Foundstone
A division of McAfee

Introduction to NFC

robert.portvliet@foundstone.com

Twitter: @rportvliet

Introduction to NFC

- Overview
  - Introduction
  - Hardware
  - Software
  - Attacks

Introduction to NFC

- NFC Introduction
  - What is NFC?
    - Near Field Communication
      - Set of standards for mobile devices for communicating between two devices, or a device and a tag in close proximity to one another.
      - Short range. 1-4cm typical
      - Frequency is 13.56MHz
        - Also used by NXP MIFARE, PayPass, ePassports, HID iClass
      - Data rates are 106kbps, 212kbps, and 424kbp/s.
      - NFC Forum maintains NFC standards

Introduction to NFC

- NFC Introduction
  - NFC Uses
    - Contactless Payment Systems
      - Google Wallet, ISIS,
        - Provides the ability to make credit card payments over NFC
    - Access Control
      - Hotel room keys, facility access, home security
    - Data transfer between devices
      - Android Beam
        - Uses NFC to bootstrap Bluetooth connection between devices
      - Samsung S Beam
        - Uses NFC to bootstrap Wi-Fi Direct connection between devices
    - NFC tags
      - Similar to other RFID tags, but can be programmed to perform actions on the device reading them

Introduction to NFC

- NFC Introduction
  - Mobile devices with NFC chipsets (partial list)
    - Samsung Galaxy Nexus
    - Google Nexus 7 and 10
    - Google Nexus 4
    - Samsung Nexus S
    - Samsung Galaxy S series (2-4) (Note + Note II)
    - Motorola Droid Razr HD, M, and I
    - Blackberry Curve, Z10, Q10, Bold 9790, 9900/9930
    - HTC One SV, X, X+, XL, VX, Incredible S, Amaze 4G
    - HTC Windows Phone 8X
    - Nokia Lumia 610, 620, 810, 820, 822, 920 (Win Phone 8)
    - IPhone 6?

Introduction to NFC

- NFC Introduction
  - Standards
    - ISO/IEC 14443 A/B
      - Type A and Type B proximity cards
        - » Modulation and bit encoding different between A/B
    - JIS X 6319-4
      - FeliCa
    - ISO/IEC 18092
      - Covers P2P communication between NFC devices
      - Uses parts of ISO 14443 and JIS 6319-4
    - ISO/IEC 15693
      - ISO standard for vicinity cards
      - Some NFC readers can read these cards as well
    - ISO 7816-4
      - Used in Card Emulation Mode / Secure Elements

Introduction to NFC

- NFC Introduction
  - 14443-1 – Physical characteristics
  - 14443-2 – Radio Frequency power and signal
  - 14443-3 – Initialization and Anti-Collision
  - 14443-4 – Transmission protocol

Introduction to NFC

- NFC Introduction
  - Inductive Coupling
    - Initiator generates field / target modulates
  - Frequency = 13.56MHz (HF)
  - ASK modulation
  - PCD to PICC:
    - 212kbps and 424kbp/s = Manchester encoding and modulates at 10%.
    - 106kbps = Modified Miller encoding, modulates at 100%.
  - PICC to PCD:
    - Manchester encoding and modulates at 10%.

Introduction to NFC

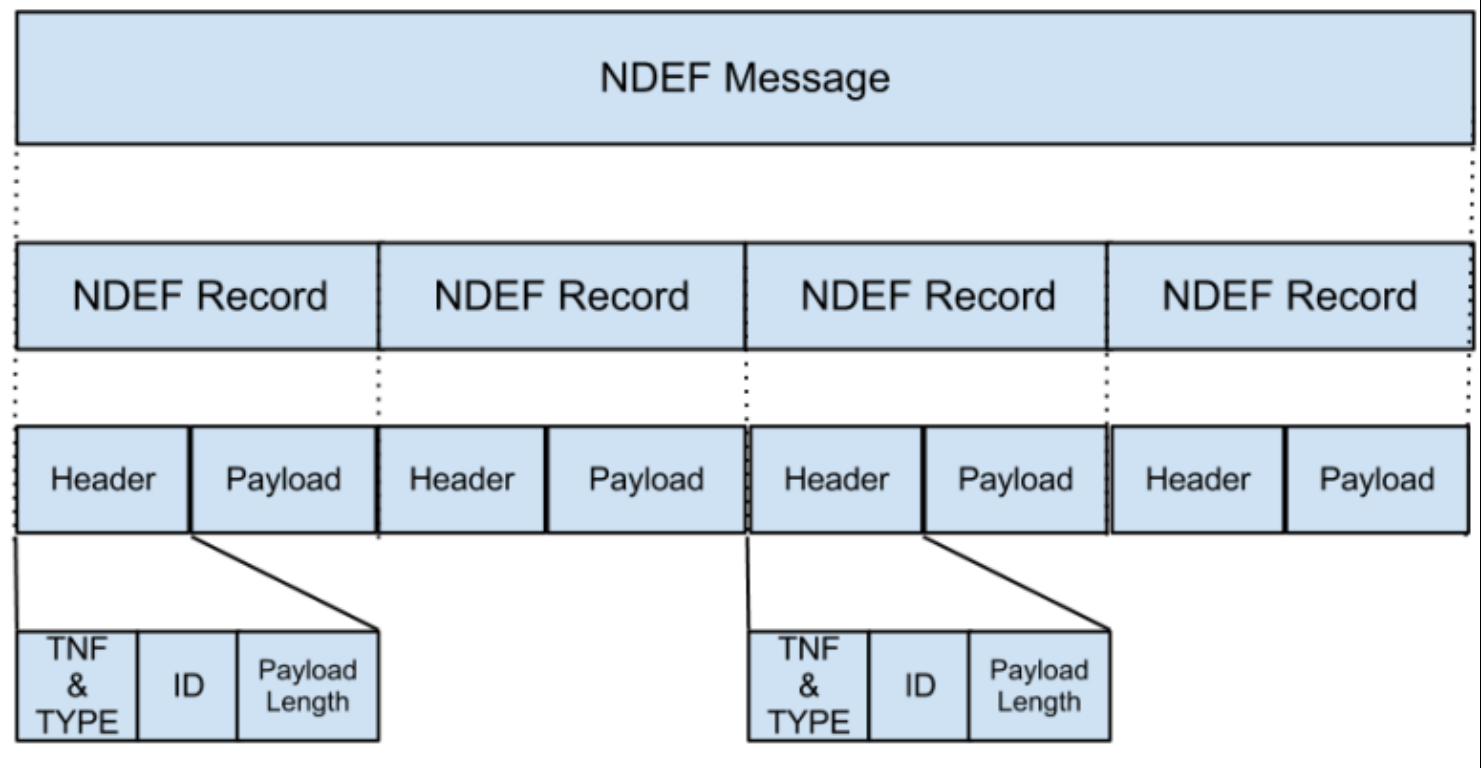
- NFC Introduction
  - 3 modes of operation
    - Reader\Writer
      - Device behaves as a Proximity Coupling Device (PCD)
    - Peer-to-Peer (P2P)
      - Two devices exchange data, such as Android Beam
      - Two modes: Active and Passive
      - Defined in ISO 18092 (NFCIP-1)
      - Frames: Polling Request, Polling Response, Transport
    - Card Emulation
      - Mobile device behaves as a PICC (Proximity Inductive Coupling Card)
      - Either done with a Secure Element or in software (HCE)
      - HCE present in Android 4.4 and CyanogenMod 10

Introduction to NFC

- NFC Introduction
  - NDEF
    - NFC Data Exchange Format
      - Used to encapsulate data sent between two devices or a reader/writer and a card
    - NDEF Message
      - Contains one or more NDEF records (no limit on how many)
    - NDEF Record
      - Encapsulates an NDEF payload
      - Can be URI, Text, MIME Types, Handover Parameters, etc.
    - NDEF Payload
      - Application data carried in an NDEF record
      - Can be up to 2^31 - 1 octets in size (4096MB)
      - NDEF does not care about payload content

Introduction to NFC

- NFC Introduction
  - NDEF Structure



Introduction to NFC

- NFC Introduction
  - NDEF Record Types

Record Type	Description	Full URI Reference	Specification Reference
Sp	Smart Poster	urn:nfc:wkt:Sp	NFC Forum Smart Poster RTD
T	Text	urn:nfc:wkt:T	NFC Forum Text RTD
U	URI	urn:nfc:wkt:U	NFC Forum URI RTD
Gc	Generic Control	urn:nfc:wkt:Gc	NFC Forum Generic Control RTD**
Hr	Handover Request	urn:nfc:wkt:Hr	NFC Forum Connection Handover Specification
Hs	Handover Select	urn:nfc:wkt:Hs	NFC Forum Connection Handover Specification
Hc	Handover Carrier	urn:nfc:wkt:Hc	NFC Forum Connection Handover Specification
Sig	Signature	urn:nfc:wkt:Sig	NFC Forum Signature RTD

Introduction to NFC

- NFC Introduction
  - URI Identifier Codes (partial list)

Value	Protocol
0x00	No Prepend
0x01	http://www.
0x02	https://www
0x03	http://
0x04	https://
0x05	tel:
0x06	mailto:
0x08	ftp://ftp.
0x09	ftps://

Value	Protocol
0x0A	sftp://
0x0B	smb://
0x0C	nfs://
0x0D	ftp://
0x0E	dav://
0x010	telnet://
0x011	map:
0x012	rtsp://
0x014	pop:

Value	Protocol
0x15	sip:
0x16	sips:
0x17	tftp:
0x18	btspp://
0x19	btl2cap://
0x1A	btgoep://
0x1B	tcpobex://
0x1C	irdaobex://
0x1D	file://

Introduction to NFC

- NFC Introduction
  - NFC in Android
    - Mandatory on Android NFC devices
      - NfcA (ISO 14443-3A)
      - NfcB (ISO 14443-3B)
      - NfcF (JIS 6319-4)
      - NfcV (ISO 15693)
      - ISO-DEP (ISO 14443-4)
      - Ndef on Type 1-4
    - Optional
      - MIFARE
      - NfcBarcode
      - NdefFormatable

Introduction to NFC

- NFC Introduction
  - NFC in Android
    1. Tag object created when tag is discovered
    2. Passed to an activity encapsulated in an intent
    3. Selects best activity to handle it
       1. Foreground Activity Dispatch
       2. NDEF Data Dispatch
       3. Technology Dispatch
       4. Tag Dispatch
    4. Apps register intent filter in AndroidManifest.xml

    - Android 4.0 introduced Android Application Records
      - Embed package name of app in NDEF record and Android will launch that app when tag is scanned

Introduction to NFC

- Reader\Writer mode
  - NFC Tag Types
    - Type 1
      - Memory capacity is 96 bytes, expandable to 2KB
      - Read and re-write capable, user can configure as read-only
    - Type 2
      - Memory capacity is 48 bytes, expandable to 2KB
      - Read and re-write capable, user can configure as read-only
    - Type 3
      - Theoretical memory limit of 1MByte per service
      - Configured by manufacturer as read + re-write, or RO
    - Type 4
      - Memory capacity varies, up to 32 KB per service
      - Configured by manufacturer as read + re-write, or RO

Introduction to NFC

- NFC Tags (partial list)

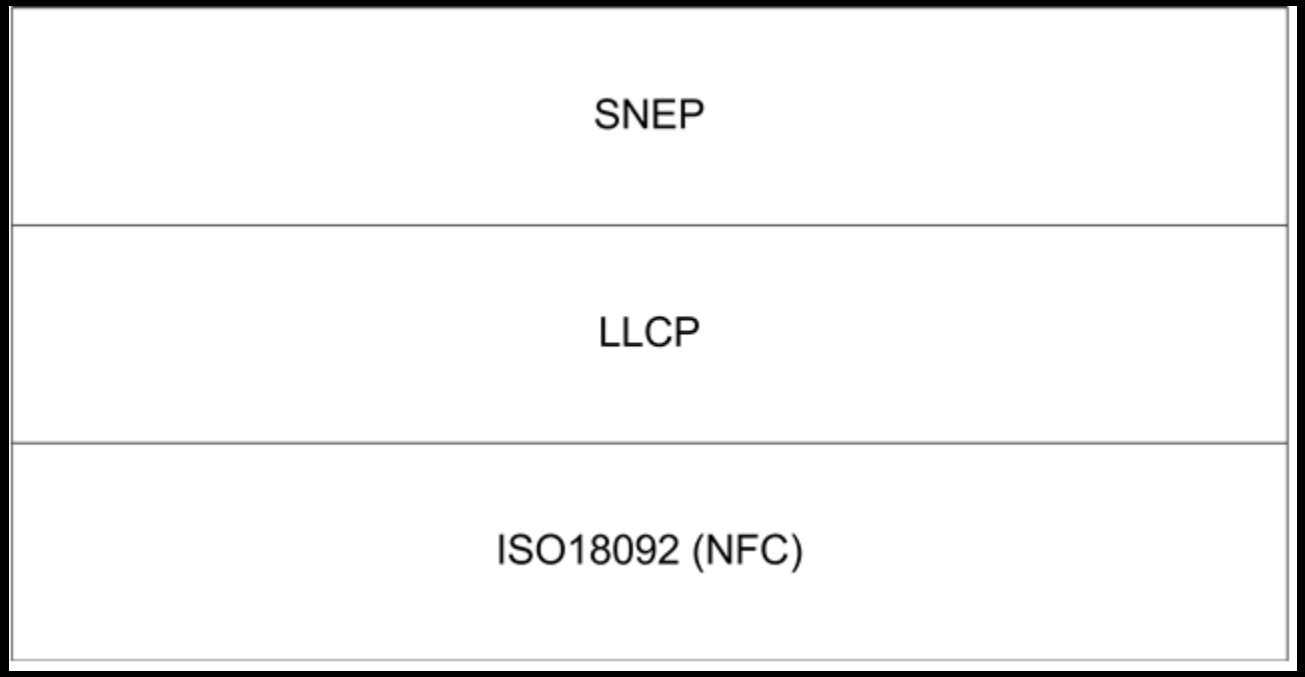
Name	Type	Memory
Innovision Topaz	Type 1	96 bytes
NXP MIFARE Ultralight	Type 2	48 bytes
NXP MIFARE Ultralight C	Type 2	144 bytes
NXP NTAG203	Type 2	144 bytes
Sony FeliCa 4K	Type 3	4096 bytes
NXP DESFire EV1 2k	Type 4	2048 bytes
NXP DESFire EV1 4k	Type 4	4096 bytes
NXP DESFire EV1 8k	Type 4	8192 bytes
NXP SmartMX	Type 4	32 kBytes
NXP MIFARE Classic 1k	NXP Specific	768 bytes
NXP MIFARE Classic 4k	NXP Specific	3584 bytes

Introduction to NFC

- Reader\Writer mode
  - NFC-V
    - Tags defined in ISO15693
      - ISO standard for vicinity cards
      - Communicates over 13.56MHz, same frequency as NFC
    - Not yet standardized in NFC forum specs
    - Code support exists in Android
      - android.nfc.tech.NfcV
    - Tags:
      - HID ICLASS
      - NXP ICODE
      - TI Tag-it (TRF796x and TRF797x), and HF-I tags
      - STMicroelectronics
        - » Dual Interface EEPROM (M24LRxx).
        - » LRIxx family (LRI1K, LRI2K, LRIS2K and LRIS64K)

Introduction to NFC

- Peer to Peer Mode (P2P)
  - Protocol Stack



Introduction to NFC

- Peer to Peer Mode (P2P)
  - Protocols
    - NFC-IP (ISO 18092)
      - Initiator
      - Target
      - Active and Passive modes
      - Provides collision detection/avoidance
      - Manchester Encoding at all data rates
      - Frames
        - » Polling request/response
        - » Transport
      - Frame format
        - » Preamble/SYNC/Length/Payload/CRC

Introduction to NFC

- Peer to Peer Mode (P2P)
  - Protocols
    - LLCP (Logical Link Control Protocol)
      - Layer-2 protocol which supports P2P communication between two NFC enabled devices
      - Necessary for bi-directional communications
      - Two service types
        - » Connectionless (minimal setup)
        - » Connection-oriented (provides reliable delivery and flow control)
      - Uses 5 field Payload Data Units
        - » DSAP, PTYPE, SSAP, Sequence, Information
      - Other protocols ride on top of it
        - » OBEX, IP, NPP, SNEP

Introduction to NFC

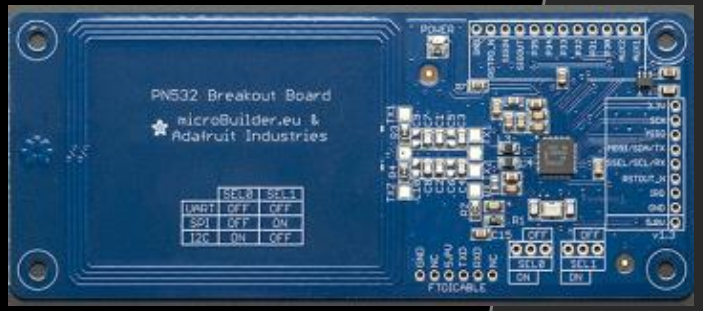
- Peer to Peer Mode (P2P)
  - Protocols (cont.)
    - NPP (NDEF Push Protocol)
      - Non standards based Android protocol (com.android.npp) to push an NDEF message from one device to another.
      - Connect, send NPP header + NDEF entries, disconnect
      - Used by default on Android from v2.3 – v3.2
    - SNEP (Simple NDEF Exchange Protocol)
      - Transfers data via GET and PUT messages
      - Supports fragmentation
      - Uses LLCP connection-oriented transport to provide reliable data exchange
      - Used by default on Android 4.0 (ICS) and later.
      - Message: Version, Request/Response, Length, Information

Introduction to NFC

- Hardware
  - NFC Readers/Writers
    - Requirements:
      - » Libnfc compatibility
      - » Be able to do card emulation
      - » Be able to perform P2P
      - » Communicate with NFC-A, NFC-B, NFC-F and DEP targets
      - » Need to be able to abort commands, and cancel polling or acting as a target.
    - Readers/Writer Reference:
      - » http://nfc-tools.org/index.php?title=Devices_compatibility_matrix

Introduction to NFC

- Hardware
  - NFC Readers/Writers
    - PN532 NFC/RFID Controller Breakout Board
    - Can read/write NFC tags
    - Interfaces: UART, SPI and I2C (two-wire)
    - Supports ISO14443 type A & B, FeliCa, and MIFARE tags
    - Supports Card Emulation Mode
    - Price: $39.95
      - » https://www.adafruit.com/products/364
    - Great with a Raspberry Pi
      - » http://learn.adafruit.com/adafruit-nfc-rfid-on-raspberry-pi/overview



Introduction to NFC

- Hardware
  - NFC Readers/Writers
    - SCM SCL3711 Contactless Mobile Reader and Writer
    - Interfaces: USB
    - Chipset: PN533
    - Supports ISO14443 type A & B, FeliCa, and MIFARE tags
    - Supports Card Emulation Mode
    - Price: $39.00

Introduction to NFC

- Hardware
  - NFC Readers/Writers
    - OpenPCD2
      - » Open Source Hardware\Firmware for NFC/RFID hacking
      - » http://www.openpcd.org/OpenPCD_2_RFID_Reader_for_13.56MHz
    - Interfaces: HSU, SPI and I2C (two-wire)
    - NXP reader ASIC (can do MIFARE Crypto1)
    - Supports Card Emulation, reading and writing tags.
    - Chipset: PN532
    - Price: $60.00
      - » Or build your own! ☺
    - Webstore closed atm ☹

Introduction to NFC

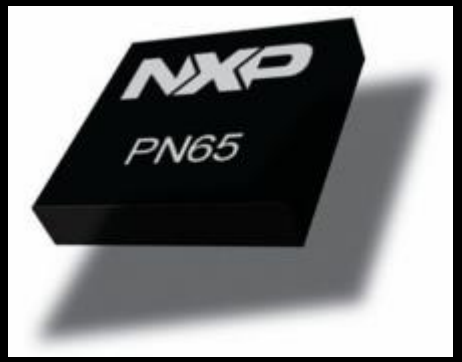
- Hardware
  - NFC Readers/Writers
    - ACR122U (Read Only)
      - **Frequency:** HF – 13.56MHz
      - **Interface:** USB
      - **Chipset:** PN53X
      - **Price:** $40.00
      - **Standards:** PC/SC, CCID
      - **Cards Supported:**
        - MIFARE, ISO 14443 A\B**,** FeliCa, ISO/IEC 18092 NFC
      - Has issues being able to abort commands and deal with timeouts.
        - acr122_usb driver corrects this to a degree

Introduction to NFC

- Toolkit
  - Readers/Writers – Popular Equipment
    - Proxmark3 (Read/Write/Playback)
      - **Frequency:** HF 13.56MHz, and LF 125kHz
      - **Interface:** USB
      - **Other:** Open/Programmable firmware
      - **Price:** $399.00 ($229 ‘naked’)
        - » $59 for HF antenna
      - **Site:** www.proxmark3.com

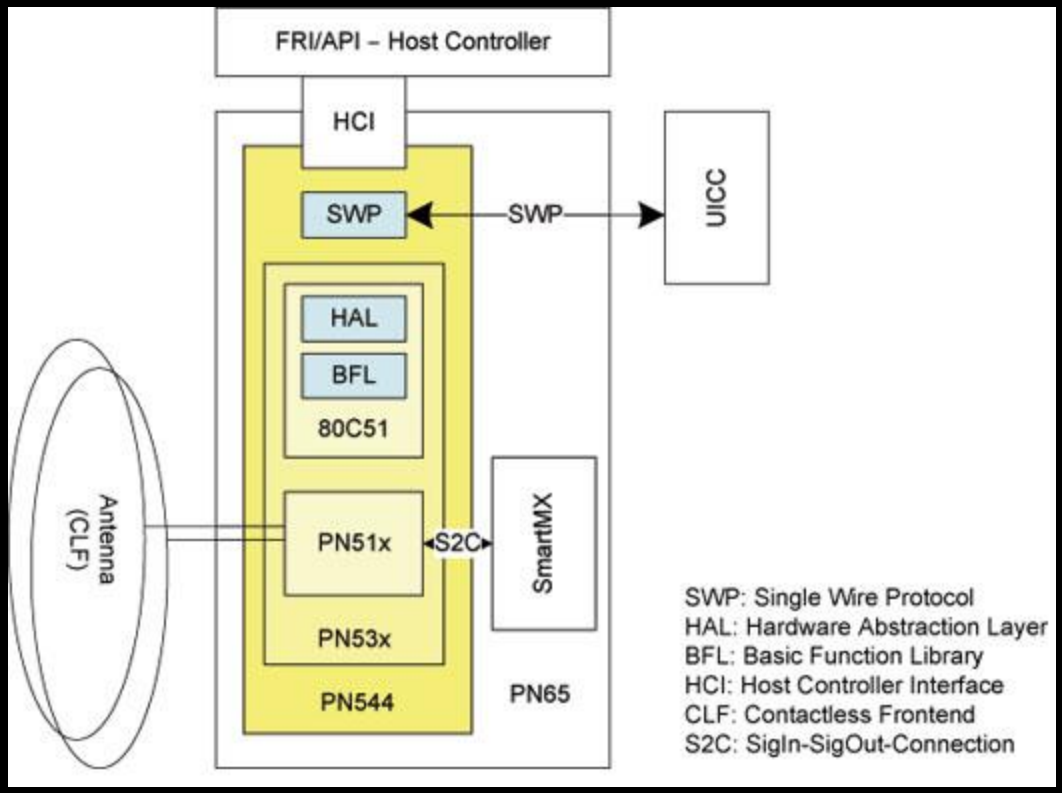
Introduction to NFC

- Hardware
  - NFC Chipsets
    - NXP PN65N
      - PN512 NFC radio
      - 80C51 MCU running the firmware for the PN512
      - The combination of the 80C51 MCU and the PN512 NFC radio is known as the PN531
      - Interface to use SIM card as the Secure Element over SWP (Single Wire Protocol)
      - Embedded P5CN072 Secure Dual Interface PKI Smart Card Controller (SmartMX)
      - NXP PN544 chip is identical except it lacks the embedded Secure Element (P5CN072)



Introduction to NFC

- Hardware
  - NFC Chipsets
    - NXP PN65N



Introduction to NFC

- Hardware
  - Secure Element (SE)
    - Tamper resistant secure microcontroller
      - Will self-destruct if tampered with (sometimes accidently)
    - Can’t utilize it without knowing the keys
      - Keys are controlled by TSM’s
    - Used primarily for mobile payments or access control systems
    - No public API on Android
    - Three Form Factors
      - UICC (SIM Card)
      - Embedded in Device
      - SD Card

Introduction to NFC

- Hardware
  - Communicating with the embedded Secure Element
    - NFC-WI (S2C) used to talk to NFC RF interface
    - Three modes of communication
      - Off
      - Wired
        - » Secure Element is visible to NFC controller as a smartcard
        - » Used by apps to communicate with the Secure Element
      - Virtual
        - » Secure Element is visible to external readers as a smartcard
        - » Used by readers to communicate with the Secure Element through the NFC contactless interface

Introduction to NFC

- Hardware
  - Communicating with the UICC Secure Element
    - UICC is connected only to the baseband processor, so all communications must go through the Radio Interface Layer (RIL)
      - AT Commands
      - Proprietary IPC interface
      - Support needs to be added to proprietary library for access
    - SWP (Single Wire Protocol)
      - Used by UICC Secure Element to communicate with NFC RF frontend
      - NFC controller must support it
    - SEEK for Android provides patches that allow for both.

Introduction to NFC

- Hardware
  - Secure Element in Mobile Devices
    - PN65N
      - Supports both UICC and Embedded Secure Elements
        - Galaxy Nexus
        - Galaxy S III
        - Nexus S
      - Integrated SmartMX chip
        - JavaCard OS
        - Global Platform Card Manager - Provides interface to install remove, and access applications on the secure element
    - PN544
      - No built-in Secure Element
      - Supports UICC SE
        - Galaxy S
        - Galaxy S II

Introduction to NFC

- Software
  - libnfc
    - Open Source C library for NFC
    - Supports:
      - ISO 14443 A/B
      - MIFARE
      - FeliCa
    - Card Emulation
    - Lots of useful utilities (nfc-*)
    - libfreefare
      - Provides API to manipulate MIFARE cards
    - Many tools require libnfc
    - http://nfc-tools.org

Introduction to NFC

- Software
  - RFIDIOt
    - Collection of Python tools and libraries for working with RFID
    - Has scripts for interacting with:
      - Mifare Classic 1k, 4k
      - Mifare Ultralight
      - ISO 14443a /b
    - Works with libnfc and PC/SC
    - https://github.com/AdamLaurie/RFIDIOt

Introduction to NFC

- NFC Attacks
  - Prior Work
    - Charlie Miller - Fuzzing NFC
    - MWR Labs - Delivering exploits over NFC
    - Collin Mulliner - All kinds of stuff
      - http://www.mulliner.org/nfc/
    - Dan Rosenberg – Multiple buffer overflows in in Linux NFC stack.
      - http://marc.info/?l=linux-kernel&m=134030878917784
    - Attacks against MIFARE encryption
      - Nicolas T. Courtois – Darkside Attack
    - Corey Benninger and Max Sobell – Cloning Mifare Ultralight cards used in transit systems
    - Bughardy and Eagle – Locking OTP in Ultralight cards

Introduction to NFC

- NFC Attacks
  - Sniffing
    - NFC does not provide encryption
      - Apps must provide their own encryption, such as SSL/TLS
    - While effective range for NFC is 1-4cm, the signal can be sniffed from a few meters away
    - Proxmark3 can intercept NFC communications using HF antenna

Introduction to NFC

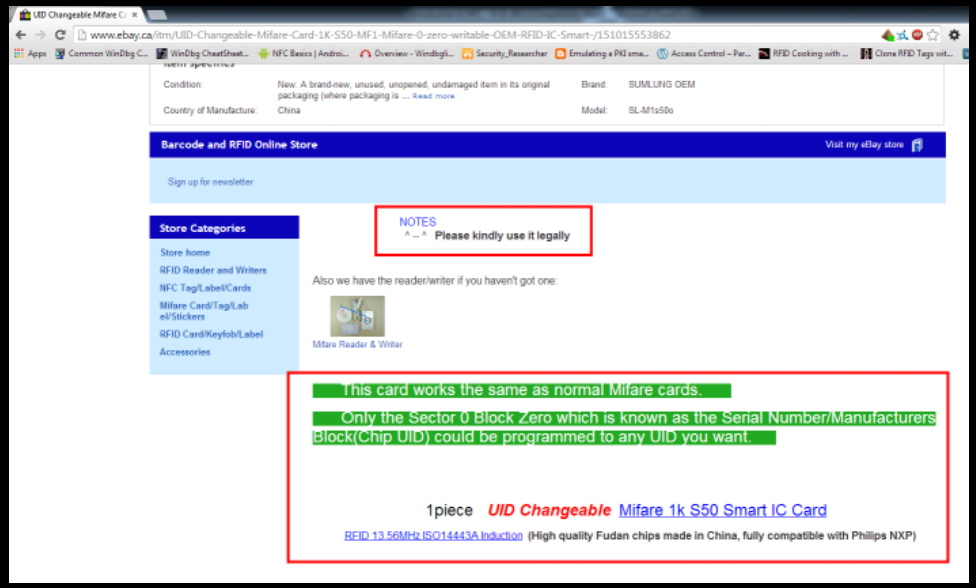
- NFC Sniffing
  - Wireshark Dissectors
    - FeliCa dissector
      - http://anonsvn.wireshark.org/viewvc/trunk/epan/dissectors/packet-rfid-felica.c
    - MIFARE dissector
      - http://anonsvn.wireshark.org/viewvc/trunk/epan/dissectors/packet-rfid-MIFARE.c
    - NXP PN532 dissector
      - http://anonsvn.wireshark.org/viewvc/trunk/epan/dissectors/packet-rfid-pn532.c
    - wireshark-nfc
      - Wireshark plugin for the LLCP libpcap file format
      - http://code.google.com/p/wireshark-nfc/

Introduction to NFC

- Attacking NFC
  - Rewriting tags
    - Mifare Ultralight
      - Used by a number of transit systems
      - 32 bit OTP (One-Time-Pad) gets set to ‘1’ after each trip.
      - Some transit systems never used the OTP
      - OTP broken at Defcon 21 by leveraging lock bytes to lock the OTP, making it impossible to write.

Introduction to NFC

- Attacking NFC
  - Cloning tags
    - Clone with PM3 or MFOC and NFC-MfClassic
    - Many access control systems use UID of card
    - UID not RO on Chinese cards, Ebay is your friend
    - Proxmark3 can replay static UID



Introduction to NFC

- Attacking NFC
  - Card Reading
    - EMV chip on MasterCard Paypass and Visa PayWave stores same info as magstripe.
      - Can be read just by following the spec
        - http://www.freepatentsonline.com/y2010/0108758.htm
        - http://www.emvco.com/specifications.aspx
    - Can use Pwnpass.py and Vivopay reader or nfc-paycardreader app (or Omnikey Cardman 5231)
      - Can read:
        - Card Number
        - Name (first, last)
        - Expiration Date
    - Android 4.4 provides EMV card emulation

Introduction to NFC

- Attacking NFC
  - Breaking Encryption
    - MIFARE
      - Developed by NXP (formally Philips)
      - Most widely installed contactless smartcard
      - A number of different variants exist for different purposes:
        - MIFARE Classic
        - Ultralight
        - Ultralight C
        - MIFARE Plus
        - DESFire
        - DESFire EV1
        - SmartMX

Introduction to NFC

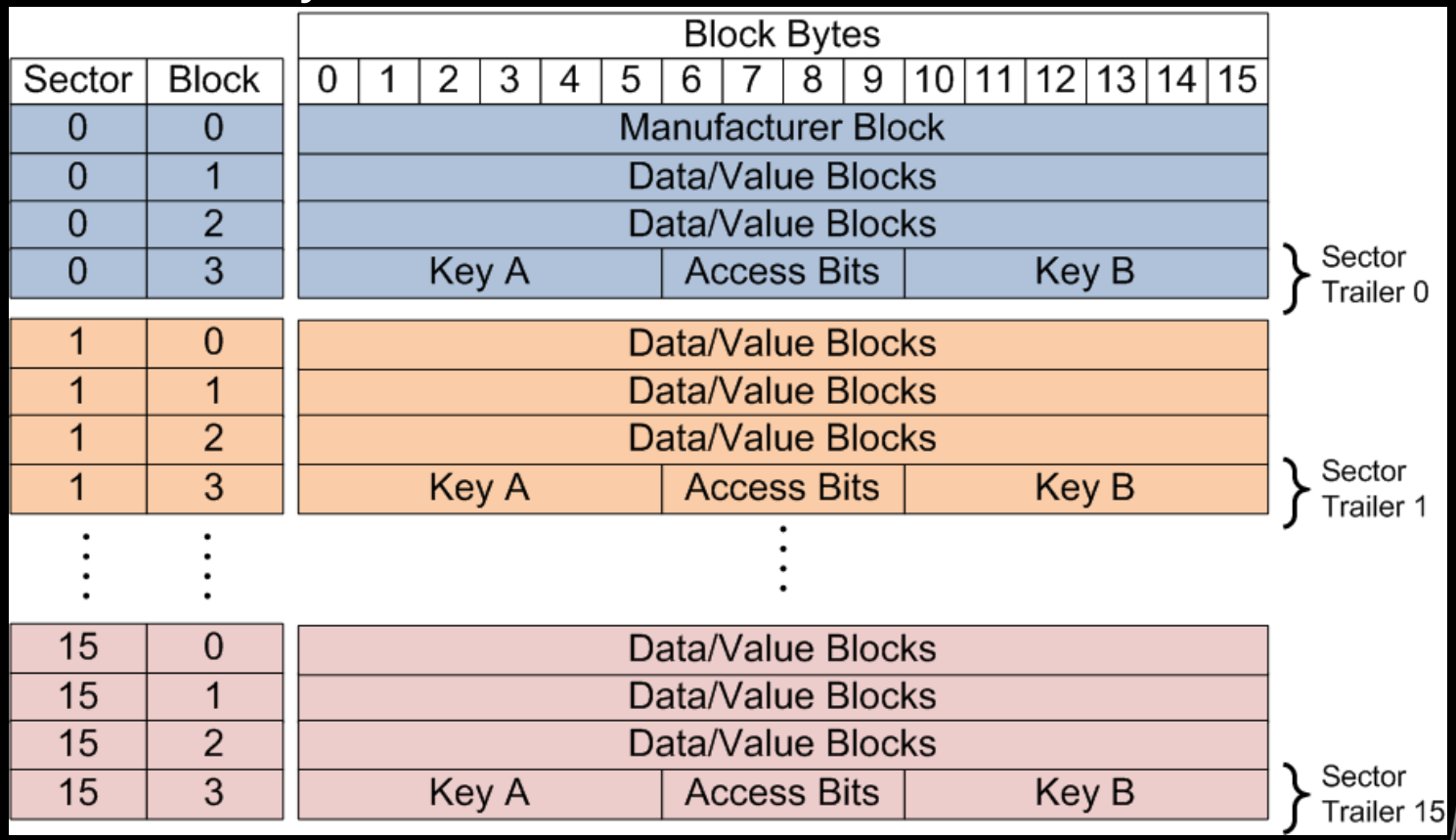
- Attacking NFC
  - Breaking Encryption
    - MIFARE Classic
      - Popular with public transit systems
      - Operates at 13.56MHz
      - ISO 14443-3 compliant
        - » ISO 14443-4 defines high level protocol, NXP did this themselves
      - Crypto-1 (NXP proprietary crypto algorithm)

Introduction to NFC

- Breaking Encryption
  - MIFARE Classic
    - Memory Structure
      - **Blocks:** 16-bytes of memory, can be either:
        - » Data block – arbitrary data, usually used in access control systems
        - » Value block – stores signed value of credit used, used in electronic wallet systems
      - **Sectors:** 4 Blocks
        - » **Sector Trailer:** Last block of the sector, contains keys and access conditions for sector
        - » Each sector is encrypted with its own key
    - Protocol Commands for Memory
      - Read, Write, Decrement, Increment, Restore, or Transfer

Introduction to NFC

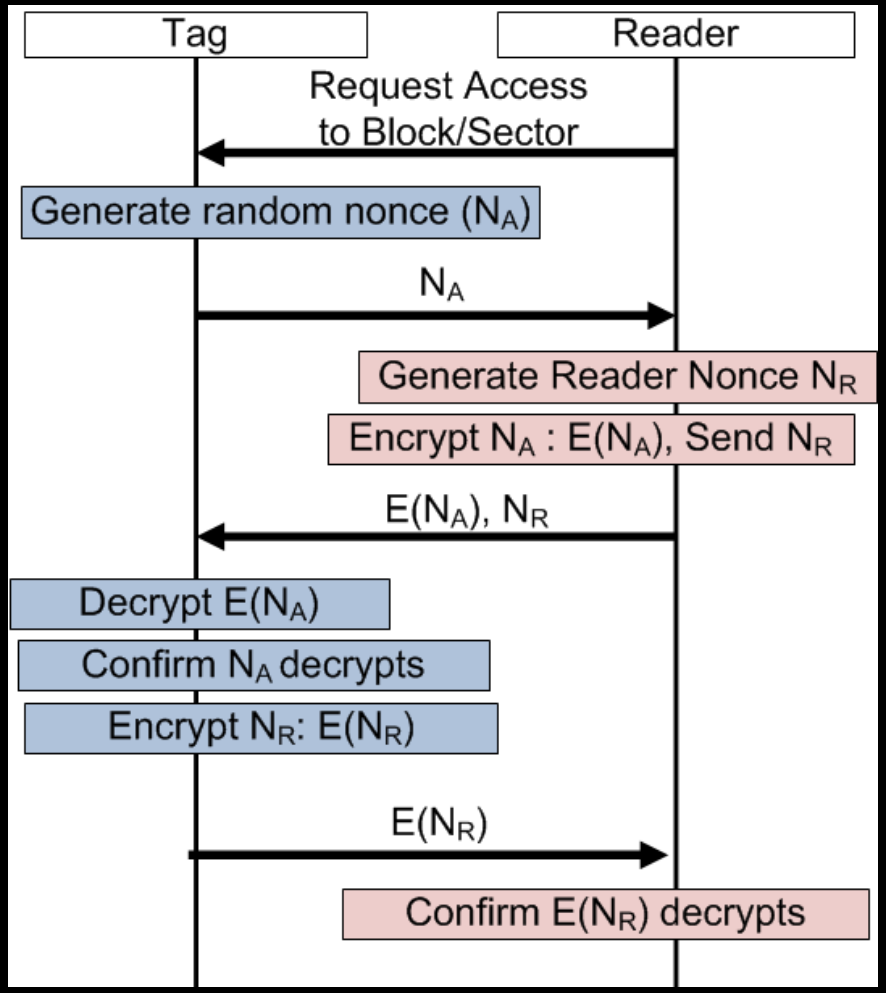
- Breaking Encryption
  - MIFARE Classic
    - Memory Structure



Sector	Block	Block Bytes 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	Manufacturer Block																
0	1	Data/Value Blocks																
0	2	Data/Value Blocks																
0	3	Key A						Access Bits				Key B						Sector Trailer 0
1	0	Data/Value Blocks																
1	1	Data/Value Blocks																
1	2	Data/Value Blocks																
1	3	Key A						Access Bits				Key B						Sector Trailer 1
⋮	⋮	⋮																
15	0	Data/Value Blocks																
15	1	Data/Value Blocks																
15	2	Data/Value Blocks																
15	3	Key A						Access Bits				Key B						Sector Trailer 15

Introduction to NFC

- Breaking Encryption
  - MIFARE Classic
    - Authentication



Introduction to NFC

- Breaking Encryption
  - MIFARE Classic
    - Crypto -1 Flaws
      - Low entropy in PRNG (16 bits)
      - Timing Attack on the16b Tag/Reader Nonce
        - » Nonce is created ONLY between the time it takes for the reader to power the tag and ask for challenge
      - Parity Keystream Leakage
        - » Known parity error messages are returned encrypted
        - » Parity bit and first bit of next plaintext byte encrypted with same keystream bit
      - Cryptographic Cipher Weaknesses
        - » Only Odd Bits Used to Generate Keystream
        - » The Linear Feedback Shift Register (LFSR) can be rolled back to deduce the key if valid keystream is known

Introduction to NFC

- MIFARE Classic
  - Attack Tools
    - MFOC (MIFARE Classic Offline Cracker)
      - Implements the ‘offline nested’ attack
      - Built on libnfc
      - Can recover keys from MIFARE Classic cards
      - Requires one known key
        - » Many cards have a least one block encrypted with default keys
        - » http://code.google.com/p/mfcuk/wiki/MifareClassicDefaultKeys
      - http://code.google.com/p/mfoc/

```
# ./mfoc -O output.mfd
```

Introduction to NFC

- MIFARE Classic
  - Attack Tools
    - MFCUK
      - Implements the ‘dark side’ attack
      - Does not need to know any keys
      - Built on libnfc and Crapto1 libraries
        - » http://code.google.com/p/crapto1/
      - Integrated into the Proxmark3 firmware
      - http://code.google.com/p/mfcuk/

```
# ./mfcuk –R 1 –C –v 1
```

      - -R 1 (Request first sector_
      - -C (Connect to card reader)
      - -v (Verbosity level one)

Introduction to NFC

- Reference
  - Recommended Reading
    - BlackBerry® Developer Resource Center
      - http://supportforums.blackberry.com/t5/Java-Development/NFC-Article-and-Code-Index/ta-p/1538775
    - Android Developer Guides
      - https://developer.android.com/guide/topics/connectivity/nfc/index.html]
    - NFC Forum Specifications
      - http://www.nfc-forum.org/specs/spec_license
        - » Requires agreeing to license
    - Android Explorations
      - http://nelenkov.blogspot.com/2012/08/accessing-embedded-secure-element-in.html

Introduction to NFC