Ein Mittelsmannangriff auf ein digitales Signiergerät

Alexander Koch

2011

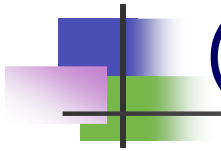
Inhalt

- Gesetzliche Rahmenbedingungen
- Umsetzung
- Produktauswahl
- Ablauf eines Signaturvorgangs
- Analyse
  - Entstehende Signatur
  - Transfer
- Angriff
  - Vorgehen zur Fälschung
  - Zugriff auf die transferierten Daten

Gesetzliche Rahmenbedingungen (Europäische Union)

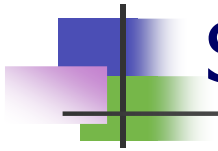
- EU-Richtlinie 1999/93/EG
- Vereinfachung des Binnenhandels
- Gleichgestellte Rechtswirkung elektronischer Signaturen für ein unterzeichnetes elektronisches Dokument mit „handschriftlichen Unterschriften in Bezug auf Daten, die auf Papier vorliegen“
- Umzusetzen bis 19. Juli 2001

Gesetzliche Rahmenbedingungen (Deutschland)



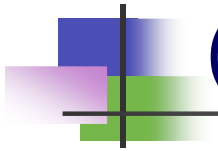
- SignaturGesetz (SigG)
- SignaturVerordnung (SigV)
  - Erweiterung um Anforderungen an Aussteller (Zertifizierungsdiensteanbieter) und Verfahren

Gesetzliche Rahmenbedingungen SignaturGesetz (SigG)



- 3 Güteklassen
  - Einfache elektronische Signatur
  - Fortgeschrittene elektronische Signatur
    - Ausschließlich dem Unterzeichner zugeordnet
    - Ermöglicht identifizierung des Unterzeichners
    - Kann mit Mitteln erstellt werden, die der Unterzeichner unter alleiniger Kontrolle halten kann
    - Macht Veränderung der unterzeichneten Daten erkennbar
  - Qualifizierte elektronische Signatur
    - Fortgeschrittene elektronische Signatur
    - Qualifiziertes Zertifikat
    - Sichere Signaturerstellungseinheit (SSEE)

Gesetzliche Rahmenbedingungen (qualifiziertes Zertifikat)



- Zertifikat
  - Datensatz, der verwendet wird, um Eigenschaften zu bestätigen
  - Kann durch kryptografische Verfahren geprüft werden
- Qualifiziert, wenn folgende Daten enthalten:
  - Aussteller + Staat
  - Name des Inhabers / Pseudonym
  - Öffentlicher Schlüssel des Inhabers
  - Gültigkeitszeitraum
  - Seriennummer
  - (mindestens) Fortgeschrittene Signatur des Ausstellers
  - Angabe, dass es sich um qualifiziertes Zertifikat handelt

Gesetzliche Rahmenbedingungen (SSEE)

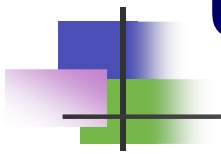
Müssen gewährleisten, dass

- Signaturschlüssel
  - nur einmal auftreten
  - geheim gehalten werden
  - nicht abgeleitet werden können
  - vor Fälschungen geschützt sind
  - vor der Verwendung durch andere geschützt werden
- Daten beim Signieren
  - nicht verändert werden
  - dargestellt werden können

Fazit (Gesetzliche Rahmenbedingungen)

Der Gesetzgeber versucht elektronische Signaturen mit herkömmlichen Signaturen gleichzustellen in Bezug auf

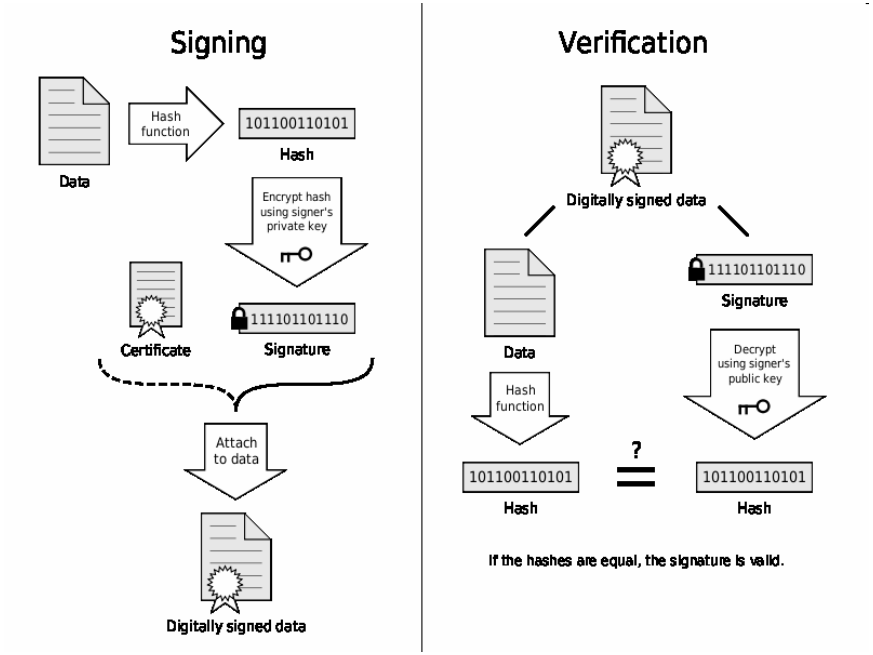
- Rechtswirkung
- (Fälschungs-)Sicherheit



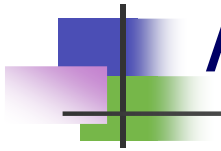
Umsetzung

- (meist) mit Prozessorchipkarten
  - beinhalten
    - Zertifikate
    - Privaten Schlüssel
  - benötigen
    - PIN (oder anderes Identifizierungsmerkmal)
    - Kartenlesegerät
    - Software zum Ansteuern

Umsetzung (Signaturvorgang)



de.wikipedia.org/w/index.php?title=Datei:Digital_Signature_diagram.svg



Aufgabenstellung

Es soll überprüft werden,

inwieweit die USB-Verbindung zwischen Arbeitsplatzrechner und Kartenleser abgesichert ist. Insbesondere soll ermittelt werden, ob und ggf. wie ein Mittelsmannangriff erfolgreich sein kann.

Produktauswahl

- Deutsche Post Com GmbH
  - Signtrust
    - Chipkartenterminal
      - Cherry Smart Terminal ST2000-U
        - USB / Circuit Card Interface Device
        - Class 2
    - Chipkarte
      - Infineon SLE66CX680PE
      - Giesecke & Devrient StarCos 3.2
      - ISO® **7816**-3/**-4**/**-8**/-9
    - Signatursoftware
      - **intarsys SignLive! CC**
      - OpenLimit CC Sign

Ablauf

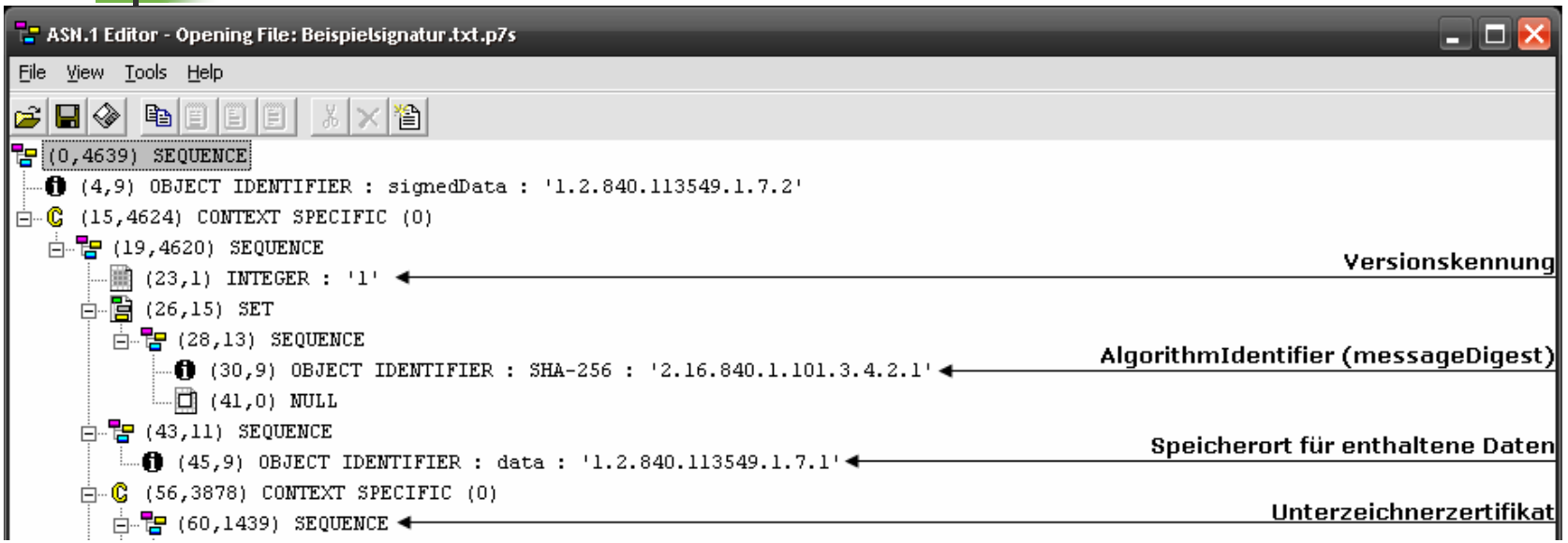
- Auswahl einer Datei
- Auswahl des Signaturverfahrens
- Auswahl von Attributzertifikaten
- Einleitung des eigentlichen Signaturvorgangs
- Eingabe der PIN auf dem Chipkartenterminal
- Erfolgs- / Fehlermeldung

Analyse (entstehende Signatur)

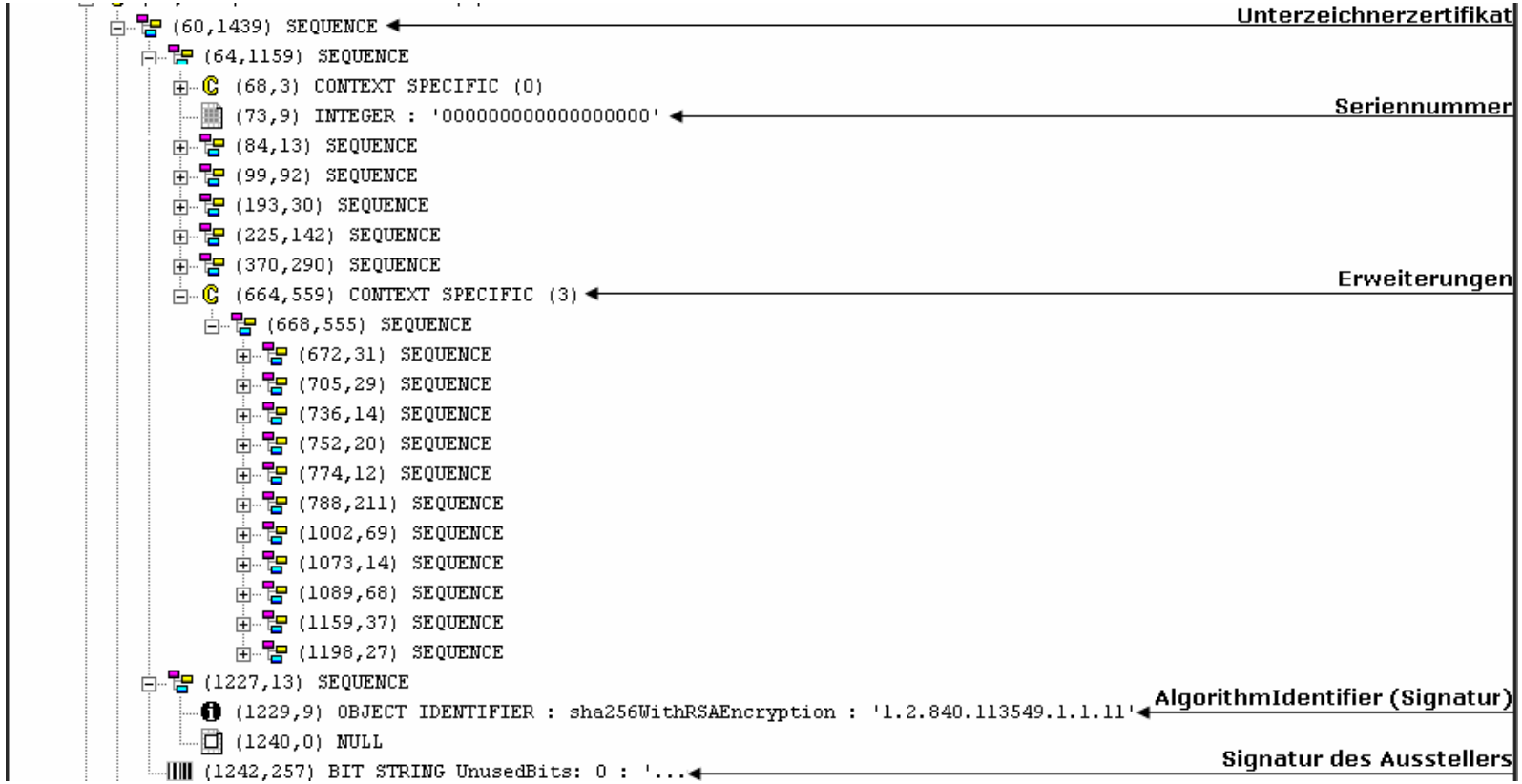
- Dateitypen
  - .p7s
  - .p7m
    - PKI-Signatur im PKCS#7-Format
      - RFC5652
        - ASN.1

PKI: Public Key Infrastructure
PKCS: Public Key Cryptograpy Standards
ASN.1: Abstract Syntax Notation One

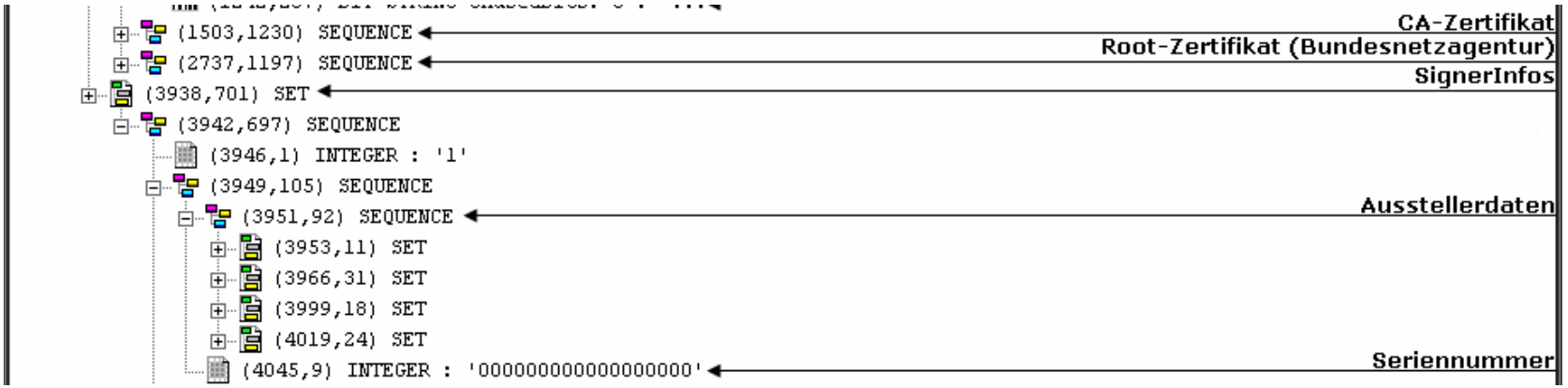
Analyse (entstehende Signatur)



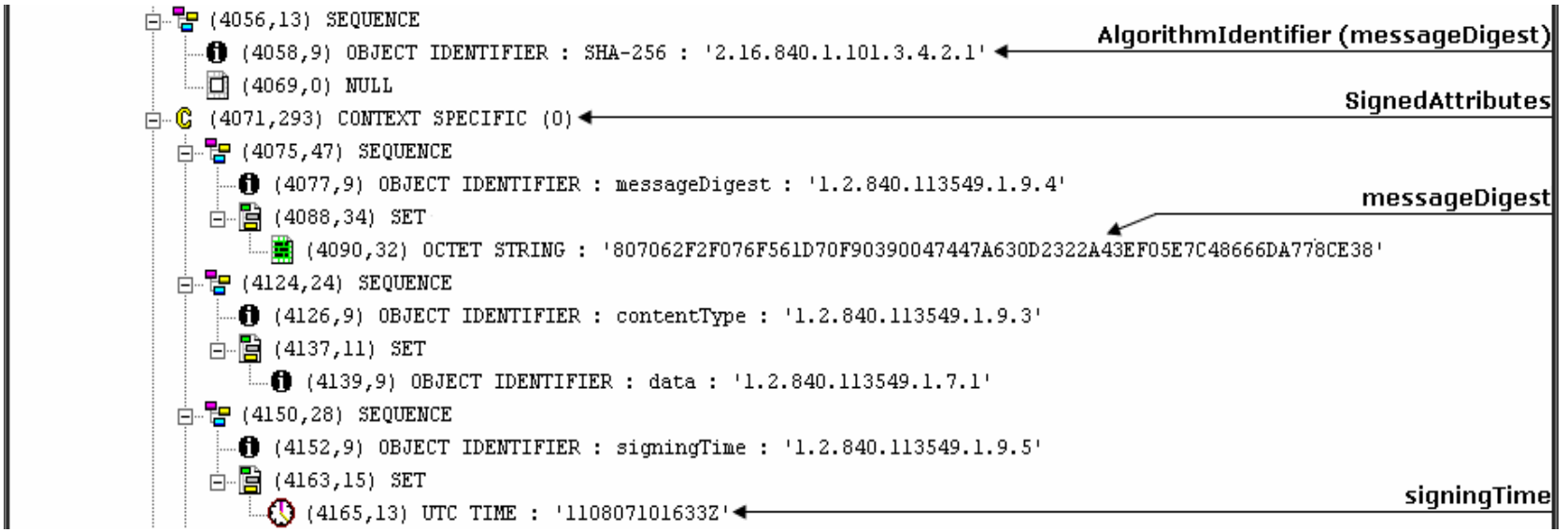
Analyse


(60,1439) SEQUENCE
Unterzeichnerzertifikat
(64,1159) SEQUENCE
(68,3) CONTEXT SPECIFIC (0)
(73,9) INTEGER : '000000000000000000'
Seriennummer
(84,13) SEQUENCE
(99,92) SEQUENCE
(193,30) SEQUENCE
(225,142) SEQUENCE
(370,290) SEQUENCE
(664,559) CONTEXT SPECIFIC (3)
Erweiterungen
(668,555) SEQUENCE
(672,31) SEQUENCE
(705,29) SEQUENCE
(736,14) SEQUENCE
(752,20) SEQUENCE
(774,12) SEQUENCE
(788,211) SEQUENCE
(1002,69) SEQUENCE
(1073,14) SEQUENCE
(1089,68) SEQUENCE
(1159,37) SEQUENCE
(1198,27) SEQUENCE
(1227,13) SEQUENCE
(1229,9) OBJECT IDENTIFIER : sha256WithRSAEncryption : '1.2.840.113549.1.1.11'
AlgorithmIdentifier (Signatur)
(1240,0) NULL
(1242,257) BIT STRING UnusedBits: 0 : '...
Signatur des Ausstellers

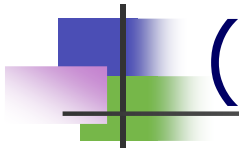
Analyse
(entstehende Signatur)



Analyse (entstehende Signatur)



Analyse (entstehende Signatur)



Analyse (entstehende Signatur)

- Signaturdatei enthält
  - Zertifikate
  - AlgorithmIdentifier
  - *signedAttributes*
    - Nimmt Bezug auf Zertifikate
    - Enthält
      - *messageDigest*
      - *signingTime*
  - *signatureValue*
  - Längenangaben für einzelne Abschnitte

Wo kommen die Daten her?

Analyse (Transfer)

- USBSnoop / SniffUSB
- Byteweise Anzeige der übertragenen Rohdaten

```
TransferBufferMDL    = 89256860
  00000000: 09 02 5d 00 01 01 03 a0 32 09 04 00 00 03 ff 00
  00000010: 00 04 36 21 00 01 00 01 03 00 00 00 a0 0f 00 00
  00000020: 40 1f 00 00 00 01 2a 00 00 29 40 05 00 00 fe 00
  00000030: 00 00 00 00 00 00 00 00 00 00 ba 00 01 00 0e 01
  00000040: 00 00 ff ff 00 00 03 01 07 05 01 02 40 00 00 07
  00000050: 05 82 02 40 00 00 07 05 83 03 10 00 10
UrbLink              = 00000000
SetupPacket          =
  00000000: 80 06 00 02 00 00 09 02
```

```
Interface[0]: NumberOfPipes     = 3
Interface[0]: Pipes[0] : MaximumPacketSize = 0x00000040
Interface[0]: Pipes[0] : EndpointAddress   = 0x00000001
Interface[0]: Pipes[0] : Interval          = 0x00000000
Interface[0]: Pipes[0] : PipeType          = 0x00000002 (UsbdPipeTypeBulk)
Interface[0]: Pipes[0] : PipeHandle        = 0x863b3ee4
Interface[0]: Pipes[0] : MaxTransferSize   = 0x00400000
Interface[0]: Pipes[0] : PipeFlags         = 0x00000000
Interface[0]: Pipes[1] : MaximumPacketSize = 0x00000040
Interface[0]: Pipes[1] : EndpointAddress   = 0x00000082
Interface[0]: Pipes[1] : Interval          = 0x00000000
Interface[0]: Pipes[1] : PipeType          = 0x00000002 (UsbdPipeTypeBulk)
Interface[0]: Pipes[1] : PipeHandle        = 0x863b3f04
Interface[0]: Pipes[1] : MaxTransferSize   = 0x00400000
Interface[0]: Pipes[1] : PipeFlags         = 0x00000000
Interface[0]: Pipes[2] : MaximumPacketSize = 0x00000010
Interface[0]: Pipes[2] : EndpointAddress   = 0x00000083
Interface[0]: Pipes[2] : Interval          = 0x00000010
Interface[0]: Pipes[2] : PipeType          = 0x00000003 (UsbdPipeTypeInterrupt)
Interface[0]: Pipes[2] : PipeHandle        = 0x863b3f24
Interface[0]: Pipes[2] : MaxTransferSize   = 0x00400000
Interface[0]: Pipes[2] : PipeFlags         = 0x00000000
```

```
  TransferFlags          = 00000003 (USBD_TRANSFER_DIRECTION_IN, USBD_SHORT_TRANSFER_OK)
  TransferBufferLength = 00000030
    00000000: 83 26 00 00 00 00 1e 01 00 00 06 01 03 03 00 01
    00000010: 00 01 01 1c 32 00 31 00 31 00 32 00 31 00 30 00
    00000020: 34 00 32 00 31 00 32 00 31 00 34 00 34 00 30 00
  UrbLink                = 00000000
[181 ms]  >>>  URB 8 going down  >>>
-- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
  PipeHandle             = 863b3ee4 [endpoint 0x00000001]
  TransferFlags          = 00000002 (USBD_TRANSFER_DIRECTION_OUT, USBD_SHORT_TRANSFER_OK)
  TransferBufferLength = 0000000b
    00000000: 6b 01 00 00 00 00 2b 00 00 00 2b
  UrbLink                = 00000000
[184 ms]  <<<  URB 9 coming back  <<<
-- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
  PipeHandle             = 863b3f04 [endpoint 0x00000082]
  TransferFlags          = 00000003 (USBD_TRANSFER_DIRECTION_IN, USBD_SHORT_TRANSFER_OK)
  TransferBufferLength = 0000000e
    00000000: 83 04 00 00 00 00 2b 01 00 00 01 00 00 00
  UrbLink                = 00000000
[184 ms]  >>>  URB 10 going down  >>>
-- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
  PipeHandle             = 863b3ee4 [endpoint 0x00000001]
  TransferFlags          = 00000002 (USBD_TRANSFER_DIRECTION_OUT, USBD_SHORT_TRANSFER_OK)
  TransferBufferLength = 0000000c
    00000000: 6b 02 00 00 00 00 87 00 00 00 87 00
  UrbLink                = 00000000
[187 ms]  <<<  URB 11 coming back  <<<
```

```
-- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
  PipeHandle            = 863b3f04 [endpoint 0x00000082]
  TransferFlags         = 00000003 (USBD_TRANSFER_DIRECTION_IN, USBD_SHORT_TRANSFER_OK)
  TransferBufferLength = 0000010c
    00000000: 80 02 01 00 00 00 6f 00 00 01 00 60 fe 32 38 30
    00000010: 30 30 32 31 2e 30 2c 06 09 2a 86 48 86 f7 0d 01
    00000020: 09 01 16 1f 61 6c 65 78 61 6e 64 65 72 2e 6b 6f
    00000030: 63 68 38 37 40 67 6f 6f 67 6c 65 6d 61 69 6c 2e
    00000040: 63 6f 6d 30 82 01 22 30 0d 06 09 2a 86 48 86 f7
    00000050: 0d 01 01 01 05 00 03 82 01 0f 00 30 82 01 0a 02
    00000060: 82 01 01 00 a9 0d 62 0f a2 00 dc 93 03 42 cb da
    00000070: 17 48 01 87 a3 36 84 5b ce 6f 15 19 8e b5 9c 2b
    00000080: ff 84 a1 50 d7 7b 7a 5c c9 5e 88 08 98 a4 39 7d
    00000090: a0 cb 3c 8d e3 ed 0a 83 6c 61 74 2c a4 60 2b ef
    000000a0: e3 ed 8d 53 08 fb a5 23 db 04 e0 6d 84 dd 64 ca
    000000b0: f8 e5 24 4a 1c 43 99 43 05 eb 26 44 49 51 f0 d3
    000000c0: e3 92 d2 fd 81 e1 c4 2f 21 ae 99 ce a1 bb 3b e3
    000000d0: 09 0e 62 7c 82 bd 2d da a8 2c c2 a0 95 c0 98 6d
    000000e0: 5a d2 d8 8e 5b 93 ce df 7c a5 9e 21 f4 9d 06 55
    000000f0: 3d c8 3f 71 65 0f d3 31 0c 76 a7 8d 85 e1 13 78
    00000100: ed d3 ea 2d 9b 52 62 ab c9 2b ce 9c
```

```
[28863 ms]  <<<  URB 198 coming back  <<<
-- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
  PipeHandle           = 863b3f24 [endpoint 0x00000083]
  TransferFlags        = 00000003 (USBD_TRANSFER_DIRECTION_IN, USBD_SHORT_TRANSFER_OK)
  TransferBufferLength = 00000002
    00000000: a0 91
  UrbLink              = 00000000
[29167 ms]  <<<  URB 199 coming back  <<<
-- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
  PipeHandle           = 863b3f24 [endpoint 0x00000083]
  TransferFlags        = 00000003 (USBD_TRANSFER_DIRECTION_IN, USBD_SHORT_TRANSFER_OK)
  TransferBufferLength = 00000002
    00000000: a0 91
  UrbLink              = 00000000
[29648 ms]  <<<  URB 200 coming back  <<<
-- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
  PipeHandle           = 863b3f24 [endpoint 0x00000083]
  TransferFlags        = 00000003 (USBD_TRANSFER_DIRECTION_IN, USBD_SHORT_TRANSFER_OK)
  TransferBufferLength = 00000002
    00000000: a0 90
```

Analyse (Transfer)

- Zu Beginn: USB-Konfigurationsdaten
  - 3 Endpunkte
    - Bulk OUT
    - Bulk IN
    - Interrupt IN
- Datenblöcke
- PIN-Eingabe
- Unterschiede in wenigen URBs

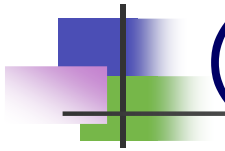
URB: USB Request Block

Analyse (Transfer)

- Verwendete Kommunikationsarten
  - USB
  - CCID Messages
    - 10 Byte Header für den Kartenleser
  - ISO 7816-4 / -8 APDUs
    - Verschiedene Kommandos für die Karte

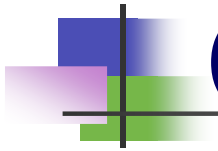
Analyse (Transfer)

- Datenblöcke auf bestimmte Kommandos folgend:
  - SELECT (00 A4 02 04 + FileID)
  - READ BINARY (00 B0 + 2 Byte Offset)
  - VERIFY (00 20 00 81)
    - Antwort bei Pin korrekt: 90 00
    - Antwort bei Pin falsch: 63 CX X = Versuche übrig
  - COMPUTE DIGITAL SIGNATURE
    (00 2a 9e 9a + Hash) (Hash != Datei-Hash)
    - Antwort: signatureValue



Fazit (Transfer)

- Ungesicherte Datenübertragung
  - Erkennen der Nutzdaten möglich
- Elemente der Signaturdatei werden in benötigter Reihenfolge übertragen
  - Abgreifen der Nutzdaten ohne besonderen Aufwand möglich



Angriff (Fälschung der Signaturdatei)

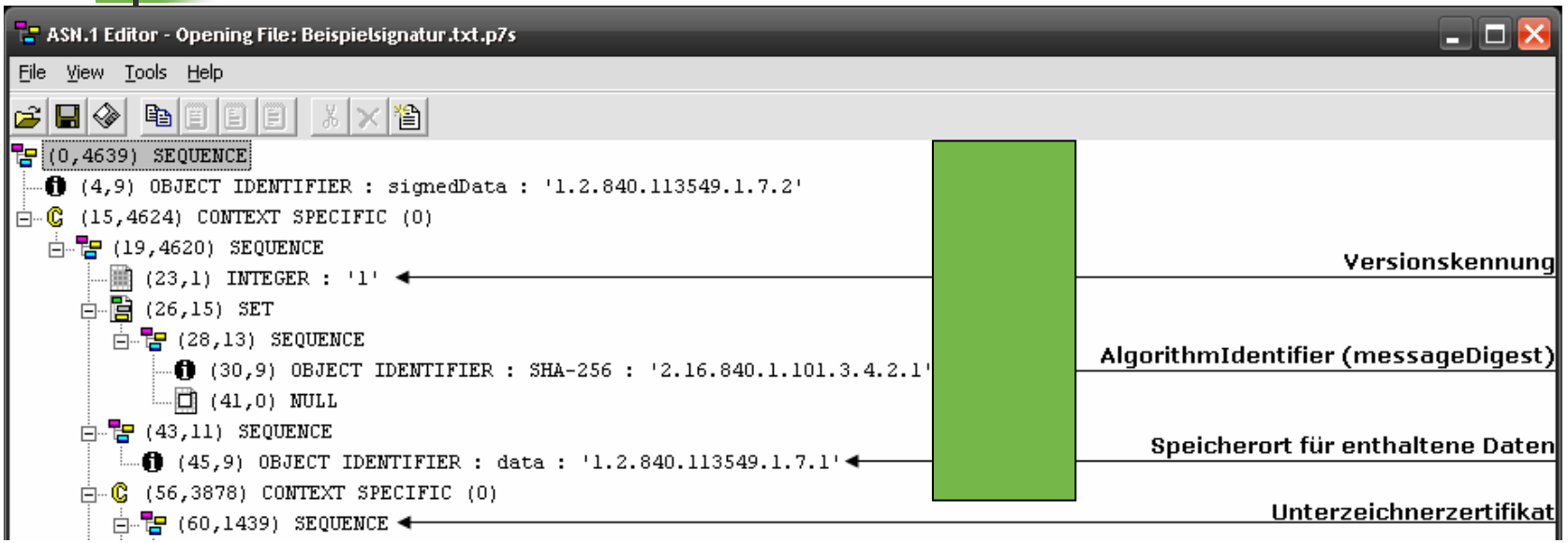
Angriff auf Signaturgerät

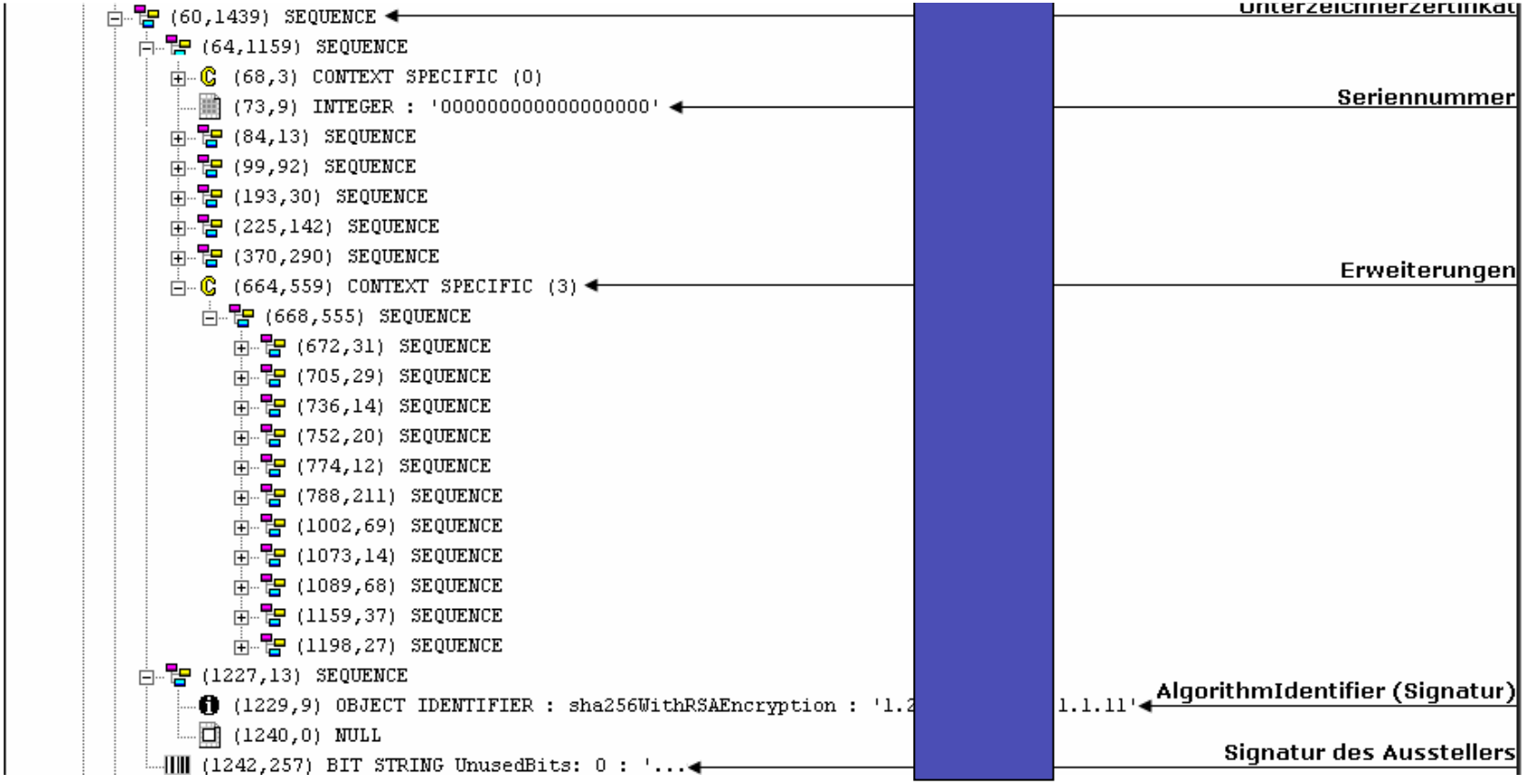
= Fälschen der Signatur

= Fälschung der Signaturdatei

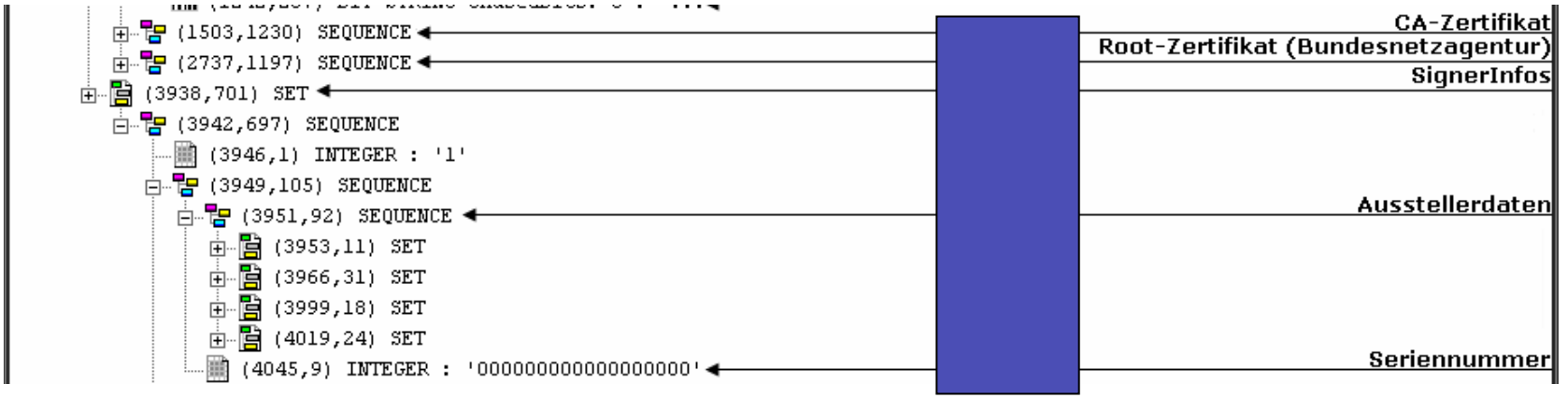
- wenige Variablen
- (nahezu) linearer Aufbau möglich

Analyse (entstehende Signatur)

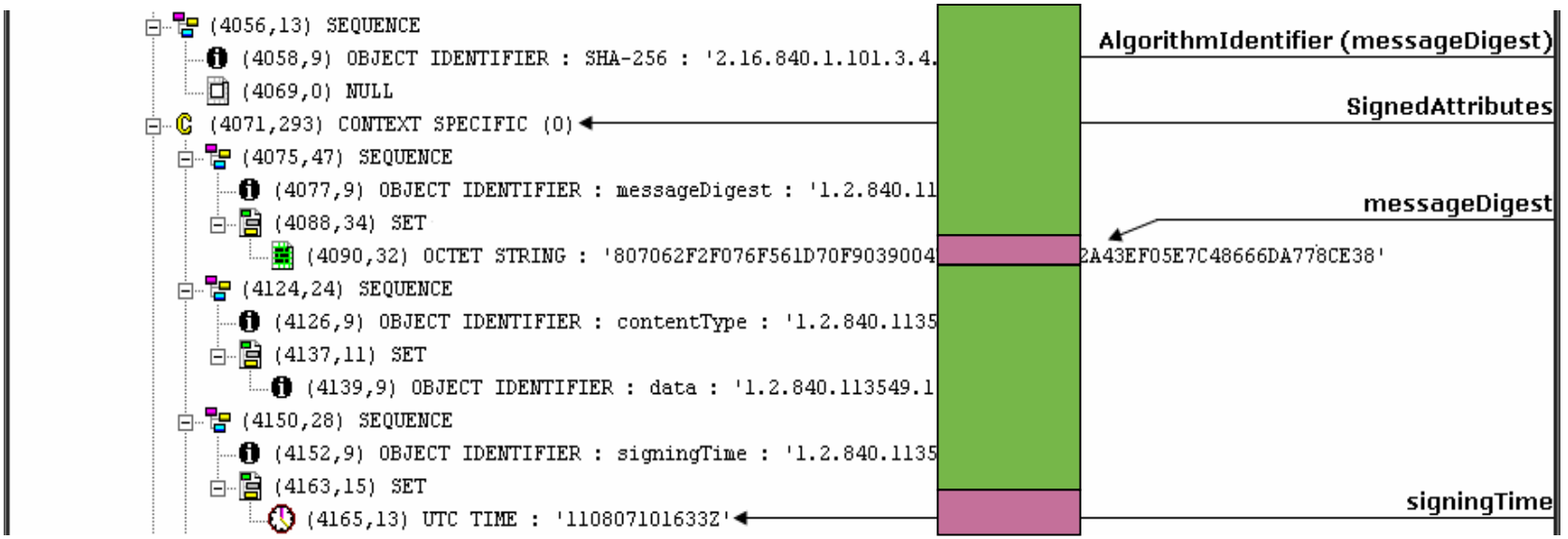



(60,1439) SEQUENCE
(64,1159) SEQUENCE
(68,3) CONTEXT SPECIFIC (0)
(73,9) INTEGER : '0000000000000000000'
(84,13) SEQUENCE
(99,92) SEQUENCE
(193,30) SEQUENCE
(225,142) SEQUENCE
(370,290) SEQUENCE
(664,559) CONTEXT SPECIFIC (3)
(668,555) SEQUENCE
(672,31) SEQUENCE
(705,29) SEQUENCE
(736,14) SEQUENCE
(752,20) SEQUENCE
(774,12) SEQUENCE
(788,211) SEQUENCE
(1002,69) SEQUENCE
(1073,14) SEQUENCE
(1089,68) SEQUENCE
(1159,37) SEQUENCE
(1198,27) SEQUENCE
(1227,13) SEQUENCE
(1229,9) OBJECT IDENTIFIER : sha256WithRSAEncryption : '1.2
1.1.11'
(1240,0) NULL
(1242,257) BIT STRING UnusedBits: 0 : '...
Unterzeichnerzertifikat
Seriennummer
Erweiterungen
AlgorithmIdentifier (Signatur)
Signatur des Ausstellers

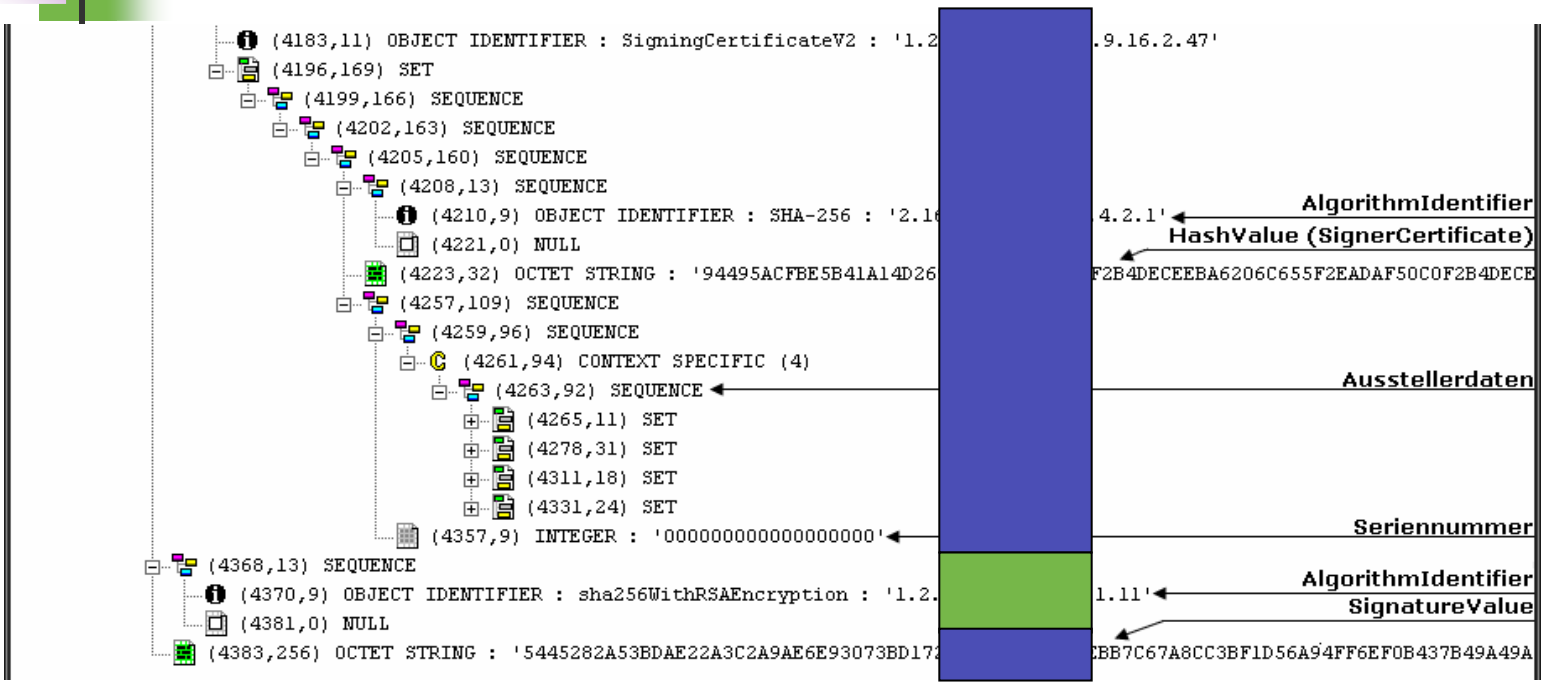
Analyse (entstehende Signatur)



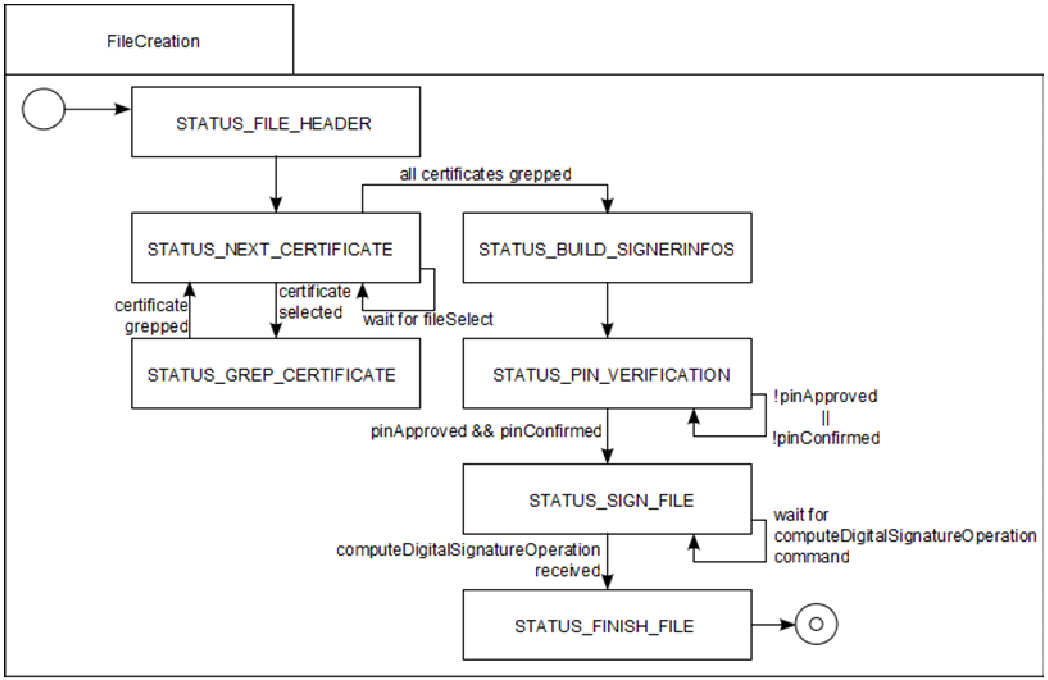
Analyse (entstehende Signatur)



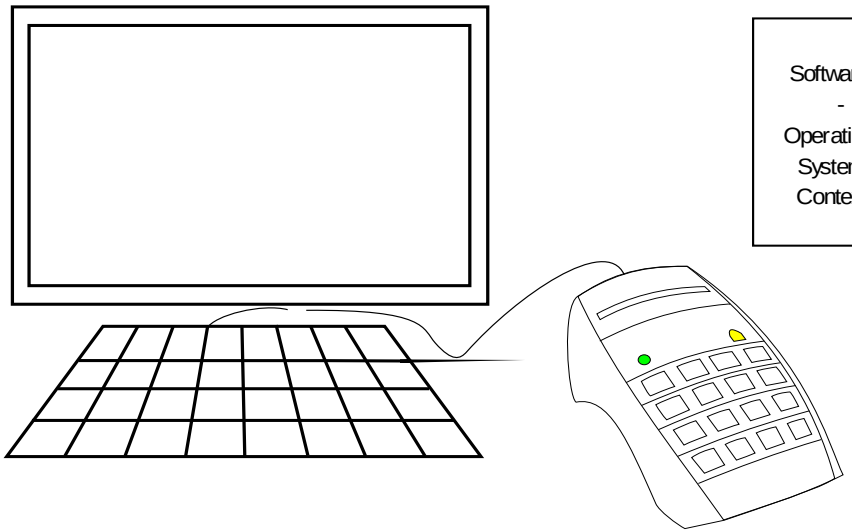
Analyse (entstehende Signatur)

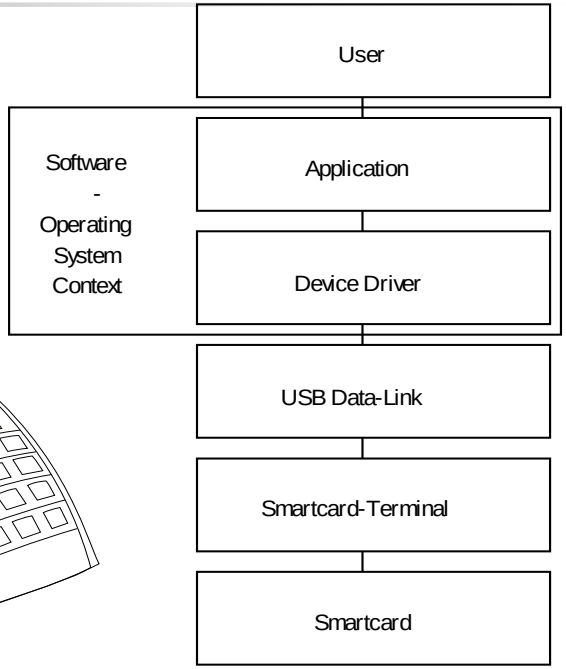


Angriff Dateierzeugung

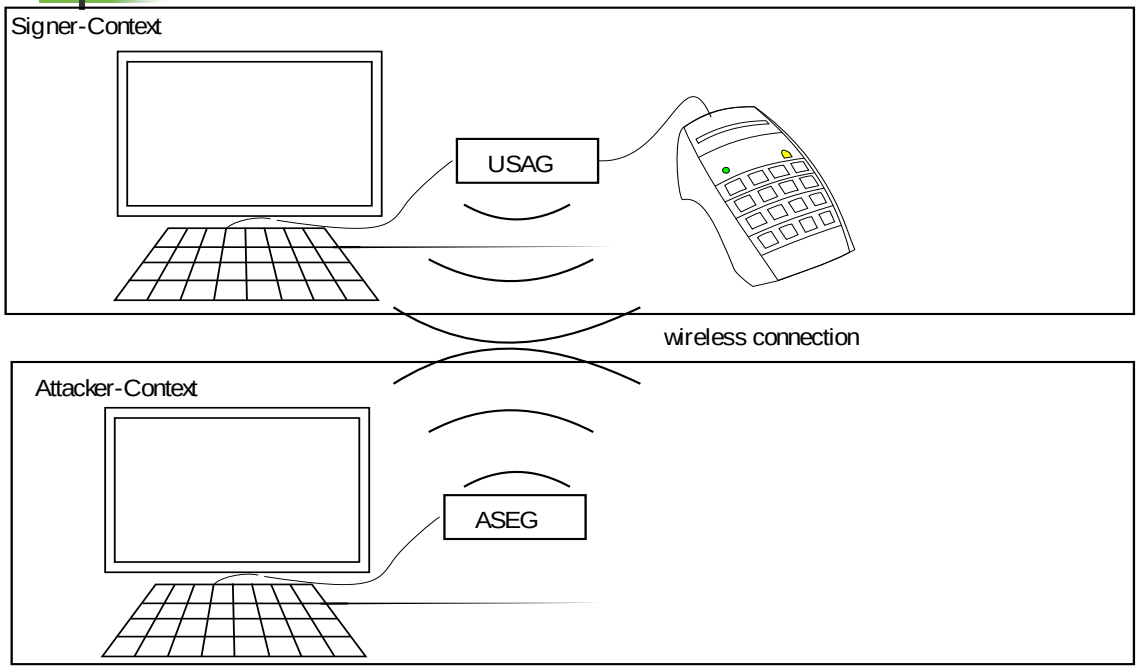


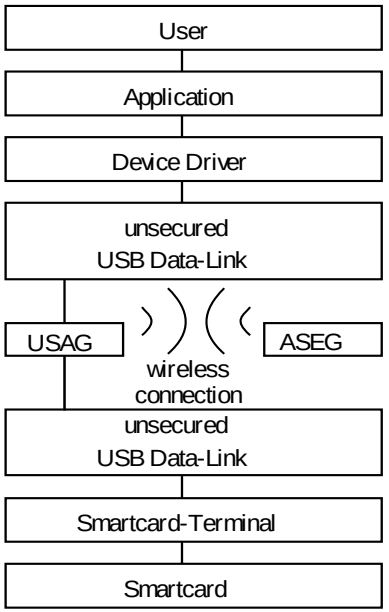
Angriff Szenario





Angriff Szenario





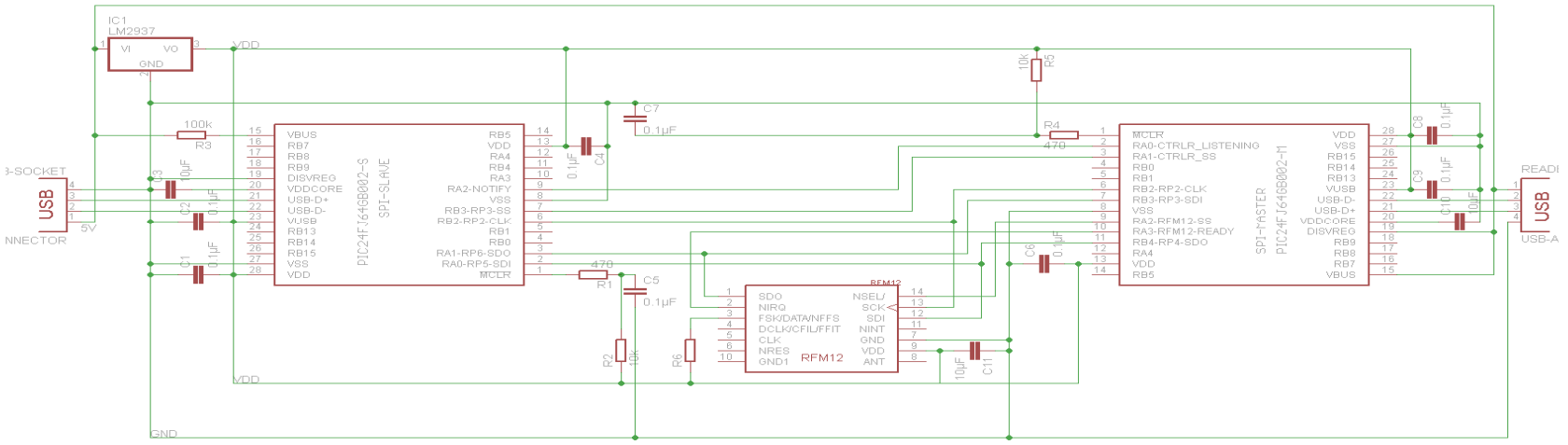
Angriff (USAG / Featureliste)

USAG: unsicheres Signatur Abgreif Gerät

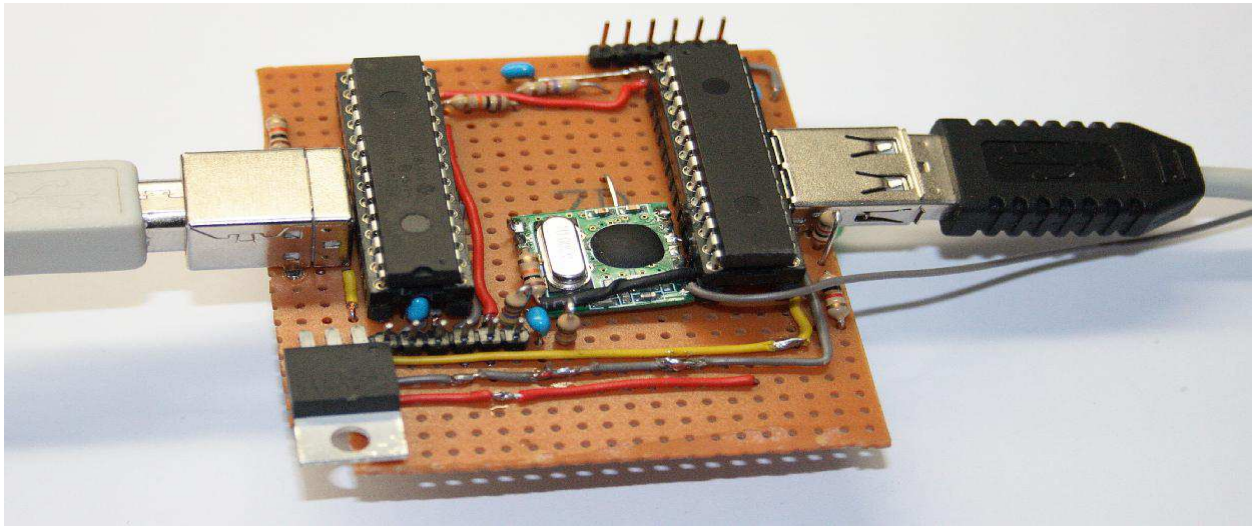
- Gerät zum Eingriff in USB-Verbindung
  - 2 Verbindungen
    - Device - Identifizierung als Kartenleser
    - Host
  - Pro Microcontroller nur eine USB-Verbindung:

    => 2 Controller + Kommunikation
- Funkanbindung für externen Zugriff

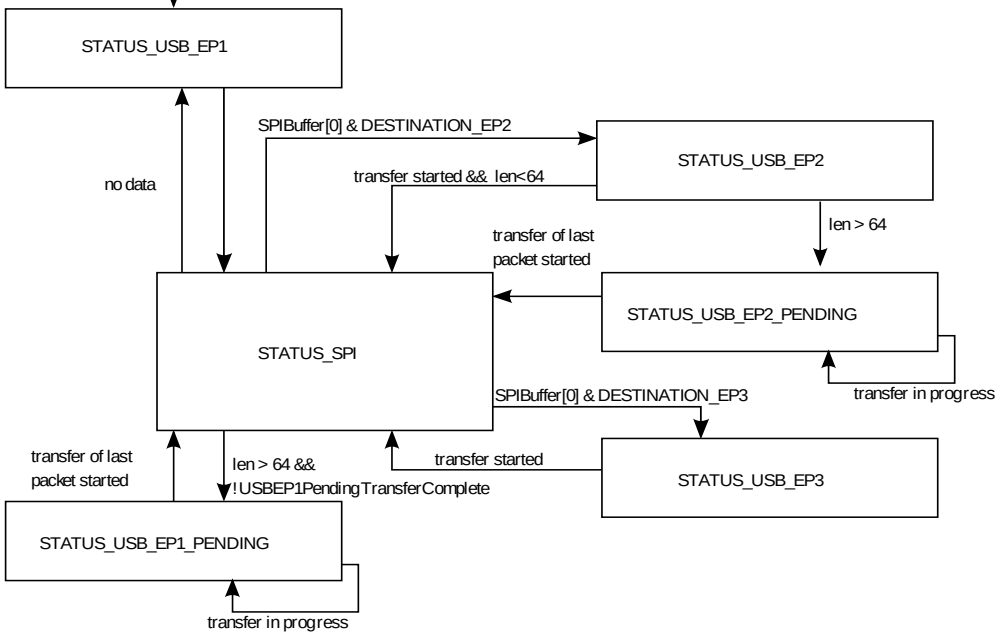
Angriff (USAG / Schaltplan)



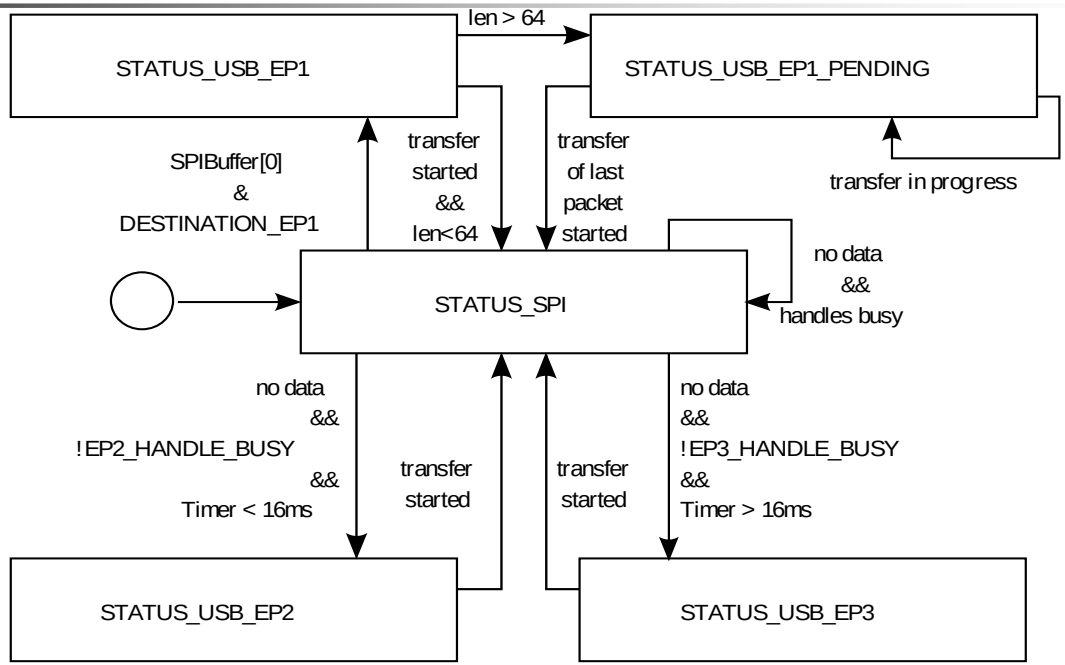
Angriff (USAG / Testmuster)



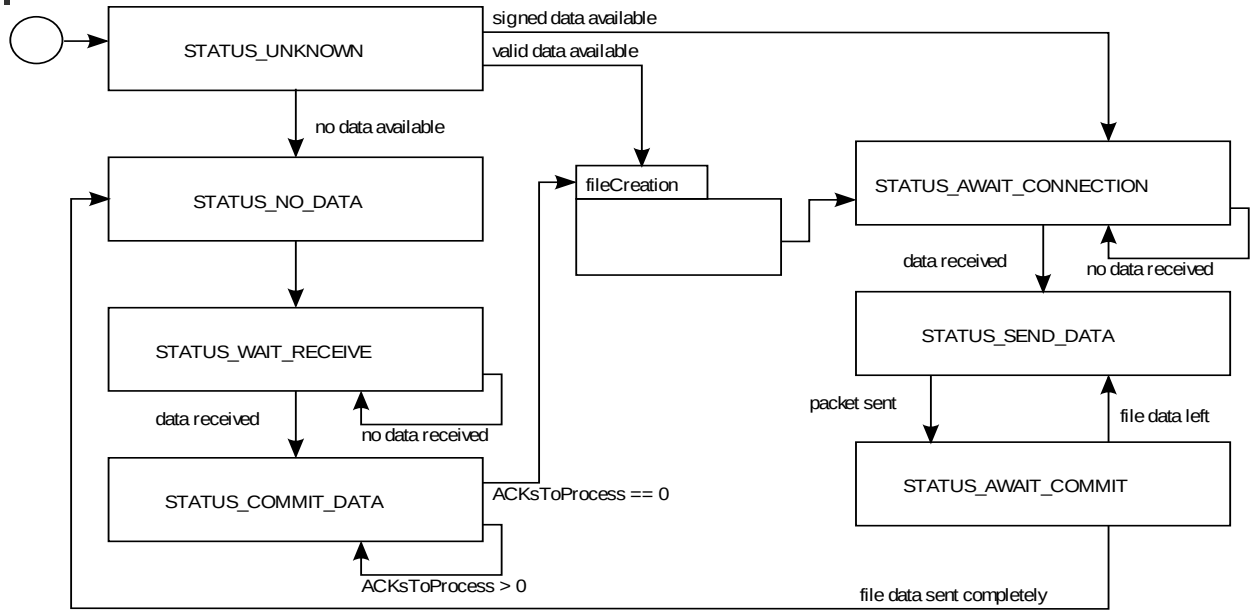
Angriff (USAG / Programmierung 1)



Angriff (USAG / Programmierung 2)



Angriff (USAG / Programmierung 3)

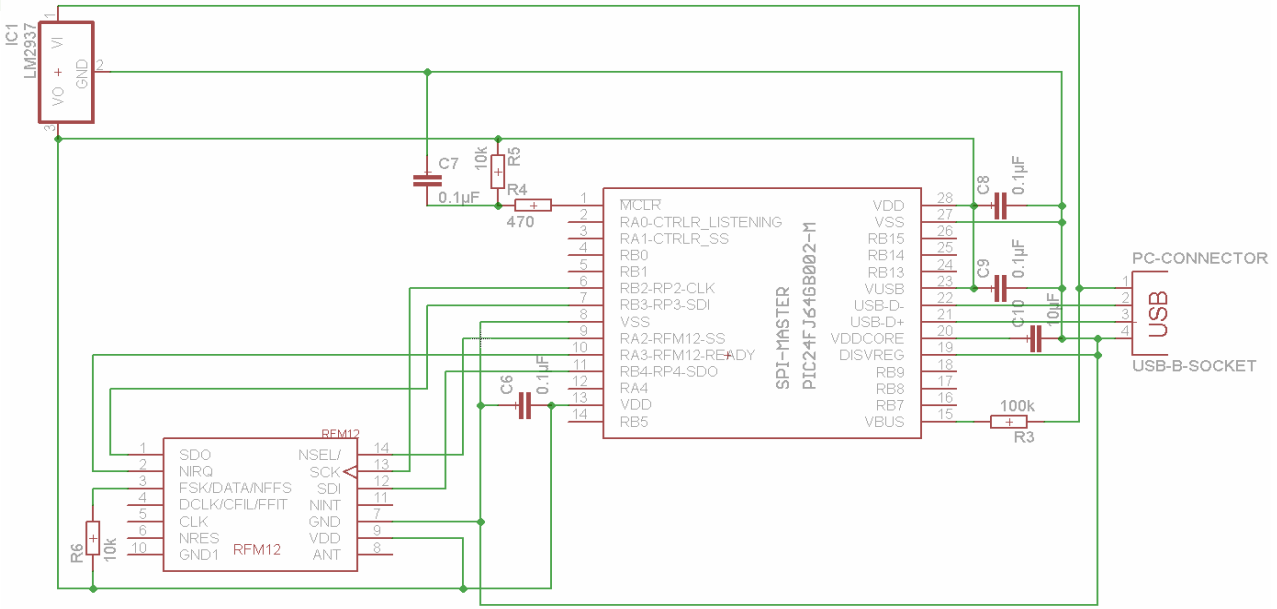


Angriff (ASEG / Featureliste)

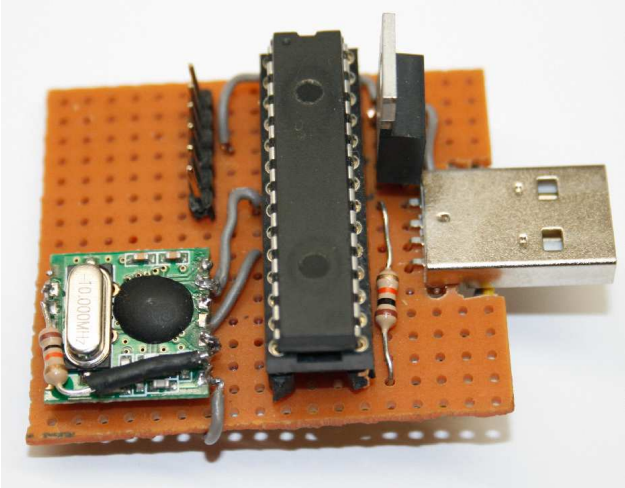
ASEG: Angreifer Signatur Erstellungs Gerät

- Verhält sich wie Mass-Storage-Device
- Sendet Daten zum USB-Stick
- Empfängt erzeugte Signatur

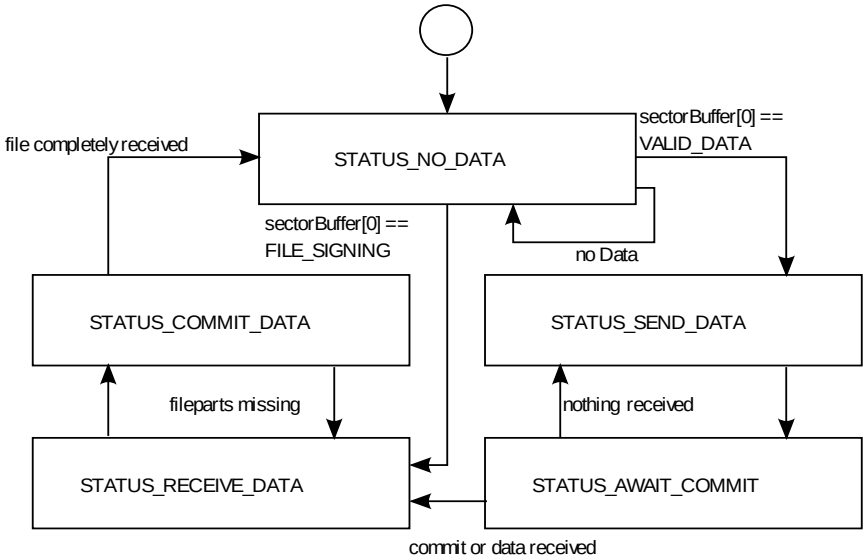
Angriff
(ASEG / Schaltplan)



Angriff
(ASEG / Testmuster)



Angriff (ASEG / Programmierung)



Angriff
Vorgehensweise

Ablauf

- USAG wird platziert
  - verhält sich transparent
  - wartet auf Funkverbindung
- ASEG wird konfiguriert
- ASEG wird in Reichweite gebracht
- Daten werden übertragen
- USAG wartet auf definierten Zustand des Kartenlesers (Rechnerneustart – hier simuliert durch
  - Abziehen des Kartenlesers
  - Neustart der Anwendungssoftware

)
- Signaturvorgang wird durchgeführt
- „PIN Falsch" - Meldung wird angezeigt – USAG verhält sich wieder transparent
- USAG wartet auf ASEG in Reichweite
- Signaturdatei wird versendet

Fazit

- Gesetzgeber hat keine Sicherung der Verbindung gefordert
- Die Datenverbindung zwischen SSEE und Anwender-PC ist also „zu Recht“ ungesichert
- auch qualifizierte elektronische Signaturen lassen sich daher fälschen
- Sichert eure Verbindungen!

Ausblick

- „ePerso“ setzt verschlüsselte Kommunikationsverbindung ein
  - hat aber noch keine Signaturfunktion
- Evtl. sind weitere Attacken möglich
  - PIN ausspähen mittels Timing-Attacke?