Relay Attacks on Secure Element-enabled Mobile Devices

Virtual Pickpocketing Revisited

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

SEC2012 – IFIP International Information Security and Privacy Conference
4 June 2012, Heraklion, Crete, Greece

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

LAND OBERÖSTERREICH

Outline

- Introduction
  - Card Emulation / Secure Element / Mobile Phone
- Mobile Phone as Target for Attacks
- Software-based Relay Attack
  - Idea
  - Card Emulators
  - Proof of Concept
  - Measurement results
- Conclusion

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 communication protocol (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
- Mobile phone adds:
  - Over-the-Air management capabilities to the secure element
  - Applications can be added/removed throughout the secure element's whole life-cycle
- Current view:
  - 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 …)

  ⇒ **Secure element is never isolated from its surrounding world**

Android's Secure Element API

- 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** or **root access**)
  - 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**)

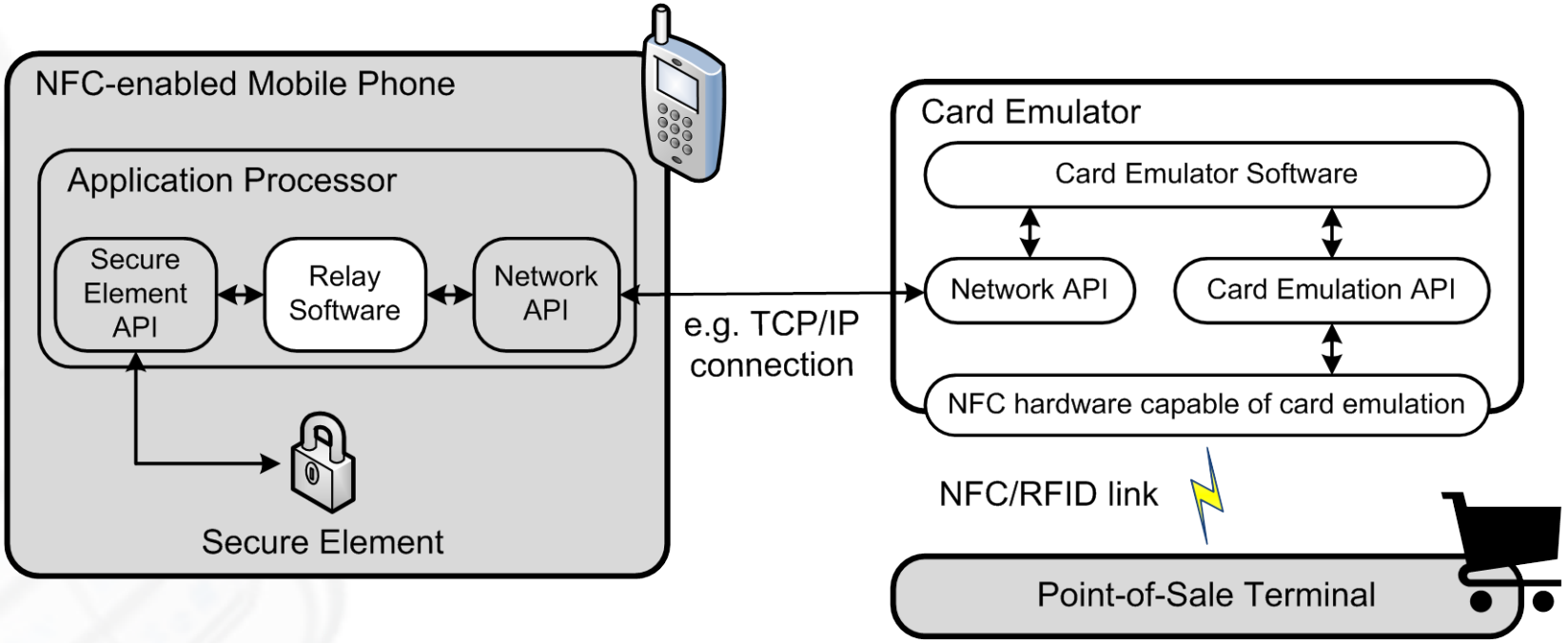
⇒ Access control is enforced by the operating system **on the application processor**

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

Mobile Phone as Target for Attacks

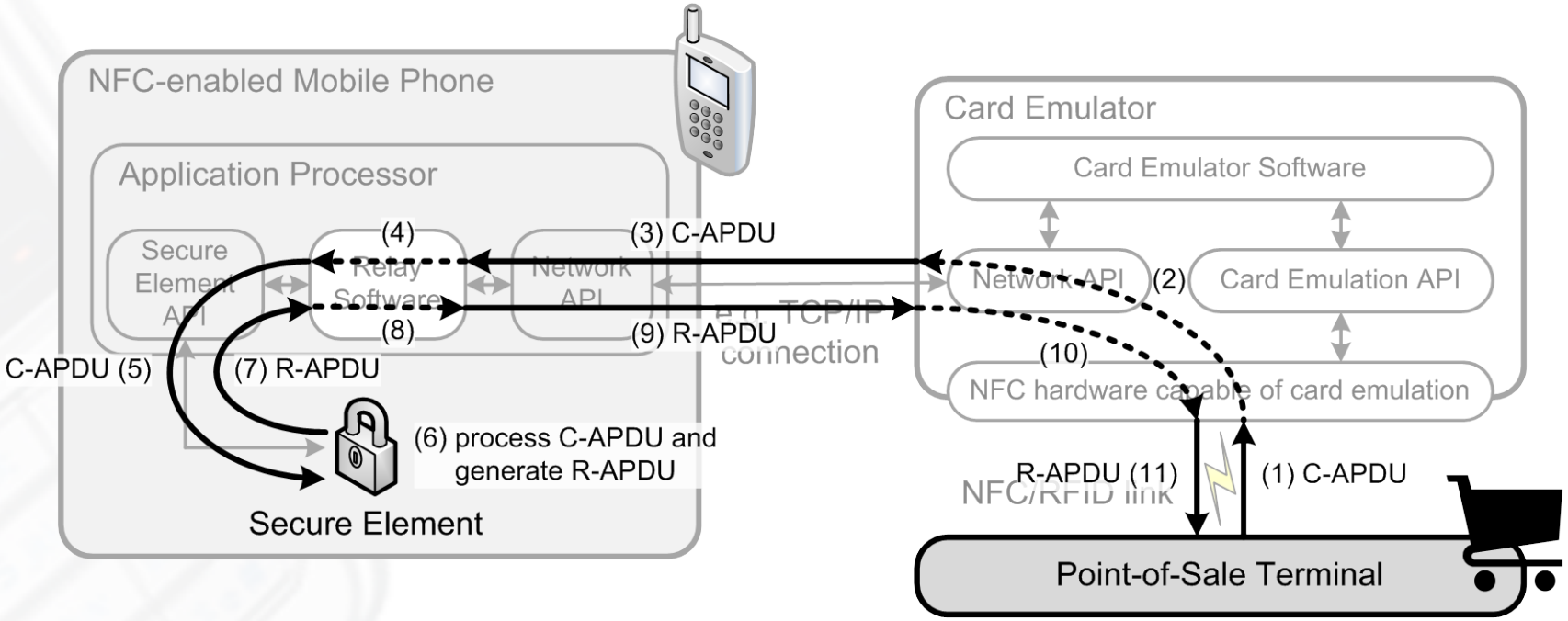
- Threat: Malicious software/privilege escalation exploits
  - Android: Continuous history of privilege escalation exploits
    - mempodroid, Levitator, zergRush, GingerBreak, ZimperLich, KillingInTheName, RageAgainstTheCage, Exploid …
    - Vulnerabilities are fixed quite fast (months), but roll-out of patches takes significantly longer or does not happen at all (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 are accepted by users

New: Software-based 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
  - Relay application may access additional resources (address book, key pad …)

New: Software-based Relay Attack



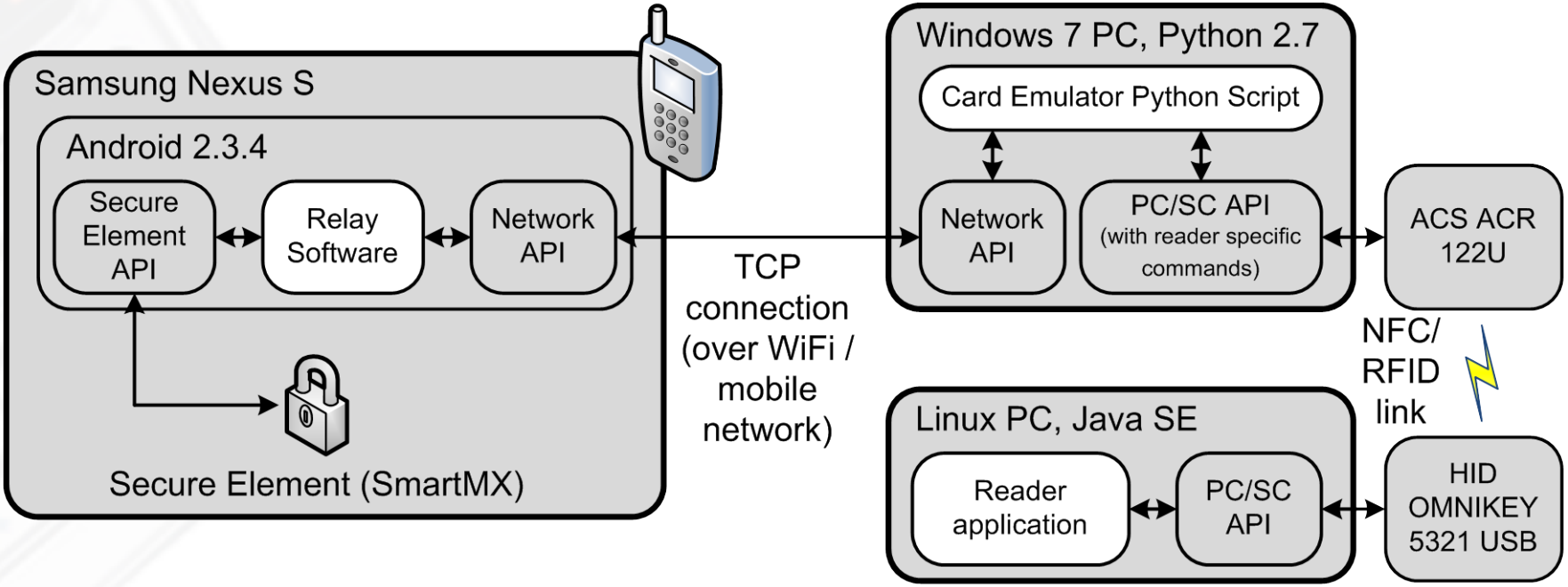
- Virtual pickpocketing without physical proximity to the mobile phone
  - Attack only requires an application on victim's mobile phone
  - Application accesses the secure element and relays APDU commands/responses over a network interface (GSM, UMTS, WiFi …)
  - Attackers can use the victims' secure elements as if they were in physical possession of them
  - Relay application may access additional resources (address book, key pad …)

Card Emulator

- Building a new device from scratch:
  - Full control over whole design process
  - Any (inconspicuous looking) shape possible
  - All parameters of the RFID protocol stack can be controlled (e.g. emulation of any UID value)
  - Highest design cost & effort
- Using a ready-made RFID card emulation device: (e.g. Proxmark)
  - All parameters of the RFID protocol stack can be controlled (e.g. emulation of any UID value)
  - Reduced design cost & effort
  - Fixed shape
  - Additional hardware for network interface necessary
- Using an NFC reader with card emulation support: (e.g. ACR 122U)
  - A PC is necessary to control the NFC reader and for network communication
  - ACR 122U restricts some protocol parameters:
    - Only ISO/IEC 14443 Type A protocol can be emulated
    - UID restricted to random UID range (i.e. UID must start with 0x08)

  → *This is sufficient for many applications*
- Using an NFC-enabled mobile phone
  - BlackBerry NFC phones support software card emulation
    - ISO/IEC 14443 Type A and Type B protocol
    - Only random UIDs are supported (auto-generated by the firmware)
  - Nexus S & Galaxy Nexus support software card emulation through CyanogenMod 9 after-market firmware
  - Mobile phone has expected form-factor for NFC contactless transactions
  - Mobile phone has the same network interfaces as the device under attack

Proof of concept: Test setup



Test setup: Limitations

- At the time of our research:
  - Mobile phone‘s secure element did not contain an actual application!
- Instead:
  - Access the GlobalPlatform card manager application (OPEN/ISD)
  - Custom reader application
  - Tested APDUs:
    - SELECT card manager by AID
      - Command: 00A4040008A000000003000000 (13 bytes)
      - Response: File control information template (105 bytes)
    - GET_DATA object '65'
      - Command: 00CA006500 (5 bytes)
      - Response: Reference data not found error (2 bytes)
    - GET_DATA object '66'
      - Command: 00CA006600 (5 bytes)
      - Response: Card data/security domain management data (78 Byte)

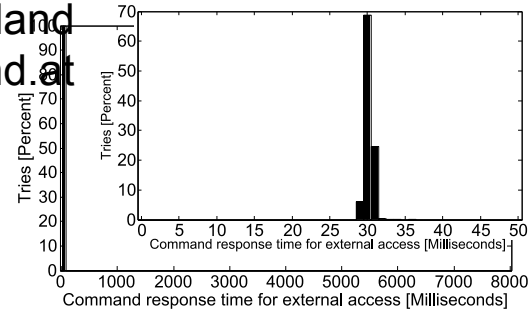
Comparison of different access scenarios

- 4 scenarios:
  1. Direct, external access through the phone‘s contactless interface
  2. Direct, internal access through the secure element API (measured by an app on the device)
  3. Access through the relay system using a WiFi link
  4. Access through the relay system using the cellular network/Internet
- For every scenario:
  - Measurement of the command-response delay at the reader side
  - Test with 5000 repetitions
    - Scenario 1, 3 & 4:
      - Card emulator/phone was removed from the reader‘s RF field between every repetition (automated with test robot)
    - Scenario 2:
      - Connection to the secure element was re-established using the API‘s close() and open() methods

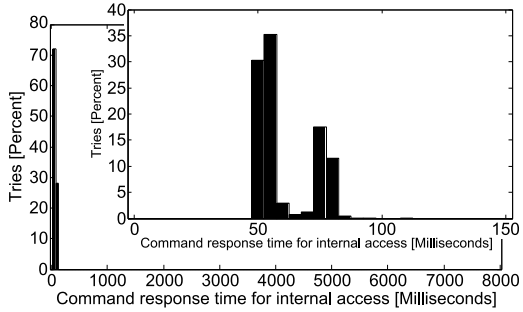
Measurement results

- Command-response delay for SELECT APDU
  - Command: 13 bytes / Response: 105 bytes
  - Scenario 1: (direct, external access)
    - ~30 ms
  - Scenario 2: (direct, internal access)
    - ~50 to 80 ms
    - Significantly slower than external access!
  - Scenario 3: (relay over WiFi)
    - ~190 to 260 ms
    - WiFi adds about 100 to 210 ms of delay
  - Scenario 4: (relay over cellular network/Internet)
    - > 200 ms, significant peak at 300 ms
    - ~45% of measured delays below 1 second
    - ~80% of measured delays below 4 seconds
    - ~97% of measured delays below 10 seconds
- Similar results for all three APDU commands
  - Results only differ in delays due to command and response lengths
- Usability of relay attack
  - ISO 14443 has no strict timeout requirements
    - Max. timeout of 4.9 seconds can be extended through “Waiting Time Extension”
  - EMV has no timeout requirements for credit card terminals
    - EMV defines only timings for maintaining user experience
    - Some terminals interrupt a transaction after several seconds
  - Payment transactions in the order of 20 to 30 seconds will not raise any suspicions (especially while the technology is new)

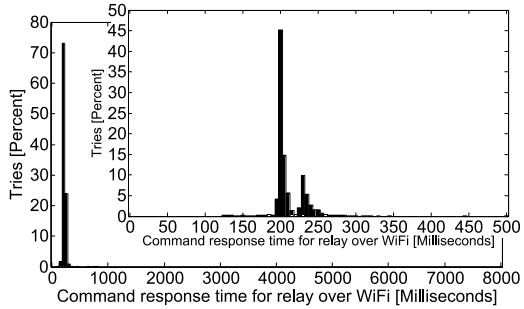
1.



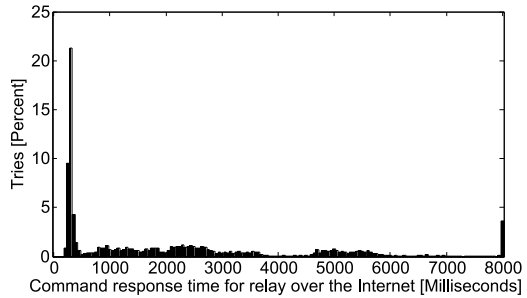
2.



3.



4.



Conclusion

- New attack: Software-based Relay Attack
  - Access to a secure element without physical proximity
  - Software on victim's device is sufficient
- Secure Element delegates access control enforcement to insecure component (application processor)
  - An app only needs to bypass the operating system's access control mechanisms
- Measurement's show that communication can even be relayed over long distances through the Internet
  - An attacker could create a world-wide network of infected mobile phones that could later be used for payment, …

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 Weltbewerbsfähigkeit OÖ 2007–2013 (Regio 13)" funded by the European regional development fund (ERDF) and the Province of Upper Austria (Land Oberösterreich).