Cnlab/CSI Herbsttagung 2017

Kryptographie in Smartphones

Agenda

Besonderheiten von Smartphones

Teil I: Verschlüsselung auf Smartphones

- PIN und Schlüssel
- Konzepte
- Vergleich

Teil II: Kryptographie in Apps

- TLS
- Certificate Pinning
- Hardwaremodule
- Bibliotheken

Besonderheiten von Smartphones

Smartphones …

- … sind überall dabei
- … sind immer eingeschaltet
- … enthalten Nachrichten und Mails
- … enthalten private Bilder und Medien
- … enthalten Zugangsdaten von Internetdiensten
- … erlauben das Ausführen von Zahlungen

Teil I: Dateiverschlüsselung

Smartphonespezifische Szenarien:

- Unberechtigter Zugriff zum Gerät (Verlust, Diebstahl)
- Unerwünschter Zugriff durch Organisationen (Unternehmen, Behörden, Zoll, Polizei, ...)

Schutzziele:

- Schutz gegen die Verwendung des Geräts (Apps)
- Schutz der gespeicherten Daten

Dateiverschlüsselung: Varianten der Umsetzung

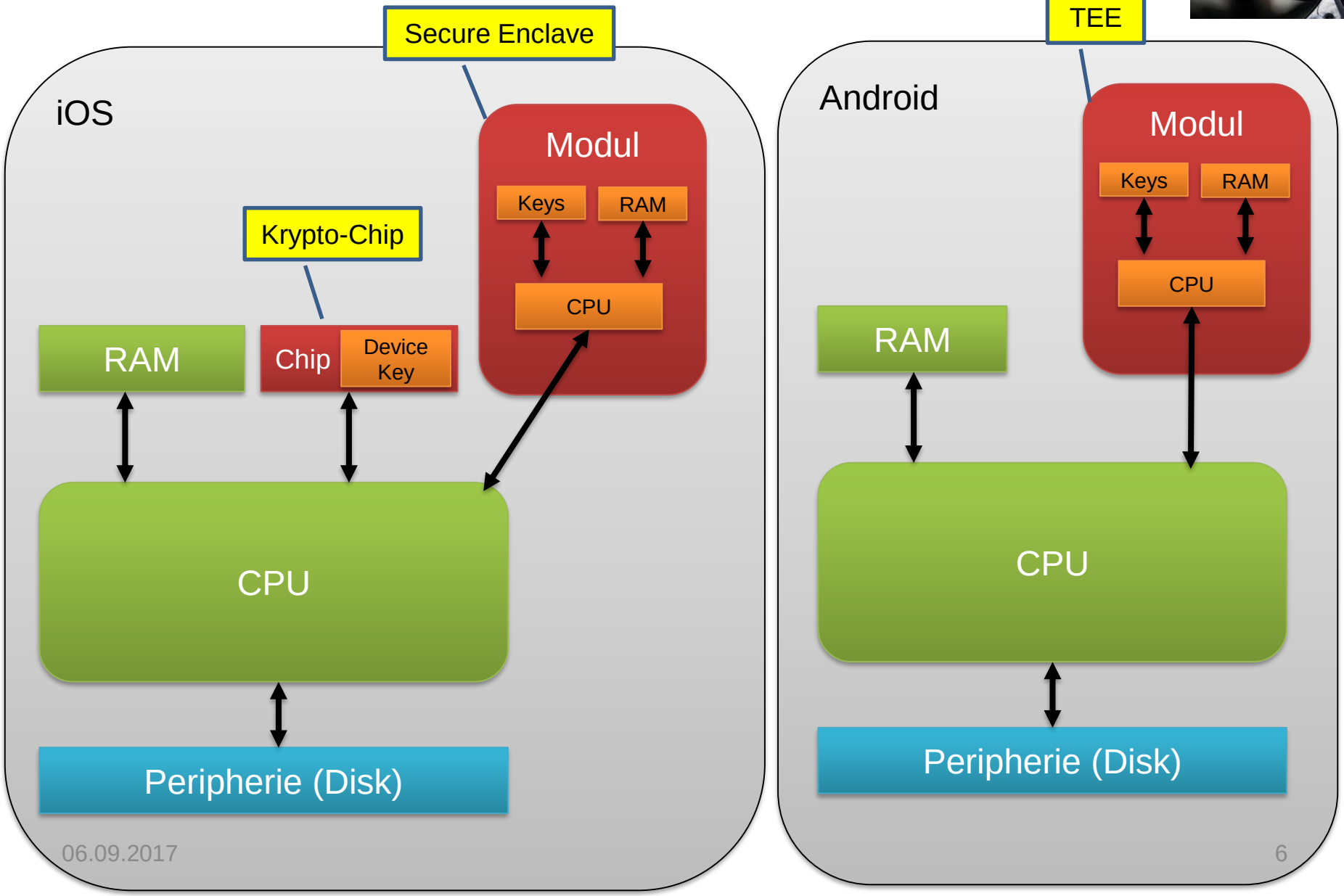
1. «Software only»
   - Gespeicherte Schlüssel können von Software gelesen werden.
   - Unbekannt sind nur PIN/Passwort.
   - «Exhaustive Search» auf PIN/Passwort ist schnell.

2. «Trusted Execution Environment» (TEE)
   - Spezieller Prozessor für sensible Aufgaben (z.B. Schlüsselzugriff).
   - Schlüssel können von der Software nicht gelesen werden.
   - «Secure Boot» detektiert Veränderungen und löscht die Schlüssel

3. «Krypto-Chips»
   - Schlüssel sind in der Hardware «eingebrannt».
   - Können nur auf diesem Gerät verwendet werden

Alle modernen Geräte haben ein TEE, Apple hat zusätzlich eingebrannte Schlüssel.

Umsetzung bei iOS und Android



Ausstattung der Geräte im Markt

Apple

- Krypto-Chip in allen aktuellen Geräten (seit iPhone 5S, 2013).
- Zusätzlich «Secure Enclave».

Android

- Trusted Execution Environment (TEE) gibt es in den meisten modernen Geräten.
- Die Qualität ist herstellerabhängig.

Die zwei Konzepte für die Datei-Verschlüsselung

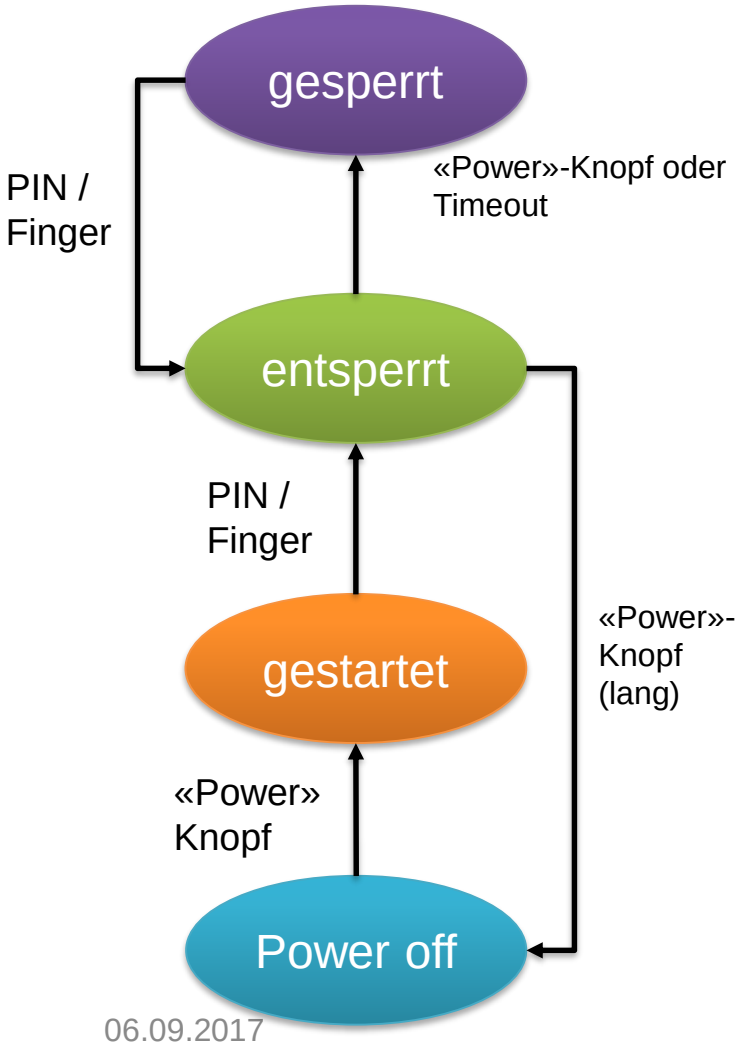
Variante 1:

Verschlüsselung des ganzen Datenträgers («full-disk encryption», FDE)

Variante 2:

Verschlüsselung einzelner Dateien («file-based encryption», FBE)

Datei-Zugriff abhängig vom Geräte-Zustand



iOS	Android <7	Android >=7
Nur Teile «UntilFirstUserAuth»	Alle Daten zugänglich	
Alle Daten zugänglich		
Nur Teile «FileProtectionNone»	Nur OS zugänglich	Nur Teile «Direct Boot»
Keine Daten zugänglich		

Was muss ein Angreifer machen?

1. Gerät läuft
   - Auslesen der Dateien (werden transparent dechiffriert).

2. Gerät läuft nicht
   - Auslesen des Disks oder einzelner chiffrierter Dateien
   - Auslesen des «Keys», falls möglich
   - «Exhaustive Search» über das Passwort

Schutz gegen Malware

Problem:

- Transparente Dateiverschlüsselung hilft nur begrenzt, wenn das Gerät läuft.

Ansätze:

- Keine unnötigen Daten auf Mobilgeräten speichern
- App verschlüsselt heikle Daten zusätzlich

Teil II: Kryptographie in Apps

Klassische Schutzziele:

- Schutz der gespeicherten Daten
- Schutz von Daten beim Transport
- Identifikation des Servers
- Authentisierung des Nutzers

Neu bei Smartphones:

- Sichere Identifikation des physischen Geräts
- Schutz der Daten in Backups

Transportsicherheit

TLS

- Verschlüsselt die Daten beim Transport
- Sollte grundsätzlich genutzt werden
- Browser wissen nicht, welche Server der Nutzer aufsucht, daher «Public Key Infrastructure»

«Certificate Pinning»

- Eine App verbindet nur zu bestimmten Servern.
- Die Identität des Servers wird in die App integriert.

Authentisierung des Nutzers

Nutzerbindung

- Schutz des Geräts durch PIN oder Fingerprint des Nutzers

Technisches Credential

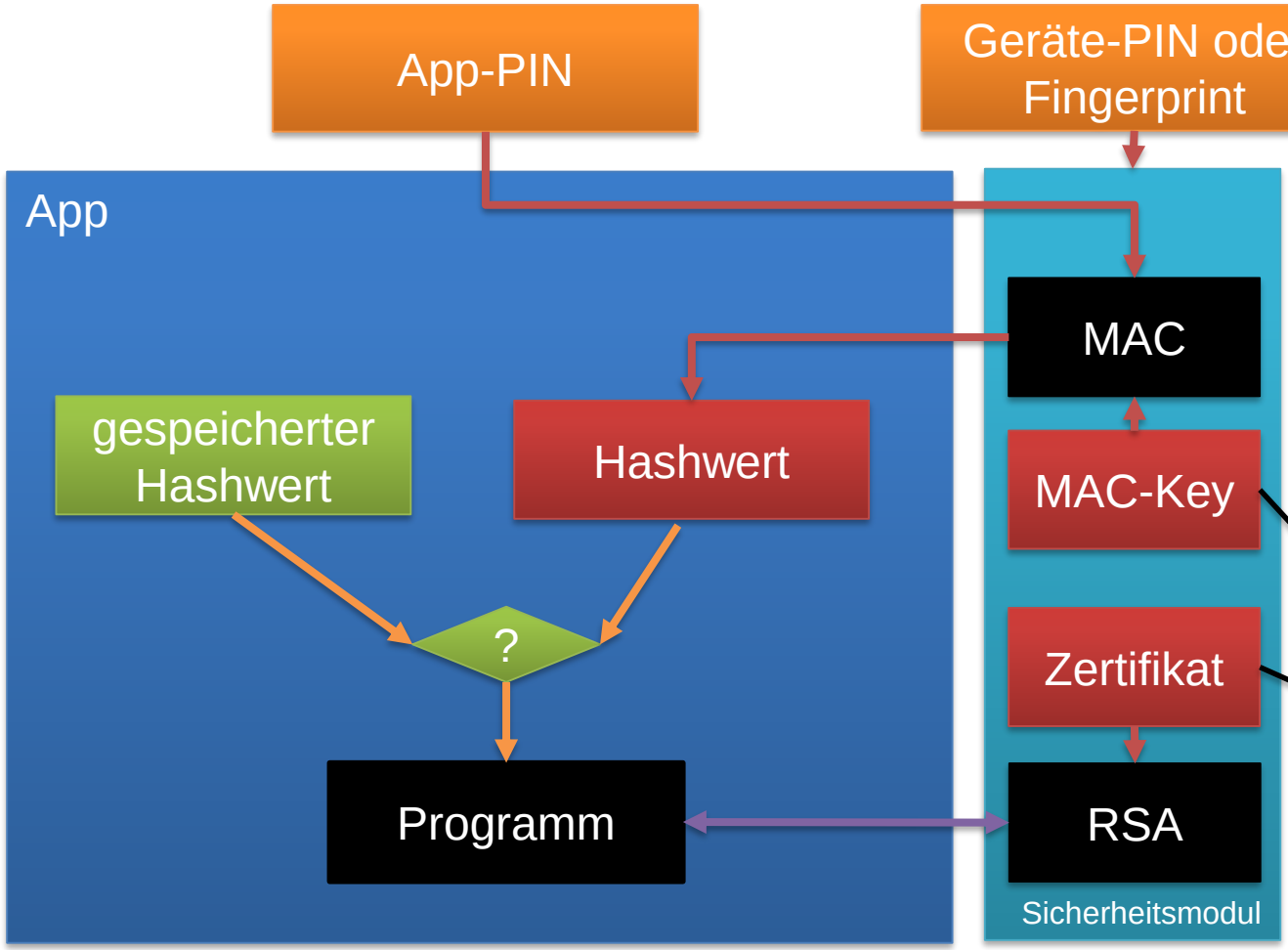
- Identifiziert den Nutzer auf technischer Stufe.
- Als fixes Passwort oder als Client-Zertifikat.
- Wird vom Nutzer freigegeben (PIN / Fingerprint)

Gerätebindung

- Technisches Credential so speichern, dass es nicht auf ein anderes Gerät kopiert werden kann:
  - Keychain oder KeyStore mit «this device only» (wo möglich).

- Hash aus PIN gerechnet
- Vergleich mit Referenz
- Programm verwendet Zertifikat aus der Hardware

Alle Keys in Hardware-Modul

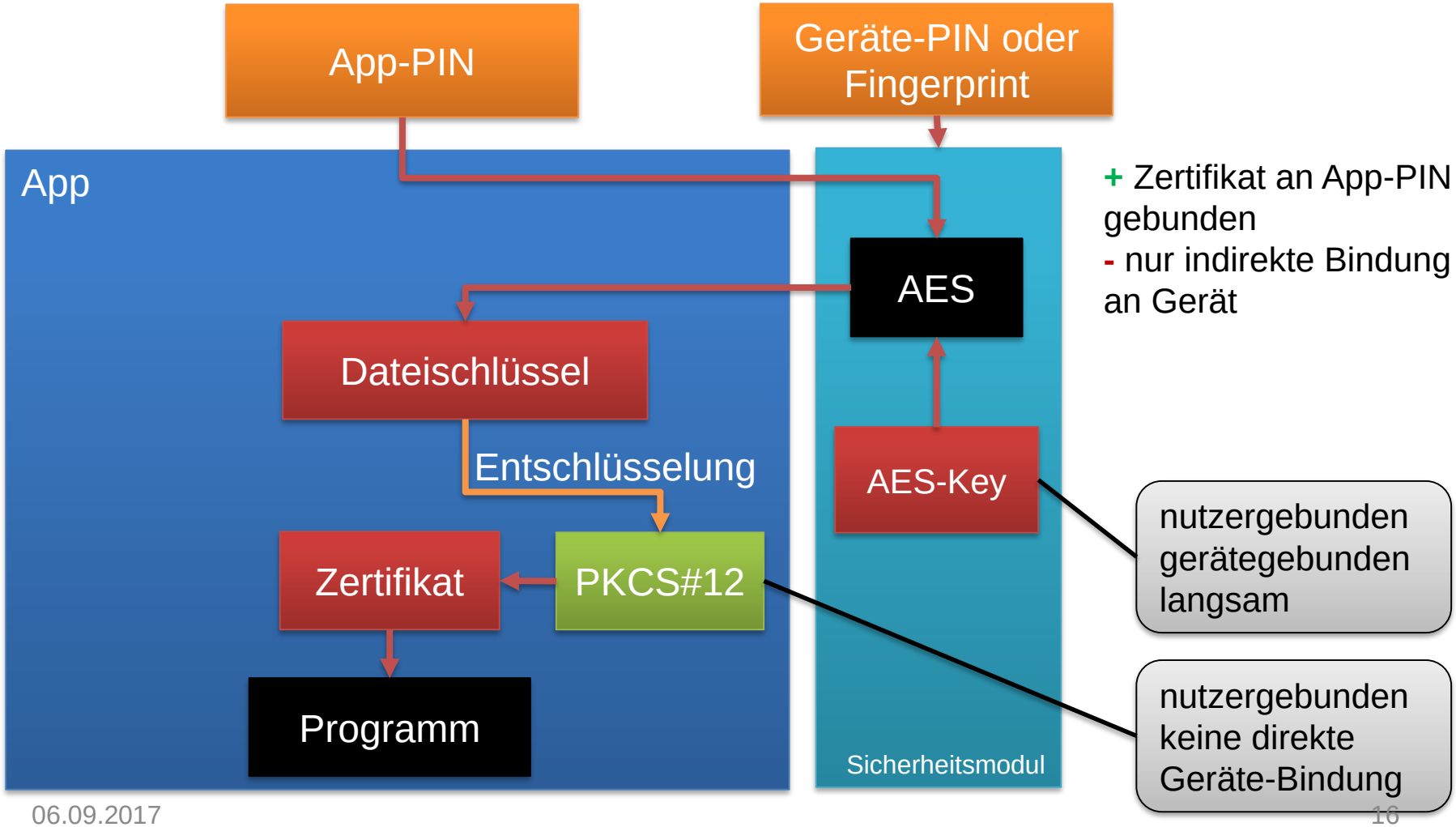


\+ alle Schlüssel an Gerät gebunden
\+ starker Schutz durch Sperrcode
\- nur indirekte Bindung an App-PIN

nutzergebunden
gerätegebunden
langsam

- Zertifikatsschlüssel aus PIN gerechnet (in der HW)

Hardware-Schutz für Key im File-System



Wie werden Krypto-Funktionen korrekt implementiert?

iOS:

- Die meisten Nutzer haben aktuelle Versionen

→ Krypto-API des Systems verwenden

Android:

- Viele Nutzer haben alte Versionen mit bekannten Schwächen

→ Besser eine bekannte Library mit der App ausliefern (z.B. «SpongyCastle»).

Zusammenfassung

- Schlüssel werden bei allen modernen Geräten gut geschützt.
- iOS-Dateien lassen sich im gesperrten Zustand schützen.
- Über Zusatzmassnahmen in den Android-Apps erreicht man einen ähnlich guten Schutz wie bei iOS.
- Standard-Krypto bei iOS und appspezifische Krypto bei Android.

Danke

Stephan Verbücheln
stephan.verbuecheln@cnlab.ch
+41 55 214 33 36

Demo Sessions