© Michael Roland
www.mroland.at

Secure Element APIs and Practical Attacks on Secure Element-enabled Mobile Devices

Michael Roland
University of Applied Sciences Upper Austria, Hagenberg, Austria

WIMA 2012 – NFC Research Track
11 April 2012, Monaco

This work is part of the project "4EMOBILITY" within the EU program "Regionale Wettbewerbsfähigkeit OÖ 2007–2013 (Regio 13)" funded by the European regional development fund (ERDF) and the Province of Upper Austria (Land Oberösterreich).

© Michael Roland
www.mroland.at

Outline

- Card Emulation + Secure Element + Mobile Phone
- APIs for Access to the Secure Element
- Mobile Phones as Target for Attacks
- New Attack Scenarios

© Michael Roland
www.mroland.at

Card Emulation

- One of three operating modes of NFC devices
- Interaction with existing RFID reader/writer infrastructure
  - E.g. POS terminals, access control readers
- Implementation of card emulation mode
  - Dedicated smartcard chip (secure element)
    - Embedded secure element
    - UICC (“SIM card”)
    - (micro) SD card
  - Software card emulation
    - No secure element
    - Communication is handled by software on the application processor

Secure Element: Current View on Security

- Secure element is as secure as a regular (contactless) smartcard
  - Same security features (Secure storage, Secure execution environment, Hardware-based cryptography, Certified high security standard)
  - Same weaknesses
- Main weakness: Relay attack
  - Cannot be prevented by application-layer cryptographic protocols
  - Timing requirements by ISO 14443 are too loose to prevent relay over longer channels
  - Possible countermeasures:
    - Shielding of contactless interface
    - Secondary authentication (PIN codes …)
    - Distance bounding protocols (require additional fast communication channel; not implemented on current smartcards)
- **BUT: All known relay attacks require physical proximity (< 1 meter) between the attacker and the smartcard!**

Secure Element in a Mobile Phone

- Secure element adds security features to a mobile phone
- NOW: Mobile phone is **not** considered a security risk for the secure element
- **BUT: Mobile phone environment is a significant part of secure element security**
  - Potential host for malicious software
  - (Global) wireless connectivity (GSM, UMTS, WiFi, Bluetooth …)

APIs for Access to the Secure Element

- Security and Trust Services API (JSR 177) → Java ME
  - APDU-based connection to one specific smartcard applet
    - no application selection, no logical channel management
  - Access is only granted to applications with trusted signatures
    + user confirmation
  - Optional: fine-grained access control
    - Applications' signatures must chain back to root certificates provided by SE
    - Secure element provides access control lists (ACLs)
      - Per secure element and per applet policies
      - Access based on application's security domain and on APDU header

APIs for Access to the Secure Element

- BlackBerry 7 API[1]
  - Interface based on JSR 177
    - Additional API to handle multiple secure elements
  - Access is only granted to applications with trusted signatures
    - Special code-signing certificates need to be obtained from RIM (registration required)

[1] http://www.blackberry.com/developers/docs/7.0.0api/net/rim/device/api/io/nfc/se/package-summary.html

APIs for Access to the Secure Element

- Android
  - NFC-Extras API introduced in Android 2.3.4 (com.android.nfc_extras)
    - Not included in public SDK
  - Interfaces for APDU-based secure element access and for activation of card emulation
    - Connection to whole smart card and not limited to a single smartcard applet
  - Access control:
    - Android 2.3.4: NFC permission required
      - Any application with access to NFC has access to the SE
    - Android 2.3.5+: Applications require special permission
      - com.android.nfc.permission.NFCEE_ADMIN, only granted to applications signed with same key as NFC service (**effectively limited to manufacturer**)
    - Android 4.0+: Permissions defined in an XML file
      - XML file contains list of allowed application certificates (can only be modified with **OTA updates** or **root access**)

APIs for Access to the Secure Element

- SEEK for Android[1]
  - Project aims at bringing a standardized smartcard API to Android
  - Interface compliant to Open Mobile API
  - Fine-grained access control similar to JSR 177
    - Access based on application certificate, applet AID and APDU header information
    - Access policy is stored on secure element
    - Access control is enforced by the smartcard service on the application processor

[1] http://code.google.com/p/seek-for-android/

© Michael Roland
www.mroland.at

Comparison of APIs

- Common to all APIs:
  - Some form of access control
  - Access control is always enforced by the operating system **on the application processor**

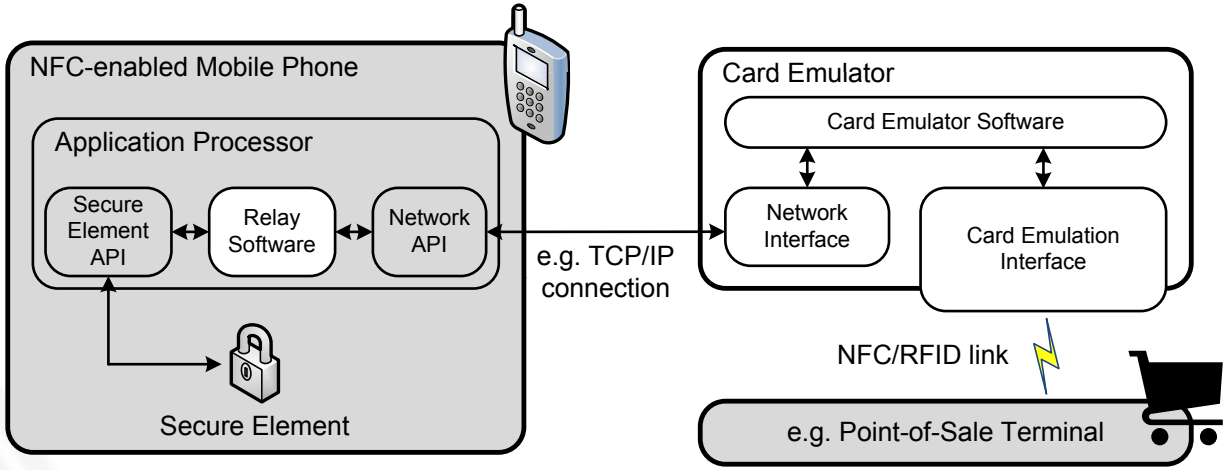
⇒ **APIs are designed to ultimately trust the operating system and the underlying mobile phone hardware**

⇒ **Secure Element trusts the operating system's access control decision**

Mobile Phone as Target for Attacks

- Threat: Malicious software/privilege escalation exploits
  - Android: Continuous history of privilege escalation exploits
    - mempodroid, Levitator, zergRush, GingerBreak, ZimperLich, KillingInTheName, RageAgainstTheCage, Exploid …
    - Vulnerabilities are fixed quite fast (months), but roll-out of patches takes significantly longer (many devices still don't run the latest firmware version)
- Threat: User
  - Jail breaking / Rooting
    - Security measures are intentionally circumvented by the user
    - Gain "improved" control over device or bypass DRM
    - **Not** limited to experienced users!
    - Elevated privileges may be used by malicious applications!
  - Carelessness
    - Apps are installed without review of requested permissions
    - Even dangerous combinations of privileges accepted by users

New Attack Scenario: Relay Attack



- Virtual pickpocketing without physical proximity to the mobile phone
  - Attack only requires an application on victim's mobile phone
  - Application accesses the secure element and relays APDU commands/responses over a network interface (GSM, UMTS, WiFi …)
  - Attackers can use the victims' secure elements as if they were in physical possession of them
  - Application may access additional resources (address book, key pad …)
- Timing constraints for the relay channel: none!
  - ISO 14443 has no timing requirements on the APDU layer
  - Payment protocols (defined by EMV) don't enforce any timing requirements either

New Attack Scenario: Denial of Service

- GlobalPlatform card management of many secure elements contains security feature:
  - After 10 successive authentication failures: card is put into TERMINATED state
    - Final (irreversible) state of GlobalPlatform card life cycle
    - Once in this state, installed applets continue to function, but card management (installation, removal … of applets) is no longer possible!

    ⇒ Secure Element is unusable for further card emulation applications
- What's necessary for a successful attack?
  - Access to the secure element
  - Three APDU commands have to be executed for one authentication attempt:
    - SELECT [Issuer Security Domain]
    - INITIALIZE UPDATE
    - EXTERNAL AUTHENTICATE
- Attack is possible on
  - Nokia 6131, Nokia 6212
  - Nexus S (with Android 2.3.4)
  - Nexus S/Galaxy Nexus (with Android 2.3.5+, if the application is able to gain root privileges)
  - …

Conclusion

- Attacks on contactless smartcards are well-known and can be prevented by physical measures (e.g. shielding)
- Adding a secure element to a mobile phone opens a new attack vector that has not been considered before
  - Attacks can be achieved **with pure software** on the victim's device
    - Denial of Service
    - Relay Attack
- Protection of secure element APIs varies widely
  - For all APIs: access control is managed by the operating system of the mobile phone
  - The secure element always trusts the operating system's access control decisions

© Michael Roland
www.mroland.at

NFC Congress 2012 Hagenberg

September 11th – 12th, 2012
Hagenberg, Austria

http://congress.nfc-research.at/

Thank You!

Michael Roland
Research Associate, NFC Research Lab Hagenberg
University of Applied Sciences Upper Austria, Hagenberg, Austria

michael.roland (at) fh-hagenberg.at

This work is part of the project "4EMOBILITY" within the EU program "Regionale Wettbewerbsfähigkeit OÖ 2007–2013 (Regio 13)" funded by the European regional development fund (ERDF) and the Province of Upper Austria (Land Oberösterreich).