IT-Architekturen und Sicherheitskonzepte für mobile NFC-Geräte

Michael Roland
FH Oberösterreich, Campus Hagenberg, Austria

17. April 2012, JKU Linz

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).

Inhalt

- Überblick
  - Ziele der Arbeit
  - Was ist NFC
  - Sicherheitsrisiken bei NFC-Anwendungen
- Bisherige Arbeiten & Ergebnisse
- Relay-Attacke gegen Google Wallet
  - Ablauf einer Payment-Transaktion
  - Demo
  - Lösungsansätze

Ziele der Arbeit

- Evaluierung vorhandener Sicherheitskonzepte für Mobiltelefonsysteme
- Evaluierung von Sicherheitsrisiken für NFC-Anwendungen
- Entwicklung von Lösungsansätzen für diese Sicherheitsprobleme

Was ist NFC?

- kontaktlose Übertragungstechnologie (Basis: RFID & Smartcards)
- NFC-Gerät kann
  - kontaktlose Chipkarten/Tags lesen
  - mit anderen NFC-Geräten kommunizieren
  - selbst als kontaktlose Chipkarte verwendet werden
- typische Anwendungen:
  - Payment, Ticketing, Loyalty
    - NFC-Mobiltelefon ersetzt vorhandene (kontaktlose) Chipkarten
  - Smart Poster
    - Zugriff auf interaktive Inhalte durch einfache Berührung eines Objekts mit einem NFC-Gerät/Mobiltelefon
  - Enabler für andere Kommunikationstechnologien
    - Bluetooth, WiFi, Wireless USB

Sicherheitsrisiken bei NFC-Anwendungen

- Verwendung manipulierter NFC-Tags
  - Phishing
  - ungewollte Nutzung von teuren Mehrwertdiensten (SMS, …)
  - Einschleusen von Schadcode
- „virtueller Taschendieb“
  - unbemerkter Zugriff auf das Secure Element (SE)
  - Abbuchung von Kleinstbeträgen
  - Nutzung von Tickets, Berechtigungen bei Zutrittssystemen (z.B. durch Relay-Attacke)
- Angriffe durch „fremde“ Mobiltelefonapplikationen
  - Phishing oder Abhören von PINs, Kennwörtern, …
  - Verfälschung von Statusinformationen (z.B. Anzeige eines falschen Geldbetrags bei SE-Transaktionen)

Bisherige Arbeiten & Ergebnisse

- Signatur-Spezifikation für Daten auf NFC-Tags
  - Mangelhafte Umsetzung!
  - Datensemantik im Nachhinein veränderbar ohne Signatur zu verändern!
  - Keine Kriterien für das Vertrauen in signierte Daten definiert!
  - Security WG des NFC Forum hat mit Überarbeitung der Spezifikation begonnen
    → meine Vorschläge werden derzeit in die Spezifikation eingearbeitet
- Angriffsszenarien gegen das Secure Element
  - Secure Element ist so sicher wie andere kontaktlose Smartcards
    - sicherer Datenspeicher, sicheres Execution Environment, Kryptographie in Hardware, zertifiziert nach hohen Standards (Common Criteria)
    - gleiche Schwachstellen: Relay-Angriff *aus kurzen Distanzen*
  - Einbettung in Mobiltelefon bringt zusätzliche, jedoch *noch unbeachtete* Angriffsfläche!
    - Apps können auf Secure Element zugreifen
      - Entweder freier Zugriff oder über Privilege Escalation
    - Secure Element ist über Mobiltelefon permanent mit globalen Netzwerken verbunden

Secure Element APIs

- Security and Trust Services API (JSR 177)
  - Zugriff nur durch signierte, vertrauenswürdige Anwendungen
  - Optional: Access Control Lists im Secure Element regeln den Zugriff (Einhaltung wird vom Mobiltelefonbetriebssystem kontrolliert)
- Nokia‘s Extensions to Contactless Communication API (JSR 257)
  - Zugriff nur durch signierte, vertrauenswürdige Anwendungen
- BlackBerry 7
  - Zugriff nur durch signierte, vertrauenswürdige Anwendungen
- Android
  - API versteckt, aber von jedem verwendbar
  - Android 2.3.4: Jede Anwendung mit Berechtigung für NFC kann auch auf SE zugreifen
  - Ab Android 2.3.5: Spezielle Berechtigung notwendig, die nur an Anwendungen vergeben wird, die vom selben Herausgeber wie der NFC-Service signiert wurden (= Google)
  - Ab Android 4.0: Berechtigungsvergabe über XML-Datei (Liste mit Zertifikaten)
- SEEK for Android
  - Zugriffsschutz vergleichbar mit dem von JSR 177
- Gemeinsamkeit: Zugriff wird immer vom Mobiltelefonbetriebssystem geregelt!
  - **Es wird vorausgesetzt, dass das Betriebssystem und die Mobiltelefonhardware vertrauenswürdig sind**

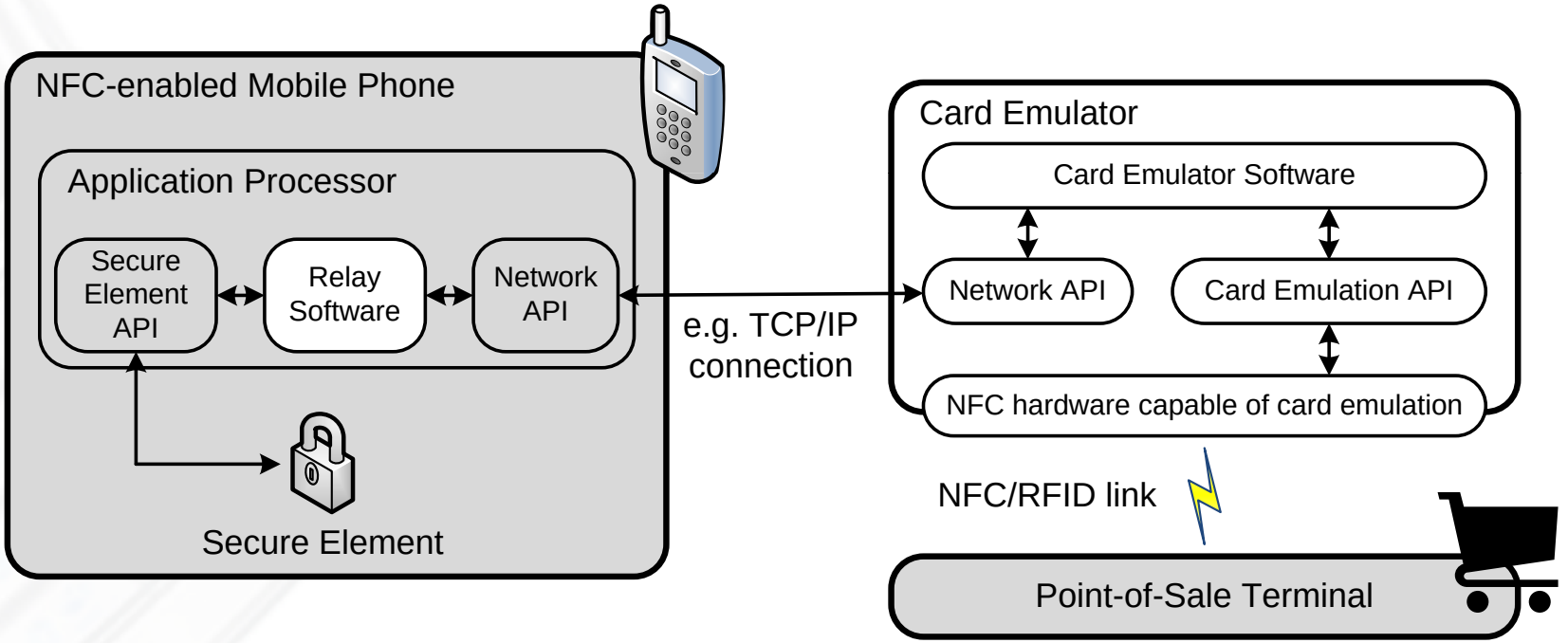
Schwachstelle: Mobiltelefon

- Gerade Smartphones sind derzeit beliebtes Ziel für Angriffe (vgl. Kaspersky Lab Monthly Malware Statistic)
- Android:
  - Es existieren zahlreiche Schwachstellen zur Privilege-Escalation
  - Bekannte Schwachstellen werden innerhalb weniger Monate behoben
    - Allerdings gibt es große Verzögerungen bis neue Betriebssystemversion auch wirklich auf Endgeräten installiert wird
  - Ständig werden neue Schwachstellen gefunden
    - Jänner 2012: mempodroid (funktioniert ab Android 4.0)
    - Oktober 2011: Levitator (funktioniert bis Android 2.3.5)
    - Oktober 2011: zergRush (funktioniert bis Android 2.3.3)
    - April 2011: GingerBreak (funktioniert bis Android 2.3.3)
    - davor: ZimperLich, KillingInTheName, RageAgainstTheCage, Exploid …
- Weiteres Problem: „Rooting“, „Jailbreaking“
  - (bewusstes) Aushebeln von Sicherheitsmaßnahmen durch den Benutzer
  - Problem: auch Schadsoftware profitiert davon!

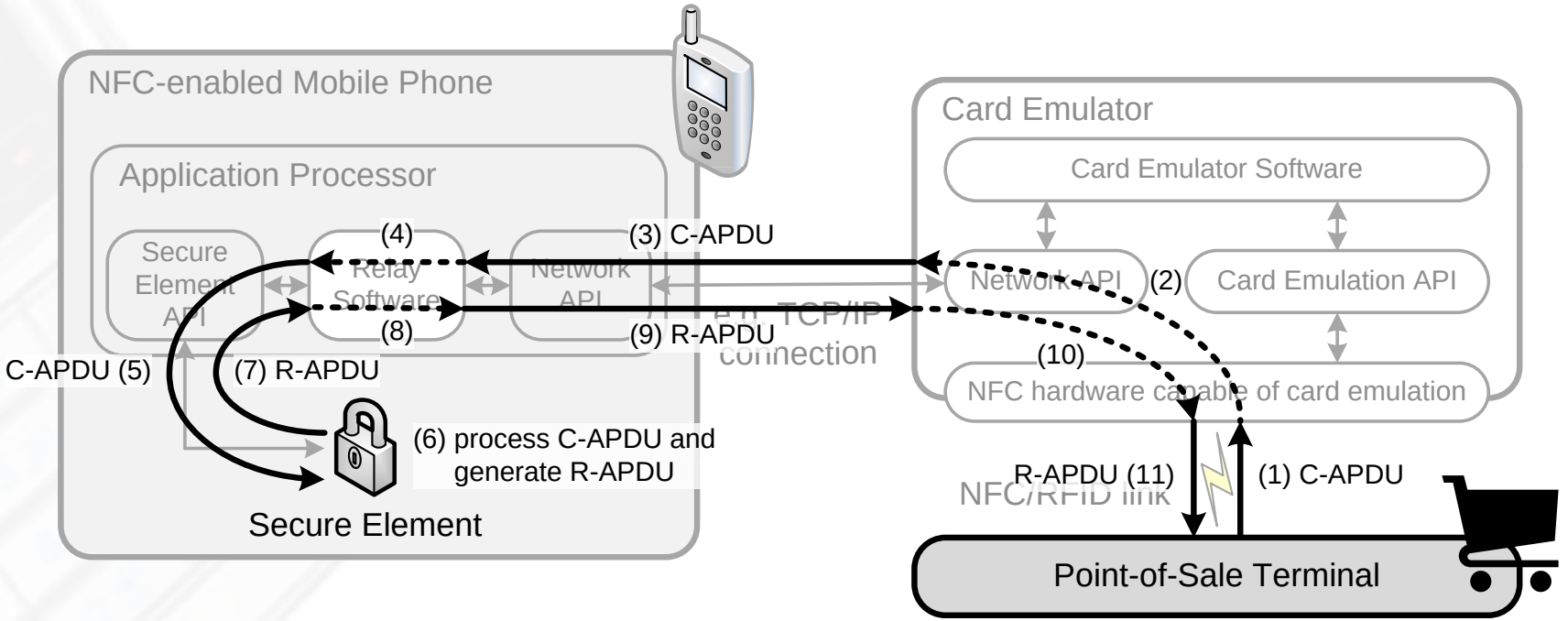
Angriffsplattform: Mobiltelefon

- Zugriff auf Secure Element
  - Worst-case: Anwendung kann beliebige Befehle an das SE senden
  - Beispiel: Denial-of-Service
    - Card Manager (Komponente zum Installieren/Löschen von Anwendungen im SE) hat besonderen Zugriffsschutz:
      - Nach 10 fehlgeschlagenen Authentisierungsversuchen: irreversible Deaktivierung (installierte Anwendungen bleiben weiter verfügbar)
      - Card-Emulation ist dann nur mehr eingeschränkt nutzbar!
- Zugang zu Mobilfunknetz, WiFi, Bluetooth, **Internet**
  - Beispiel: Relay-Attacke
    - Keine unmittelbare Nähe des Angreifers zum Telefon mehr notwendig
    - App kann Kommunikation zwischen einem Chipkartenemulator und dem Secure Element über das Internet tunneln
- Zugriff auf Adressbuch, Notizen, Tastatureingaben …
  - App könnte Mobiltelefon nach PINs, … durchsuchen (werden oft im Adressbuch gespeichert)
  - Keylogger könnte PIN/Kennwort-Eingaben ausspähen
    - Für Android existieren bereits Keylogger (vgl. Höbarth)

Angriffsszenario: Relay-Attacke



Angriffsszenario: Relay-Attacke



Card Emulator

- NFC-Reader
  - Einige NFC-Reader (z.B. ACR 122U) unterstützen Software Card Emulation
  - PC zur Steuerung des NFC-Readers und für Netzwerkkommunikation notwendig
  - Oft sind nicht alle Protokollparameter frei wählbar
    - ACR 122U erlaubt z.B. nur UIDs die mit '08' beginnen (Random UID)
    - ACR 122U emuliert nur ISO/IEC 14443 Type A

    → Für Kreditkarten ist das ausreichend!
- NFC-Mobiltelefon
  - Aktueller Stand
    - BlackBerry 7 API unterstützt Software Card Emulation
    - CyanogenMod für Nexus S und Galaxy Nexus bekommt Software Card Emulation (Patches bereits vorhanden)
    - Ev. auch andere Telefone nach Firmwareanpassungen verwendbar
  - Vorteile
    - Passender Formfaktor für Kontaktlostransaktionen (erregt keine Aufmerksamkeit am Point-of-Sale)
    - Identische Netzwerkschnittstellen wie Device-under-Attack (anderes Mobiltelefon)

Zugriffszeiten auf das Secure Element

- Messung:
  - 1 Befehl-Antwort-Paar
  - Befehl: 13 Byte
  - Antwort: 105 Byte
  - 5000 Wiederholungen
- Variante 1: direkter, externer Zugriff auf Secure Element über Kontaktlosschnittstelle
  - Verzögerung Befehl-Antwort: etwa 30 ms
- Variante 2: direkter, interner Zugriff auf Secure Element über App
  - Verzögerung Befehl-Antwort: 50 und 80 ms
- Variante 3: Relay über WiFi-Verbindung
  - Verzögerung Befehl-Antwort: 190 und 260 ms
- Variante 4: Relay über Mobiltelefonnetz und Internet
  - Verzögerung Befehl-Antwort: > 200 ms (80% unter 4 Sekunden)

Ablauf einer Transaktion mit Google Wallet

- Entspricht EMV Mag-Stripe Kreditkarten-Standard (MasterCard PayPass)
- POS-Terminal: Proximity Payment System Environment (PPSE) Applet auswählen
  - Secure Element: Auswahl bestätigen und Liste der unterstützen EMV Payment Anwendungen zurückgeben
- POS-Terminal: MasterCard Google Prepaid Applet auswählen
  - Secure Element: Auswahl bestätigen und Detailinformationen über die Anwendung zurückgeben (z.B. es handelt sich um eine „MasterCard“)
- POS-Terminal: Parameter des Payment-Systems auslesen („Get Processing Options“)
  - Secure Element: Gibt Parameter zurück (Mag-Stripe Modus, nur Online-Transaktionen, keine Cardholder Verification ...)
- POS-Terminal: Datenfile des Mag-Stipe Modus auslesen
  - Secure Element: Mag-Stripe Track-1- und Track-2-Daten zurückgeben
- POS-Terminal: Berechnung einer kryptographischen Prüfsumme für eine gegebene Zufallszahl anfordern
  - Secure Element: Transaktionszähler und dynamisch generierte CVC3s (Card Verification Code) für die Track-1- und Track-2-Daten zurückgeben

Relay-Attacke gegen Google Wallet

Demo

Relay-Attacke: Gegenmaßnahmen

- Timeout für Transaktion
  - EMV sieht 500 ms pro Transaktion für gute User Experience vor, allerdings muss POS-Terminal langsamere Transaktionen nicht abbrechen
  - Nachteil: Muss auf jedem Terminal implementiert sein!
  - Nachteil: Cloud-basierte Lösungen funktionieren nach dem selben Prinzip wie die Relay Attacke und werden somit ebenfalls verhindert
  - Nachteil: Bei guter Netzwerkverbindung könnten Transaktionen trotzdem funktionieren
- Google Wallet PIN-Code
  - Wird derzeit nur in Mobiltelefon-App(!) überprüft (bei erfolgreicher PIN-Prüfung wird Google Wallet am Secure Element mit einfachen Ein/Aus-Kommandos freigegeben)
  - Stattdessen: PIN-Prüfung direkt am Secure Element (Verwendung des Wallets ist dann nur noch mit gültiger PIN möglich)
- Internal-Mode für Payment-Applets deaktivieren
  - GlobalPlatform bietet die Möglichkeit die Kontaktlosschnittstelle (external mode) und die Kontaktschnittstelle (internal mode) getrennt zu aktivieren
  - Applets können erkennen über welche Schnittstelle sie angesteuert werden
  - Unsicherheitsfaktor: Ist diese Funktionalität auf den Secure Elements im Nexus S bereits verfügbar?

Michael Roland
Research Associate, NFC Research Lab Hagenberg
FH Oberösterreich, Campus 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).