Java Card: Secure Channel

IT Security Workshop 2024

Gliederung

- Motivation
- Java Cards
- GlobalPlatform
- Personal Identity Verification (PIV)
- Zusammenfassung & Ausblick

Motivation

- Schlüsselgeneration auf Smartcard
- Autorisierung durch PIN
- **Ziel: PIN kontaktlos übertragen**

Einführung Java Cards

```
package main;
import javacard.framework.APDU;
import javacard.framework.Applet;

public class Smartcard extends Applet {
    public static void install(byte[] bArray, short bOffset, byte bLength) {
        new Smartcard().register();
    }
    private Smartcard() {}

    public void process(APDU apdu) {}
}
```

APDUs (Application Protocol Data Units)

Command

class (CLA)	1 byte	Metadaten (z.B. ob das APDU verschlüsselt ist)
instruction (INS)	1 byte	
param 1 (P1)	1 byte	
param 2 (P2)	1 byte	
command length (Lc)	1-3 bytes	
command data	Lc bytes	
expected length (Le)	1-3 bytes	erwartete Antwortlänge (unverbindlich)

Response

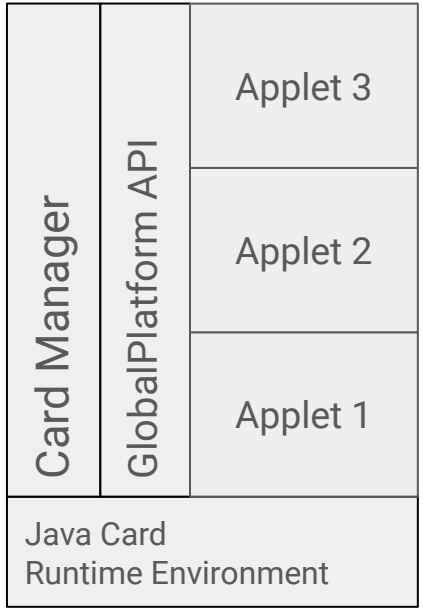
response data	* bytes	
status word (SW)	2 bytes	Fehlercode

Kompilieren + Installieren von Applets

- .cap-Datei kann auf der Java Card installiert werden
  - mit GlobalPlatformPro (gp)
- Automatisierung des Build-Prozesses:
  - Eclipse Plugin
  - Makefile
  - Ant-Javacard
  - Gradle (+ Ant-Javacard)

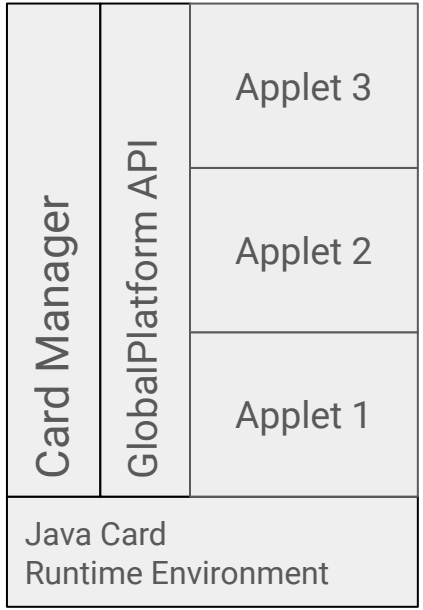
GlobalPlatform

- Organisation, die technische Standards definiert
  - z.B. Installation von Applets auf einer Javacard
- Definiert das Secure Channel Protocol (SCP)
  - Vertrauliche und authentische Kommunikation zwischen Client und Karte
- Unsere Java Cards unterstützen SCP in der Version 03
  - d.h. unser Java Card Applet muss es nicht selber implementieren
- Es existieren drei statische Schlüssel auf der Karte

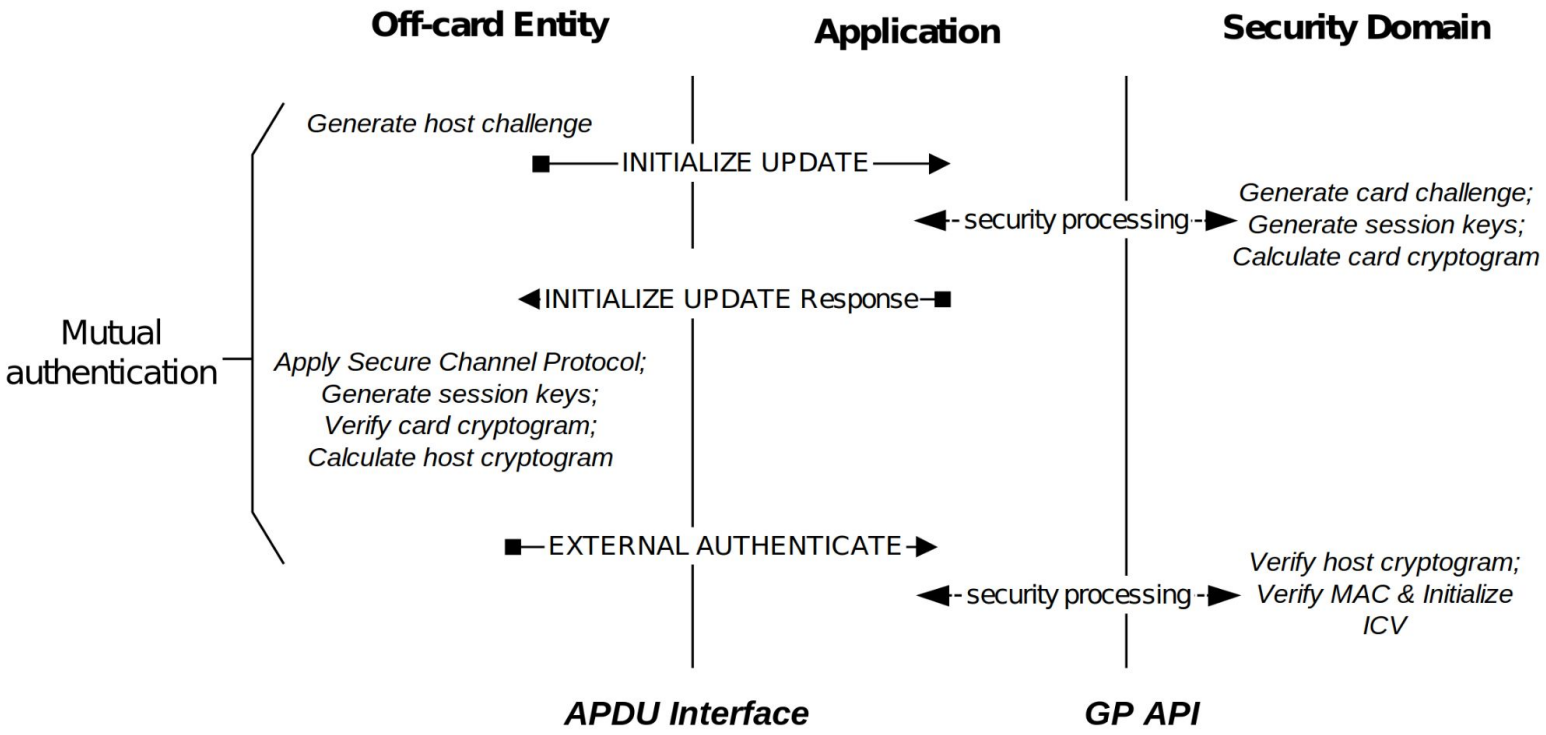


GlobalPlatform

- Organisation, die technische Standards definiert
  - z.B. Installation von Applets auf einer Javacard
- Definiert das Secure Channel Protocol (SCP)
  - Vertrauliche und authentische Kommunikation zwischen Client und Karte
- Unsere Java Cards unterstützen SCP in der Version 03
  - d.h. unser Java Card Applet muss es nicht selber implementieren
- Es existieren drei statische Schlüssel auf der Karte



GlobalPlatform: SCP 03



Quelle: [GlobalPlatform Technology - Secure Channel Protocol '03']

GlobalPlatform: Applet-Implementation

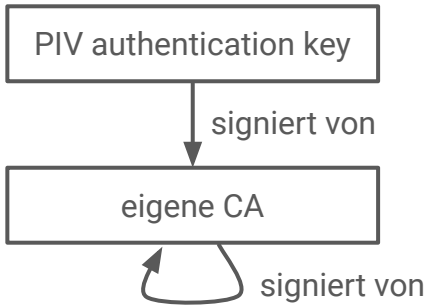
```
import org.globalplatform.SecureChannel;
import org.globalplatform.GPSystem;
/* … */
public void process(APDU apdu) {
    SecureChannel sc = GPSystem.getSecureChannel();
    if (isSecurityRelated(apdu)) {
        short len = sc.processSecurity(apdu);
        apdu.setOutgoingAndSend(ISO7816.OFFSET_CDATA, len);
        return;
    }
    /* … handle non-security ADPUs … */
}
```

GlobalPlatform: Client-Implementation

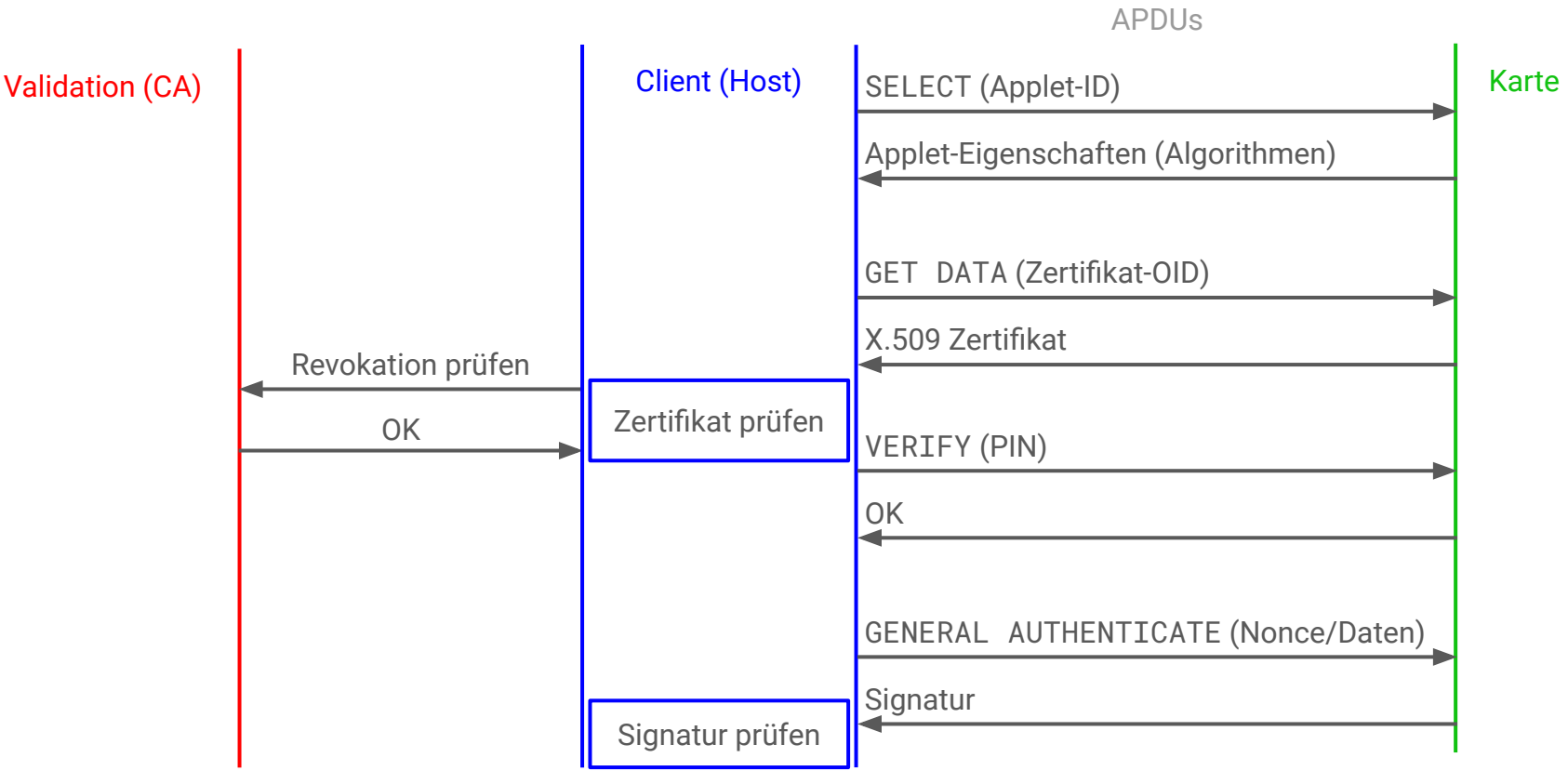
- Eigene Implementation möglich, aber aufwendig
- Es existieren quell-offene Programmbibliotheken:
  - Globalplatform
    - C Bibliothek
    - schwierig zu benutzen
  - Pyglobalplatform
    - Python Wrapper um globalplatform
    - Installation nicht möglich
  - Openjavacard-tools
    - Java Bibliothek
    - Experimental und nicht mehr gepflegt
    - Installation nicht möglich

PIV (Personal Identity Verification)

- Nutzerauthentisierung mit einer Smartcard
- Software-Teil standardisiert in NIST SP 800-73
- Schlüsselgeneration auf der Karte
  - z.B. PIV authentication key, digital signature key, card authentication key
  - verschiedene Security Level
- Authentisierung durch X.509 Zertifikate
  - Karten-Schlüssel signiert von eigener CA
- Implementation
  - Client: OpenSC
  - Applet: PivApplet

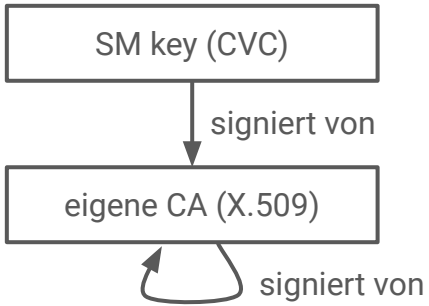


PIV-Ablauf


APDUs
Validation (CA)
Client (Host)
Karte
SELECT (Applet-ID)
Applet-Eigenschaften (Algorithmen)
GET DATA (Zertifikat-OID)
X.509 Zertifikat
Revokation prüfen
OK
Zertifikat prüfen
VERIFY (PIN)
OK
GENERAL AUTHENTICATE (Nonce/Daten)
Signatur
Signatur prüfen

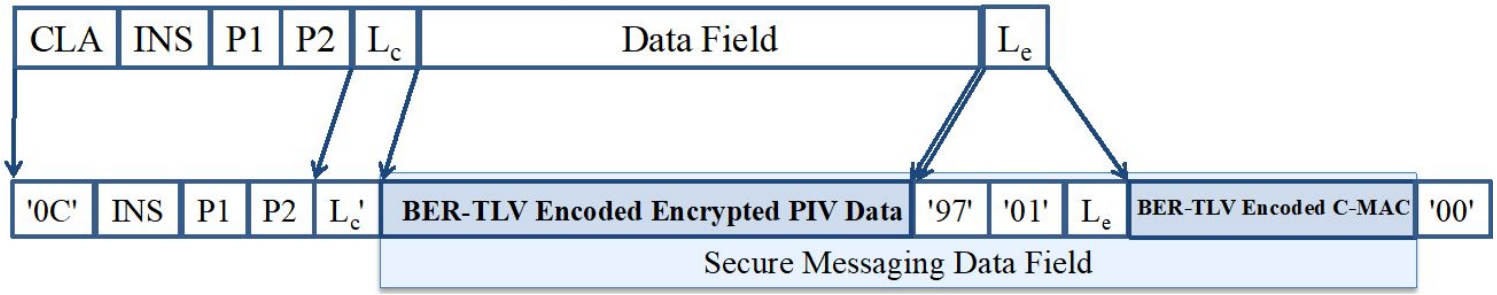
Secure Messaging

- E2E encryption + MACing mit AES (client ↔ card)
- Ähnlichkeiten zwischen OpenPGP card und PIV
  - OpenPGP-Applet wiederverwendbar
- Aber:
  - OpenPGP card:
    - statischer AES Schlüssel
  - PIV (NIST SP 800-73):
    - key agreement für AES-Schlüssel (ECC CDH 1e 1s)
    - statischer Schlüssel auf der Karte generiert
    - Authentisierung des Kartenschlüsselanteils über CVC

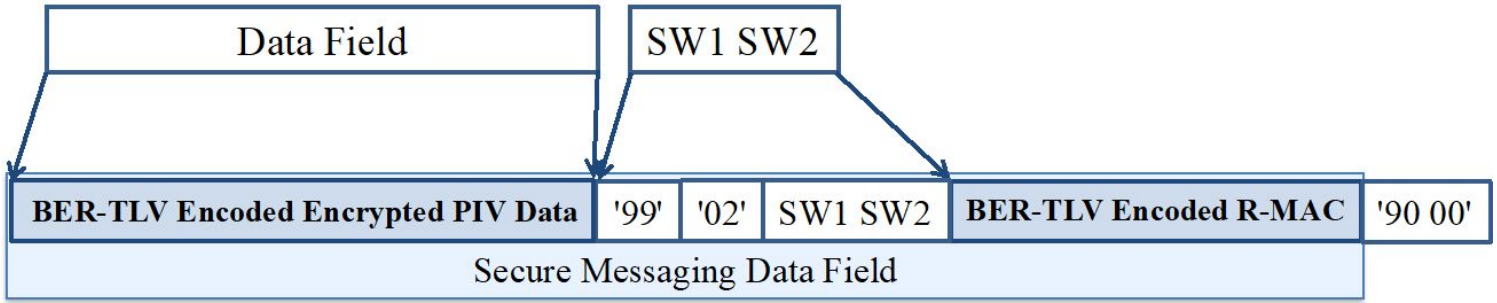


Secure Messaging: Datenaustausch

command:



response:



Quelle: NIST SP 800-73-5 part 2 § 4.2

Secure Messaging: Schlüsselübereinkunft

Client

ephemeres Schlüsselpaar generieren
öffentlichen Schlüssel senden

CVC prüfen
gemeinsames Geheimnis berechnen
Sessionschlüsselableitung von Geheimnis + Nonce
Kryptogramm von Sessionschlüssel erstellen
Kryptogramm vergleichen

Karte

gemeinsames Geheimnis berechnen
Sessionschlüsselableitung von Geheimnis + Nonce
Kryptogramm von Sessionschlüssel erstellen
CVC + Krytogramm + Nonce senden

Probleme beim Key Agreement

- Erstellung des CVC ist unklar
  - Aufbau ist in NIST SP 800-73 beschrieben
  - Es gibt Werkzeuge zur Erstellung von CVC
  - Beispiel-CVCs entsprechen nicht ganz dem NIST-Standard
- keine Standard-Instruktion zum Laden des CVC auf die Karte
  - Zu jedem Key-Slot gibt es ein X.509 Zertifikat auf der Karte
  - Jedoch nicht für das (CVC) Zertifikat des PIV Secure Messaging Key
  - Müsste selber implementiert werden
- Schlüsselableitungsfunktion in Java Card nicht vorhanden
  - Keine externe Implementation gefunden
  - Keine Zeit für eigene Implementation

Zusammenfassung

- Einstieg in das Thema nicht einfach
  - viele verschiedene Standards
  - z.B. zur ADPU-Verschlüsselung gibt es SCP, OpenPGP SM, PIV SM, …
- schlechte Dokumentation
- schlechte Toolchain
  - vgl. z.B. Arduino
- (quelloffene) Java Card Bibliotheken sind selten bis nicht existent

Nächste Schritte

- Konkrete Verfahren nach NIST Standard implementieren
  - in eigenständige Bibliotheken statt in ein Applet
  - z.B. KDF, NIST P-Kurven, …
- CVC-Verwaltung
  - Tool zum Erstellen
  - CVC laden (Client- und Appletseite)
- OpenSC
  - Prüfung des SM Certificate Signer
  - `PIV_USE_SM=always` wird ignoriert, wenn die Karte kein SM kann

Vielen Dank für Eure Aufmerksamkeit!