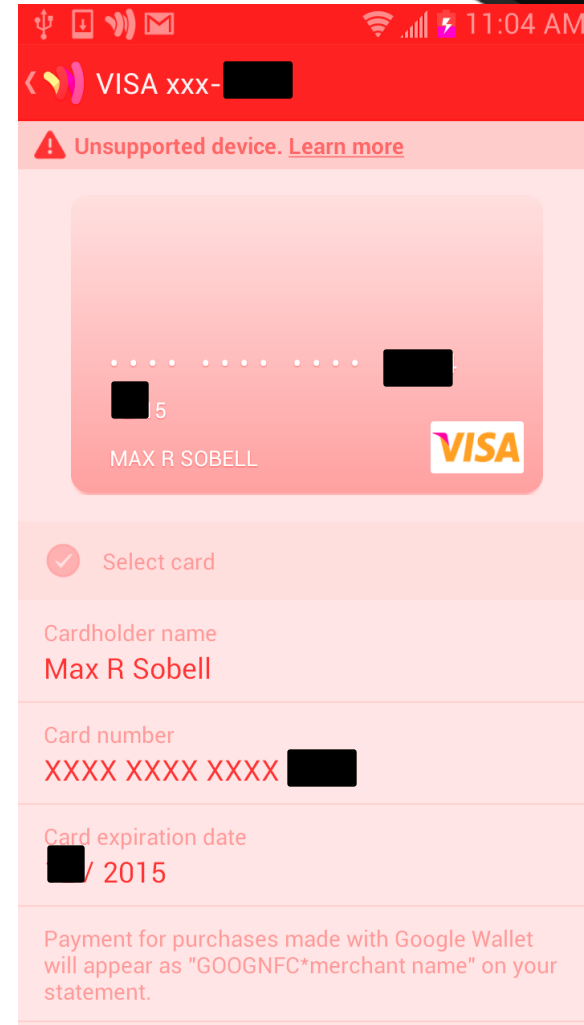
- How can we talk to the UICC SE from the OS?
 - UICC SE connected to baseband
 - Communication via Radio Interface Layer (RIL)
 - rild (open source) + proprietary OEM component
 - SEEK for Android
 - SIMalliance Open Mobile API (SmartCard API)
 - Not required/used for Embedded SE



- Whole different ball game/attack surface
 - Range
 - 4-10 cm (pshhhh)
 - Intended functionality
 - Pay for stuff! Not change cards, etc.
 - Available data
 - Uh, your payment creds
- Functions much like contactless card from before

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



- 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
```

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

- 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],
]
```

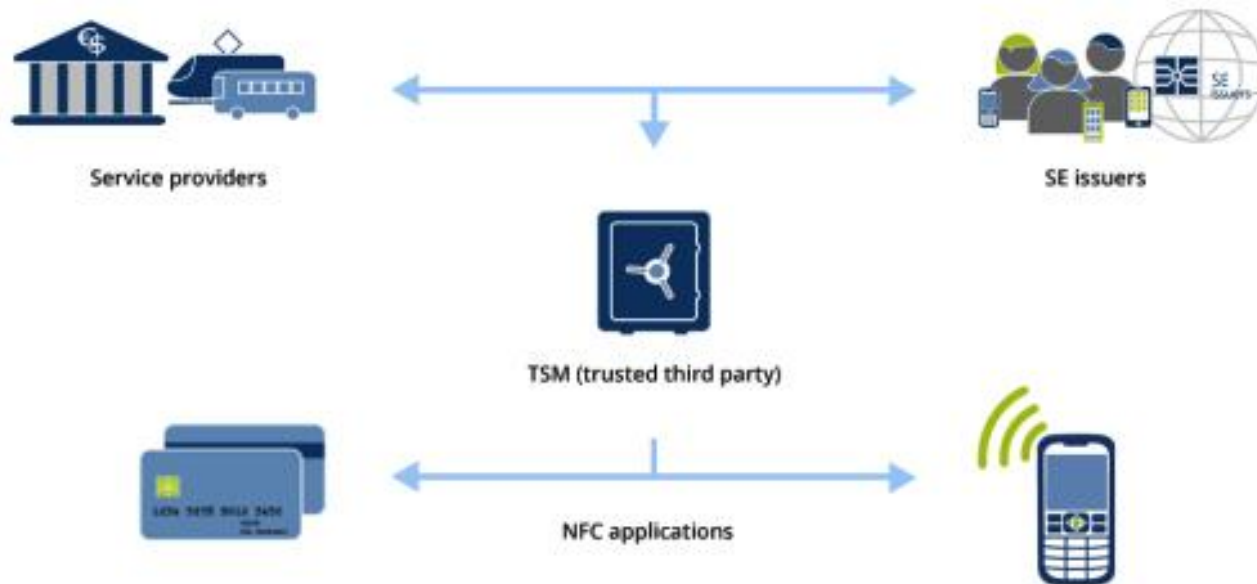
- Google Wallet makes it easy

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

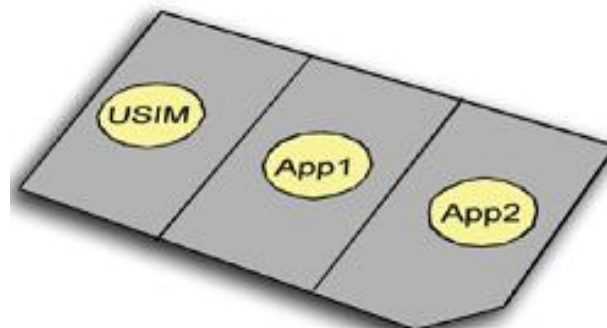
- AID like a directory – ask it for its files

```
> 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 94
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 35
38 36 31 5E 20 2F 5E 31 37 30 34 31 30 31 34 30 31 30 30 30 30 30 30 30 9F 64 01
6B 13 53 96 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 0f
9f67: GW: 8 (1 bytes): 04
```

- How does the data get there in the first place?
 - Global Platform (Visa's Open Platform)
 - Open specs; management of smartcards



- Global Platform defines “Card Manager”
- TSM authenticates to Card Manager with Issuer Security Domain (ISD) Keys
 - List applets
 - Delete applets
 - Create new security domains
- Supplementary Security Domain (SSD) Keys
 - Sandboxed areas within SE



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

- Google Wallet first to market
- Lots of scrutiny



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



- 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
 - Verbose logging (last 4 digits)
 - Triggered 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>
```

- 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...
```

Thanks for listening!

@msobell

max@intrepidusgroup.com

<https://github.com/msobell/MobileWallet>

IG is hiring!

Thanks to Corey & the IG crew