Fraunhofer Gesellschaft

Zurechenbarkeit von Aktionen in virtuellen Welten

Schlussbericht

Datum:	30. Juni 2004
Zustand:	Endfassung
Projekt:	ZAVIR – Zurechenbarkeit von Aktionen in virtuellen Welten
Fördernde Institution:	Bundesministerium für Bildung und Forschung (BMBF)
Förderkenn-zeichen:	01AK948A 01AK948B 01AK948C 01AK948D

Verfasser: *Olaf Henniger, Dirk Scheuermann, Björn Schneider, Bruno Struif, Ulrich Waldmann*
Fraunhofer-Institut für Sichere Telekooperation
Rheinstr. 75, 64295 Darmstadt

Rainer Ulrich
Fraunhofer-Institut für Integrierte Schaltungen
Am Wolfsmantel 33, 91058 Erlangen

Katrin Franke, Jan Schneider
Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik
Pascalstr. 8–9, 10587 Berlin

Henning Daum
Fraunhofer-Institut für Graphische Datenverarbeitung
Fraunhoferstr. 5, 64283 Darmstadt

Kurzfassung

Ausgangssituation

Voraussetzung für eine weite Akzeptanz höherwertiger E-Commerce-Vorgänge und für rechtsverbindliches Handeln über offene Netze ist die verlässliche Zurechenbarkeit von Aktionen zu Personen. Um elektronische Aktionen bestimmten Personen zurechenbar zu machen, müssen die folgenden Anforderungen erfüllt sein:

- Die Authentizität und Integrität von Daten muss sichergestellt sein. Elektronische Signaturen sind ein hervorragendes Mittel hierzu.
- Die korrekte Zuordnung von einem signierten Dokument zu seiner Bildschirmdarstellung muss gewährleistet sein. Anders ausgedrückt, der Signierer muss wissen, dass „what I see is what I sign“. Auf einem PC hat der Signierer i.allg. keine Gewähr dafür, dass er nur das signiert, was am Bildschirm angezeigt wird.
- Die tatsächliche Urheberschaft und der Rechtsbindungswillen beim Erzeugen der elektronischen Signatur müssen nachgewiesen werden. Die übliche, wissensbasierte Benutzerauthentisierung hat den Nachteil, dass das Geheimnis (die PIN) in die Hände unberechtigter Personen gelangen und missbraucht werden kann.

Projektziele

Im Projekt ZAVIR sollten eine sichere Signierumgebung, die es erlaubt, elektronische Aktionen verlässlich Personen zuzuordnen, entworfen und prototypisch implementiert werden und Schritte zur Nutzbarmachung biometrischer Verfahren zur Benutzerauthentisierung auf Signaturkarten unternommen werden.

Lösungsansatz

Trusted Signature Terminal – eine vertrauenswürdige Signierumgebung
Das Trusted Signature Terminal (TST) ist ein intelligentes Kartenterminal mit folgenden Eigenschaften [HSFU03]: Das TST

- ist stationär mit einem potentiell unsicheren PC verbunden,
- ist vor Manipulationen geschützt,
- gewährleistet eine vertrauenswürdige Präsentation zu signierender Daten,
- gestattet in Verbindung mit einer Signaturkarte, qualifizierte elektronische Signaturen zu erzeugen,
- ist mit biometrischen Komponenten zur Benutzerauthentisierung ausgestattet.

Das TST bedient sich existierender PC-Komponenten als Benutzerschnittstelle. Bildschirm, Tastatur und Maus des PCs sind auch für das TST nutzbar und werden wahlweise durch den PC oder durch das TST angesteuert. Um Manipulationen über den PC auszuschließen, werden Bildschirm, Tastatur und Maus direkt an das TST angeschlossen, das wiederum über ein

Kabelbündel mit dem PC verbunden ist. Außerhalb des Signiermodus leitet das TST die Bildschirm-, Tastatur- und Maussignale vom bzw. zum PC weiter. Im Signiermodus hingegen übernimmt das TST die alleinige Kontrolle über diese Geräte, zeigt das zu signierende Dokument mit Hilfe vertrauenswürdiger Präsentations-Software [Tang03] auf dem Bildschirm an und stellt die zur Auslösung der Signaturerzeugung erforderlichen Funktionen zur Verfügung.

Die Signaturerstellungsanwendung auf dem TST ist so programmiert, dass sie an Hand von Signaturkartenprofilen die Eigenschaften der benutzten Signaturkarte erkennt und mit unterschiedlichen Signaturkarten zusammenarbeiten kann [Sch02].

Biometrische Benutzerauthentisierung auf Signaturkarten mittels Fingerabdruckerkennung
Im Rahmen des Projekts wurde eine Signaturkarte spezifiziert [G&D03a, G&D03b] und implementiert, die neben der bisher üblichen PIN-Prüfung auch Fingerabdruck-On-Card-Matching zur Benutzerauthentisierung zulässt. Die Fingerabdruckerkennung basiert auf einem standardisierten Datenformat [DIN66400]. Die Standardisierung des Datenformats wurde aus dem ZAVIR-Projekt heraus initiiert. Die verwendete Signaturkarte ist eine STARCOS-Signaturkarte von Giesecke & Devrient mit einer Erweiterung der nach ITSEC E4 hoch evaluierten StarCert-Signaturapplikation. Im Rahmen des Projekts wurde außerdem als Grundlage für eine spätere Evaluierung der Sicherheit nach den Common Criteria ein Security-Target-Dokument für die erweiterte Signaturapplikation auf der STARCOS-Plattform erstellt [G&D03c].

Fingerabdruckdaten sind „öffentliche Daten“, d. h., ein Angreifer könnte die biometrischen Merkmale des rechtmäßigen Signaturkarteninhabers auskundschaften, daraus Verifikationsdaten ableiten und diese unter Umgehung des Sensors der Signaturkarte übergeben. Um solche Angriffe zu verhindern, wird die Kommunikation mit der Signaturkarte mit Hilfe einer in das TST integrierten Sicherheitsmodulkarte durch Secure Messaging abgesichert [Wal02, WSE03, WSE04].

Verbesserung der Zurechenbarkeit der elektronischen Signatur
Mit dem Einsatz biometrischer Verfahren lässt sich die Zurechenbarkeit von elektronischen Signaturen zu Personen erhöhen, da biometrische Merkmale personengebunden sind. Wenn dem Empfänger eines signierten Dokuments glaubhaft mitgeteilt wird, dass beim Signieren ein biometrisches Verfahren zur Benutzerauthentisierung verwendet wurde, und das verwendete Verfahren hinreichend sicher ist, steigt das Vertrauen des Empfängers, dass die Signatur tatsächlich vom rechtmäßigen Inhaber der Signaturkarte erzeugt wurde. Um dies zu ermöglichen, übermittelt die für das ZAVIR-Projekt funktionell erweiterte Signaturkarte dem TST zusammen mit der Signatur eine Mitteilung über das verwendete Benutzerauthentisierungsverfahren (biometrisch oder per PIN) [G&D03a].

Biometrische Benutzerauthentisierung auf Smartcards mittels Unterschriftserkennung
Aus Unterschriften, die auf einem grafischen Tablett aufgenommen werden, können biometrische Merkmale (die Unterschriftsdynamik) extrahiert werden, die zur Benutzer-

authentisierung verwendet werden können. Der Vorteil der Unterschriftsdynamik ist ihre hohe Benutzerakzeptanz. Handschriftliche Unterschriften sind als Mittel zur Authentisierung von Personen vielerorts seit langem akzeptiert. Darüber hinaus werden handschriftliche Unterschriften als Ausdruck einer willentlichen Entscheidung des Schreibers angesehen, da sie i. allg. nicht zufällig und unbeabsichtigt abgegeben werden. Im Rahmen einer Machbarkeitsstudie wurde ein Prototyp eines On-Card-Matching-Verfahrens zur Unterschriftserkennung entwickelt [Har02, HF03, HF04].

Untersuchung zur Überwindungssicherheit biometrischer Sensoren

Die Untersuchungen der Überwindungssicherheit von Fingerabdrucksensoren und einem Grafiktablett als Eingabegerät für die Unterschriftenerkennung ließen z. T. erhebliche Sicherheitslücken in diesem Bereich erkennen [BD02].

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

APDU	Application Protocol Data Unit
API	Application Programming Interface
BIT	Biometric Information Template
CA	Certification Authority
CPU	Central Processing Unit
EF	Elementary File
DF	Dedicated File
ITSEC	Information Technology Security Evaluation Criteria
JPEG	Joint Photograhic Experts Group
MAC	Message Authentication Code
PC	Personal Computer
PIN	Personal Identification Number
PKCS	Public-Key Cryptography Standard
PSO	Perform Security Operation
RSA	Rivest Shamir Adleman
SM	Secure Messaging
SMC	Security Module Card
SSD	Security Service Descriptor
TCG	Trusted Computing Group
TIFF	Tagged Image File Format
TPM	Trusted Platform Module
TST	Trusted Signature Terminal
USB	Universal Serial Bus
USCCM	Universal Signature Card Cryptoki Module

1 Einführung

1.1 Gliederung des Schlussberichts

Der Schlussbericht des ZAVIR-Projektes ist wie folgt gegliedert: Kapitel 1 führt in die Thematik des Projektes ein. In Kapitel 2 werden Anforderungen an das Trusted Signature Terminal (TST) analysiert, die grundlegenden technischen und organisatorischen Entscheidungen über das TST und seine Einbettung in den Einsatzkontext dargelegt und einige Entwurfsentscheidungen erläutert. Kapitel 3 geht auf das Sicherheitskonzept zum kryptografischen Schutz öffentlicher biometrischer Daten an der Chipkartenschnittstelle ein. Kapitel 4 behandelt das neue Konzept der Mitteilung des biometrischen Benutzerauthentisierungsmodus zur Erhöhung der Zurechenbarkeit elektronischer Signaturen zu Personen. In Kapitel 5 geht es um die biometrische Benutzerauthentisierung auf Smartcards mittels handschriftlicher Unterschriften. Kapitel 6 geht auf die im Rahmen des Projektes ausgeführte Untersuchung der Überwindungssicherheit biometrischer Sensoren ein. Kapitel 7 schließlich gibt eine Zusammenfassung.

1.2 Motivation

Neben Vertraulichkeit sind Integrität, Authentizität und Nichtabstreitbarkeit wichtige Anforderungen an die Sicherheit von Informationstechnik. Wenn Benutzer über Kommunikationsnetze Nachrichten austauschen, wollen sie i. allg., dass diese nicht von Unbefugten mitgehört oder gelesen werden können (Vertraulichkeit), sie wollen feststellen, ob Informationen unverändert beim Empfänger ankommen (Integrität), und sie wollen wissen, wer der Urheber von Informationen ist, ohne dass sich jemand anders als Urheber ausgeben kann (Authentizität) und ohne dass der Urheber seine Urheberschaft abstreiten kann (Nichtabstreitbarkeit). Diese Anforderungen gewinnen im Zusammenhang mit „elektronischem Handel“ (E-Commerce) und „elektronischer Verwaltung“ (E-Government) an Bedeutung, da sich die Kommunikationspartner nicht notwendigerweise kennen und vertrauen.

Mit Hilfe elektronischer Signaturen ist die Erfüllung von Sicherheitsanforderungen nach Integrität, Authentizität und Nichtabstreitbarkeit theoretisch möglich. In der Praxis reicht das Vertrauen in elektronische Signaturen jedoch gegenwärtig nicht aus, um sie für Geschäfte, bei denen es um höhere Werte geht, einzusetzen. Höherwertige elektronische Bestellungen oder Transaktionen werden heute kaum ohne zusätzliche Bestätigung z. B. per Telefon oder Fax ausgeführt. Eine Ursache ist, dass die bei der Erzeugung elektronischer Signaturen übliche, wissensbasierte Benutzerauthentisierung nur einen schwachen Nachweis der tatsächlichen Urheberschaft und des Rechtsbindungswillens bietet. Wissensbasierte Authentisierungsdaten (PIN oder Passwort) können in die Hände unberechtigter Personen gelangen. Nach der Präsentation gültiger Authentisierungsdaten werden elektronische Signaturen erzeugt, gleich-

gültig, ob die Authentisierungsdaten von einer berechtigten oder einer unberechtigten Person präsentiert wurden.

Eine weitere Ursache für die gegenwärtig mangelnde Vertrauenswürdigkeit elektronischer Signaturen ist, dass bei der Erzeugung elektronischer Signaturen nicht immer tatsächlich nur das signiert wird, was der Benutzer signieren will. Zum einen laufen die Signaturanwendung einschließlich der Dokumentenpräsentationskomponente oftmals auf PCs unter einem Betriebssystem ab, auf dem Manipulationen nicht ausgeschlossen werden können. Zum anderen können zu signierende Dokumente in Dateiformaten vorliegen, die nicht sichtbare Informationen enthalten und die auf verschiedenen Rechnern auf unterschiedliche Weise dargestellt werden können.

Zur Lösung der genannten Probleme wurden im Rahmen des Projektes ein vertrauenswürdiges, vor unberechtigten Manipulationen geschütztes Kartenterminal für Signaturkarten (Trusted Signature Terminal, TST) konzipiert und prototypisch realisiert und Beiträge zur Nutzbarmachung biometrischer Verfahren wie Fingerabdruckerkennung und On-line-Unterschriftenerkennung für die Benutzerauthentisierung auf Signaturkarten geleistet.

1.3 Stand der Technik

1.3.1 Elektronische Signaturen

Die gesetzliche Grundlage für elektronische Signaturen bildet in Deutschland das Signaturgesetz, das in seiner ersten Fassung 1997 in Kraft getreten ist und seit Mai 2001 in einer novellierten Form [SigG01] gilt. Auf der Grundlage des Signaturgesetzes wurde die Signaturverordnung [SigV01] erlassen.

Der Begriff der elektronischen Signatur ist in [SigG01] folgendermaßen definiert:

- Eine elektronische Signatur besteht allgemein aus elektronischen Daten, die zum Zwecke der Authentifizierung an andere elektronische Daten angehängt oder mit Ihnen logisch verknüpft sind.
- Eine *fortgeschrittene elektronische Signatur* ist eine elektronische Signatur mit folgenden Zusatzbedingungen:

  – ausschließliche Zuordnung zum Signaturschlüsselinhaber
  – Möglichkeit der eindeutigen Identifizierung des Signaturschlüsselinhabers
  – Erzeugung mit Mitteln, die der Signaturschlüsselinhaber unter seiner alleinigen Kontrolle halten kann und
  – eindeutige Erkennung einer nachträglichen Veränderung der signierten Daten.

- Eine *qualifizierte elektronische Signatur* ist eine fortgeschrittene elektronische Signatur, welche auf einem qualifizierten Zertifikat beruht und mit einer sicheren Signaturerstellungseinheit erzeugt wurde. Qualifizierte Zertifikate sind von Zertifizierungsdiensteanbietern, die bestimmte gesetzlich festgelegte Mindestanforderungen erfüllen, ausge-

stellte elektronische Bescheinigungen, mit denen Signaturprüfschlüssel einer Person zugeordnet werden und die Identität dieser Person bestätigt wird.

Nur mit der qualifizierten elektronischen Signatur können wie mit einer handschriftlichen Unterschrift rechtsverbindliche Erklärungen abgegeben werden. Die übrigen elektronischen Signaturen sind nicht durch das Signaturgesetz geregelt.

In diesem Projekt werden ausschließlich qualifizierte elektronische Signaturen betrachtet und zwar solche, die auf einem asymmetrischen kryptografischen Verfahren beruhen, d. h. einem Verfahren mit geheimen privaten und frei verfügbaren öffentlichen Schlüsseln, sowie einer Hashfunktion, d. h. einer Funktion, die aus beliebig langen Daten einen Hashwert berechnet, wobei es praktisch unmöglich sein muss, verschiedene Daten zu bestimmen, die den gleichen Hashwert ergeben.

Beim Erzeugen einer elektronischen Signatur verschlüsselt der Signierer den Hashwert eines Dokuments mit seinem privatem Schlüssel. Zum Prüfen der elektronischen Signatur kann jeder den mit dem privaten Schlüssel verschlüsselten Hashwert mit dem dazugehörigen öffentlichen Schlüssel entschlüsseln. Stimmt der entschlüsselte Hashwert mit dem erneut ermittelten Hashwert des vorliegenden Dokuments überein, ist bewiesen, dass die elektronische Signatur mit dem zugehörigen privaten Schlüssel erzeugt und das Dokument nicht verändert wurde. Somit sind zwei der Anforderungen an eine erweiterte elektronische Signatur (die ausschließliche Zuordnung zum Signaturschlüsselinhaber und die eindeutige Erkennbarkeit nachträglicher Veränderungen der signierten Daten) bereits erfüllt. Die alleinige Kontrolle des rechtmäßigen Signaturschlüsselinhabers über seinen privaten Schlüssel wird durch Zugriffsschutz- und Benutzerauthentisierungsmechanismen zu erreichen versucht. Die eindeutige Identifizierung des Signaturschlüsselinhabers kann durch persönliches Erscheinen und durch das Ausweisen mit dem Personalausweis oder einem ähnlichen Dokument bei der Ausstellung eines Zertifikates sichergestellt werden.

Mit Hilfe asymmetrischer kryptografischer Verfahren kann auch Vertraulichkeit erreicht werden, indem Daten mit dem öffentlichen Schlüssel des Empfängers verschlüsselt werden. Die mit dem öffentlichen Schlüssel verschlüsselten Daten können nur mit dem dazugehörigen privaten Schlüssel, also vom beabsichtigten Empfänger, entschlüsselt werden. In diesem Projekt wird nur das Erzeugen elektronischer Signaturen betrachtet, nicht jedoch das Verschlüsseln/Entschlüsseln und auch nicht das Prüfen elektronischer Signaturen.

1.3.2 Signaturkarten

Smartcards sind Chipkarten mit integriertem Mikroprozessorchip, also mit der Fähigkeit zur Datenverarbeitung direkt auf der Karte. Sie sind ein Instrument zur geschützten Bereitstellung und Ausführung sicherheitsrelevanter Funktionen (z. B. Erzeugung elektronischer Signaturen) sowie Träger von Werten (z. B. elektronische Geldbörse) oder sensitiven Daten (z. B. medizinische Daten).

Eine Signaturkarte ist eine Smartcard, die zur sicheren Aufbewahrung eines privaten Signaturschlüssels und als sichere Signaturerstellungseinheit dient. Für Signaturkarten gibt es Standards, die beschreiben, mit welchen Smartcard-Kommandos die Signaturdienstleistung erbracht wird, z. B. [DIN66291-1] oder die Office-Identity-Card-Spezifikation. Für berufsspezifische Anwendungen gibt es Spezifikationen für Karten, die ebenfalls die Signaturfunktion beinhalten, wie z. B. die Health Professional Card für Ärzte und Apotheker.

1.3.3 Kartenterminals

Ein Kartenterminal ist ein Gerät, das die elektrische Verbindung zu einer Chipkarte herstellt. Es gibt sehr unterschiedliche Ausführungen von Kartenterminals [Ran99]. Kartenterminals können nach verschiedenen Kriterien eingeteilt werden in:

- *einfache Kontaktiereinheiten* und *intelligente Kartenterminals*: Einfache Kontaktiereinheiten stellen nur die elektrische Verbindung zwischen einem Computer und einer Chipkarte her (einschließlich Takterzeugung). Intelligente Kartenterminals können selbst Informationen verarbeiten und enthalten neben der Kontaktiereinheit einen Prozessor und dazugehörigen Speicher sowie zumeist ein alphanumerisches Display und eine einfache Tastatur als Benutzerschnittstelle. Einfache Kontaktiereinheiten bieten im Falle von Manipulationen an einem potentiell unsicheren PC keinen Schutz, z. B. kann eine am PC eingegebene, für die Benutzerauthentisierung auf einer Chipkarte bestimmte PIN ausgespäht werden [Mas01]. Intelligente Kartenterminals verfügen über integrierte Schutzmechanismen, die z. B. dafür sorgen sollen, dass eine PIN direkt am Kartenterminal eingegeben werden kann, damit sie nicht an einen potentiell unsicheren PC gelangt.
- *On-line-Kartenterminals* und *Off-line-Kartenterminals*: Ein On-line-Kartenterminal hat während des Betriebs eine ununterbrochene Verbindung zu einem übergeordneten Computer. Ein Off-line-Kartenterminal hingegen arbeitet autark, ohne direkte Verbindung zu einem entfernten Computer.
- *externe Kartenterminals* und *interne Kartenterminals*: Ein externes Kartenterminal ist ein separates Gerät. Ein internes Kartenterminal ist in ein übergeordnetes System integriert, z. B. in die Tastatur oder einen Diskettenschacht eines PCs.
- *stationäre Kartenterminals* und *mobile Kartenterminals*: Ein stationäres Kartenterminal ist fest an einen Standort gebunden. Ein mobiles Kartenterminal hingegen ist tragbar und während des Betriebs nicht fest an einen Standort gebunden.
- *private bzw. firmeneigene Kartenterminals* und *Geschäftsterminals*: Ein privates bzw. firmeneigenes Kartenterminal wird innerhalb eines bestimmten Zeitraums nur von wenigen, bekannten Benutzern verwendet. Ein Geschäftsterminal wird geschäftsmäßig Dritten zur Nutzung angeboten.

Im Bankenbereich werden Kartenterminals in vier Sicherheitsklassen eingeteilt [Tak01]:

- Klasse-1-Kartenterminals sind Kartenterminals, die im wesentlichen aus einer einfachen Kontaktiereinheit bestehen. Sie bieten keinen Schutz der PIN-Eingabe und sollten nicht für Signaturkarten verwendet werden.

- Klasse-2-Kartenterminals sind Kartenterminals mit Tastatur und integrierten Schutzmechanismen, die dafür sorgen sollen, dass die PIN das Kartenterminal nicht in Richtung PC verlässt. Sie bieten zwar etwas Schutz bei der PIN-Eingabe, sollten jedoch nicht verwendet werden, wenn eine rechtsverbindliche elektronische Signatur gefordert ist.
- Klasse-3-Kartenterminals sind Kartenterminals mit Tastatur, Display und integrierten Schutzmechanismen, die in der Lage sind, einen Signaturvorgang an einer Signaturkarte zu erkennen, den Benutzer auf diesen Vorgang hinzuweisen und vor dem Erzeugen der Signatur die Bestätigung des Benutzers einzuholen. An Arbeitsplätzen, an denen auch qualifizierte elektronische Signaturen erzeugt werden, wird der Einsatz von Klasse-3-Kartenterminals empfohlen. Das Display dieser Terminals besteht in der Regel nur aus einigen wenigen Zeilen.
- Klasse-4-Kartenterminals sind Kartenterminals mit denen die Erzeugung der elektronischen Signatur in einer sicheren Umgebung bewiesen werden kann. Dazu muss die Sicherheit solcher Kartenterminals von unabhängiger Seite methodisch getestet und überprüft und eine hohe Mechanismenstärke der Sicherheitsfunktionen nachgewiesen und zertifiziert werden. Dadurch entstehen höhere Anschaffungskosten und laufende Kosten zur Pflege des Zertifikats des Kartenterminals.

1.3.4 Trusted Computing

Die Trusted Computing Group (TCG), die im April 2003 aus der Trusted Computing Platform Alliance (TCPA) hervorgegangen ist, verfolgt einen anderen Ansatz zur Lösung der in Abschnitt 1.1 genannten Sicherheitsprobleme. Die Trusted Computing Platform sieht vor, anfangs Computer mit einem zusätzlichen Chip, dem Trusted Platform Module (TPM) auszustatten und in späteren Entwicklungsstufen die TPM-Funktionen direkt in CPUs, Grafikkarten, Festplatten, Soundkarten usw. zu integrieren. Die Trusted Computing Platform verhindert, dass die auf dem Computer laufenden Anwendungen, welche abgesichert untereinander kommunizieren können, manipuliert werden können. Dadurch kann man auch keine unzertifizierte Software und Hardware mehr einsetzen. Die Trusted Computing Platform steht in der Kritik, da sie nicht so sehr die Sicherheit für die Benutzer erhöht, sondern hauptsächlich mehr Kontrollmöglichkeiten für PC-Hersteller, Software- und Content-Anbieter bietet.

2 Trusted Signature Terminal

2.1 Systemübersicht

Das Trusted Signature Terminal (TST) ist ein stationär mit einem potentiell unsicheren PC verbundenes intelligentes Kartenterminal, das mit biometrischen Komponenten zur Benutzerauthentisierung ausgestattet und vor Manipulationen geschützt ist, eine vertrauenswürdige Präsentation der zu signierenden Daten gewährleistet und es in Verbindung mit einer Signaturkarte als sicherer Signaturerstellungseinheit gestattet, qualifizierte elektronische Signaturen zu erzeugen. Das TST kann als privates bzw. firmeneigenes Kartenterminal zu Hause oder im Büro beim Signieren umfangreicher Dokumente und hochwertiger Transaktionen eingesetzt werden.

Um dem Anwender sowohl Kosten als auch Platz auf dem Schreibtisch zu sparen und dennoch große, komplexe Dokumente darstellen zu können, bedient sich das TST existierender PC-Komponenten als Benutzerschnittstelle. Bildschirm, Tastatur und Maus des PCs sind auch für das TST nutzbar und werden wahlweise durch den PC oder durch das TST angesteuert. Um im Signiermodus Manipulationen über den PC auszuschließen, werden Bildschirm, Tastatur und Maus direkt an das TST angeschlossen. Das TST braucht somit nicht wie andere höherwertige Kartenterminals über ein eigenes Display und ein eigenes Tastenfeld zu verfügen, ist jedoch mit Schnittstellen für Bildschirm, Tastatur und Maus ausgestattet. Über ein Kabelbündel, das ein Bildschirmkabel, ein Tastaturkabel, ein Kabel für die Maus und eines zur Ansteuerung des TST umfasst, ist das TST mit dem PC verbunden (siehe Abbildung 1).

Außerhalb des Signiermodus leitet das TST die Bildschirm-, Tastatur- und Maussignale vom bzw. zum PC weiter. Im Signiermodus hingegen übernimmt das TST die alleinige Kontrolle über diese Geräte. Im Signiermodus zeigt das TST mit Hilfe vertrauenswürdiger Dokumentenpräsentations-Software das zu signierende Dokument auf dem Bildschirm an und stellt die zur Auslösung der Signaturerzeugung auf einer Signaturkarte erforderlichen Funktionen zur Verfügung. Die Umschaltung des TSTs in den Signiermodus und zurück erfolgt automatisch.

Um biometrische Verfahren zur Benutzerauthentisierung auf Signaturkarten nutzen zu können, ist das TST mit einem Fingerabdrucksensor und/oder einem grafischen Tablett zur Aufnahme von handschriftlichen Unterschriften ausgestattet. Der biometrische Sensor ist in das TST-Gehäuse integriert, um weniger Angriffsmöglichkeiten zu bieten. Eine Signaturkarte, die neben der bisher üblichen PIN auch Fingerabdruck-On-Card-Matching zur Benutzerauthentisierung zulässt, wurde im Rahmen des Projekts entwickelt. Abbildung 1 zeigt die Einordnung eines TSTs in seine Umgebung.

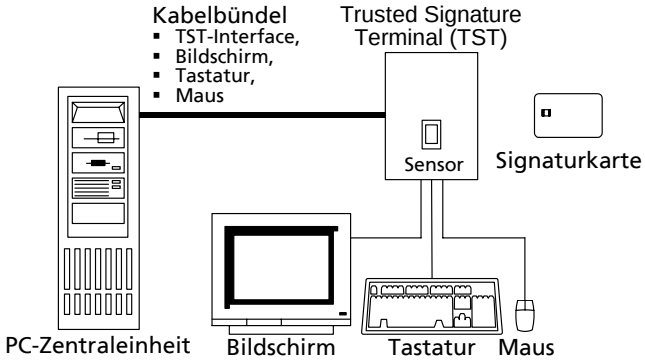


Abbildung 1 Einordnung des Trusted Signature Terminals in seine Umgebung.

2.2 Funktionalität

Die Anwendung auf dem PC stellt die Funktion „Elektronisch Signieren“ zur Verfügung. Nach dem Auslösen dieser Funktion durch den Benutzer wird zunächst das zu signierende Dokument vom PC auf das TST übertragen. Anschließend wird die Umschaltung des TST in den Signiermodus ausgelöst.

Nachdem das TST in den Signiermodus geschaltet wurde, wird das zu signierende Dokument durch die TST-Signaturanwendung auf dem Bildschirm dargestellt. Wenn der Benutzer mit dem Dokument in der angezeigten Form einverstanden ist, aktiviert er die Signierfunktion der TST-Benutzeroberfläche. Nachdem die Signaturanwendung auf dem TST ermittelt hat, was für eine Form der Benutzerauthentisierung die in den Kartenschlitz eingeführte Signaturkarte verlangt, wird der Benutzer zur Eingabe von Authentisierungsdaten aufgefordert. Wenn der Benutzer erfolgreich authentisiert wurde, berechnet die TST-Signaturanwendung den Hashwert des zu signierenden Dokuments und überträgt ihn zur Signaturkarte.

In der Signaturkarte wird die elektronische Signatur erzeugt und wieder zurück an die TST-Signaturanwendung übergeben. Neu zu implementierende Signaturkarten sollten auch auf gesicherte Weise den Benutzerauthentisierungsmodus (biometrisch und/oder wissensbasiert) mitteilen. Die elektronische Signatur wird im TST mit dem Dokument zusammengeführt und anschließend zusammen mit dem Dokument an den PC übertragen.

Nach der Übertragung des signierten Dokuments verlässt das TST den Signiermodus und Bildschirm, Tastatur und Maus werden wieder vom PC aus angesteuert. Das signierte Dokument kann vom PC aus weitergeleitet oder archiviert werden.

2.3 Sicherheitsziele

Die Sicherheitsziele, die mit einem TST angestrebt werden, sind:

- Es soll eine verlässliche Zurechenbarkeit von elektronischen Signaturen zu Personen erreicht werden.
- Es soll sichergestellt werden, dass gilt „*what you see is what you sign*“, d. h., dass nur das unterschrieben wird, was der Signierer auf dem Bildschirm gesehen und akzeptiert hat.

Eine spätere Evaluierung der Sicherheit eines TST nach den Common Criteria [ISO15408] oder den ITSEC-Kriterien [ITSEC91] sollte möglich sein.

2.4 Hardware des Prototyps

2.4.1 Überblick

Die Hardware des TST-Prototypen besteht aus einem PC-kompatiblen Einplatinenrechner inklusive Grafikkarte, einer Umschalteinheit für Maus, Tastatur und Bildschirm, zwei Smartcard-Kontaktiereinheiten, einem Fingerabdrucksensor sowie einer USB-Schnittstelle zum Anschluss weiterer biometrischer Sensoren, z. B. eines grafischen Tabletts. Die Datenübertragung vom und zum PC erfolgt über USB. Das TST wird dabei wie eine serielle Schnittstelle angesprochen. Um das TST hardwaremäßig vor Angriffen zu schützen, ist es mit einer Intrusion Detection auszustatten. Ein einfaches Intrusion-Detection-System ist in der zweiten Systemgeneration vorhanden. Durch eine Anzeige am TST, die nicht vom PC aus angesteuert werden kann, wird signalisiert, wenn sich das TST im Signiermodus befindet. So kann kein Programm auf dem PC die Ausgaben des TSTs nachbilden und dadurch den Benutzer zur Eingabe einer PIN verleiten.

Eine Integration des TST in den PC, z. B. in einen freien Laufwerksschacht, wurde verworfen. Dies hätte zwar Vorteile, wie einfachere Stromversorgung und Kühlung. Die Signalwege von Tastatur, Maus und Bildschirm wären jedoch nicht mehr einfach optisch nachzuvollziehen.

Das Netzteil, im ersten Prototyp noch im TST integriert, wanderte in der zweiten Systemgeneration aus thermischen und Platzgründen in ein eigenes Gehäuse, das nicht auf dem Schreibtisch Platz finden muss.

2.4.2 Einplatinenrechner

Aus ergonomischen Gründen muss die zusätzlich auf dem Schreibtisch zu platzierende Komponente TST möglichst klein und leise sein. Andererseits soll die rechenintensive Verarbeitung biometrischer Merkmale innerhalb kurzer Zeit erfolgen, was eine leistungsfähige CPU voraussetzt. Ein Kompromiss aus Rechenleistung und (für lüfterlosen Betrieb) möglichst kleiner Verlustleistung stellt der eingesetzte Single-Board-Computer nach ETX-Industrienorm mit einem 533 MHz Eden Prozessor dar.

2.4.3 Smartcard-Kontaktiereinheiten

Die beiden in das TST integrierten Smartcard-Kontaktiereinheiten sind über die auf dem Einplatinenrechner integrierten USB-Schnittstellen angeschlossen. Auf diese Weise können ohne großen Aufwand Komponenten aktualisiert oder durch leistungsfähigere ersetzt werden.

Eine der Smartcard-Kontaktiereinheiten ist für Karten in Standardgröße ausgelegt. In diese Kontaktiereinheit wird die Signaturkarte eingeführt. Um das TST mittels einer Sicherheitsmodulkarte gegenüber einer Signaturkarte zu authentisieren und später ein sicheres Update der Software über die USB-Schnittstelle vom PC aus durchführen zu können, besitzt das TST eine weitere Kontaktiereinheit, die für Plugin-Karten ausgelegt ist.

2.4.4 Fingerabdrucksensor

Der Fingerabdrucksensor ist in das TST-Gehäuse integriert und ebenfalls über eine auf dem Einplatinenrechner integrierte USB-Schnittstellen angeschlossen. Der im TST verwendete Sensor Veridicom FPS110 ist ein chipbasierter kapazitiver Sensor mit einer für kapazitive Sensoren relativ großen aktiven Sensorfläche von 15,2 × 15,2 mm², die das Bild des Fingerabdrucks mit 500 dpi auf 300 x 300 Pixel abbildet. Des weiteren ausschlaggebend für die Auswahl dieses Sensors für das TST ist sein gutes Kontrastverhalten sowie eine relativ geringe Anfälligkeit gegenüber latenten Fingerabdrücken, einem Phänomen, das für chipbasierte Sensoren typisch und für die automatische Bildauswertung problematisch ist. Der Sensor wurde zur Projektlaufzeit vom IGD (siehe auch Kapitel 6) bezüglich der Überwindungssicherheit als gut eingestuft („obere Klasse“ innerhalb der Menge aller getesteten Sensoren). Ein dedizierter Versuch innerhalb der Klasse der kapazitiven Sensoren war zum Zeitpunkt der Sensorauswahl allerdings noch nicht durchgeführt worden.

2.4.5 Bildschirm-/Tastatur-/Maus-Umschaltung

Eine Umschalteinheit, wie sie auch zur Steuerung von Rechnerpools Verwendung findet, schleift im Normalbetrieb die vom PC kommenden Leitungen zu Maus, Tastatur und Bildschirm durch. Im Signiermodus übernimmt das TST die Kontrolle über diese Peripheriegeräte.

Die Umschalteinheit bietet in erster Linie die Funktionalität handelsüblicher KMV (Keyboard, Mouse, Video)-Umschalter. Eine Umschaltung der Peripherie per Mausklick oder Tastatureingabe, wie bei diesen Umschaltern üblich, ist hier natürlich nicht gegeben, da dies allen Sicherheitsanforderungen entgegen stehen würde. Ein Umschalten kann nur durch ein Signal vom ETX-Single Board Computer hin erfolgen.

Da jeder PC beim Bootvorgang das Vorhandensein von Maus und Tastatur abfragt und diese auch programmiert, ist es mit einem einfachen physischen Umschalter nicht getan: Ein programmierbarer Mikrocontroller PIC14000 überwacht und interpretiert die Signale von Maus und Tastatur und simuliert die entsprechenden Signale für den gerade nicht mit der Peripherie

verbundenen Single Board Computer bzw. externen PC. Das Videosignal hingegen wird über analoge Videoverstärker umgeschaltet. Es ist daher nicht nötig, das TST und den PC auf die gleiche Videoauflösung einzustellen.

2.4.6 Intrusion Detection

Ein einfaches Intrusion-Detection-System ist in der zweiten Systemgeneration vorhanden: Öffnungsschalter am Gehäuse erkennen ein unbefugtes Öffnen des TSTs auch im stromlosen Zustand. Der Mikrocontroller schaltet daraufhin die Peripherie auf den internen Single Board Computer um, hält diesen aber durch ein permanentes RESET-Signal an, so dass keine weiteren elektronischen Signaturen mit dem Gerät mehr erzeugt werden können. Da diese Funktion an den Prototypen demonstriert werden soll, ist jedoch eine Rückstellmöglichkeit über spezielle Tastaturbefehle vorgesehen.

Eine weitergehende Intrusion Detection würde wesentlich mehr Aufwand erfordern (z. B. Eingießen aller Komponenten) und ist nicht sinnvoll für einen Prototyp, dessen Software noch geändert werden kann.

2.5 Software des Prototyps

2.5.1 Überblick

Abbildung 2 zeigt die Software-Architektur des TST und der PC-seitigen Interaktionskomponenten. In den folgenden Abschnitten werden die Software-Komponenten des TST näher beschrieben.

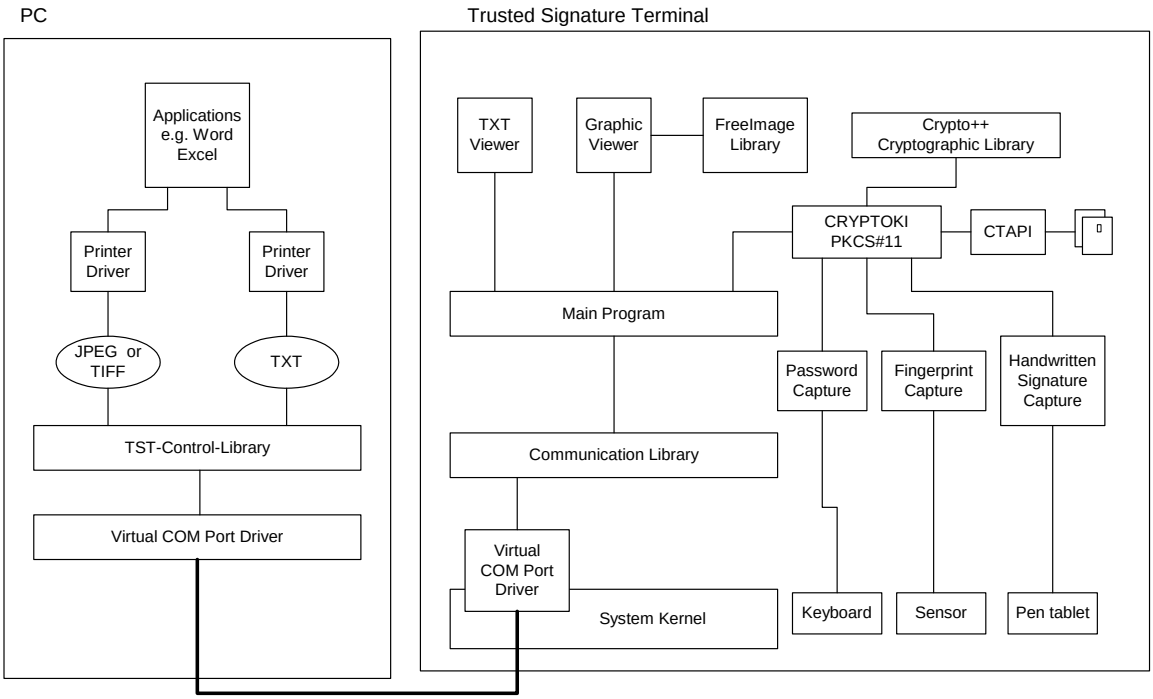


Abbildung 2 Software-Architektur

2.5.2 Betriebssystem

Auf der gewählten Hardware kommen als Betriebssystem z. B. Windows 2000, Windows XP, Windows XP Embedded oder Linux in Frage. Die meisten Hersteller von Fingerabdrucksensoren stellen standardmäßig keine Treiber unter Linux zur Verfügung, Standard ist hier Windows NT/Windows 2000. Als Betriebssystem für den TST-Prototyp wird auf Grund der Verfügbarkeit von Treibern für die biometrischen Komponenten Windows 2000 verwendet.

2.5.3 PC-seitige TST-Interaktionskomponente und TST-seitige Host-Interaktionskomponente

Das TST wird vom PC aus über Druckertreiber angesteuert. Im Druckertreiber werden zu signierende Dokumente in ein pixel-orientiertes Graphikformat wie JPEG (Joint Photograhic Experts Group) oder TIFF (Tagged Image File Format) konvertiert.

Für die Datenübertragung zwischen PC und TST wird OpenOBEX, eine Open-Source-Implementierung des ursprünglich aus der Infrarot-Technologie stammenden Object Exchange (OBEX) Protokolls [OBEX03] eingesetzt.

Um Manipulationen auszuschließen, dürfen Software-Upgrades auf dem TST nur über einen Secure-Download-Mechanismus erfolgen. Im Prototyp ist dies noch nicht implementiert.

2.5.4 Dokumentenpräsentationskomponente

Um das Sicherheitsziel „what you see is what you sign“ zu erreichen, müssen bei der Präsentation des zu signierenden Dokumentes Vorkehrungen getroffen werden, dass keine Eigenschaften des Dokumentes versteckt bleiben oder auf verschiedenen Rechnersystemen unterschiedlich dargestellt werden.

Für präsentationssensitive Dokumente gilt, dass sie unabhängig von der Rechner-Konfiguration eindeutig darstellbar sein müssen. Nicht darstellbare Informationen sind vor dem Signieren aus präsentationssensitiven Dokumenten zu entfernen. Präsentationssensitive Dokumente werden vom Druckertreiber in der PC-seitigen TST-Interaktionskomponente wahlweise in eines der pixel-orientierten Graphikformate JPEG oder TIFF konvertiert und in dieser Form von der Dokumentenpräsentationskomponente auf dem TST dargestellt und auch in dieser Form signiert. Maschinell zu verarbeitende Dokumente sollten für eine vertrauenswürdige Präsentation in einem einfachen Textformat vorliegen.

Die Dokumentenpräsentationskomponente wurde auf der Grundlage der FreeImage-Open-Source-Library [Fre03] implementiert [Tang03].

2.5.5 Universal Signature Card Cryptoki Module

Überblick

Bei dem Universal Signature Card Cryptoki Module (USCCM) handelt es sich um eine Funktionsbibliothek nach dem RSA-Labs-Industriestandard PKCS#11 [RSA01]. Sie stellt die Signaturfunktionalität einer Signaturkarte einem beliebigen Programm mit einer Schnittstelle nach PKCS#11 wie z. B. Netscape Communicator bereit. Um ein möglichst breites Feld verschiedener Signaturkarten zu unterstützen, nutzt die Bibliothek sogenannte „Kartenprofile“, die auf der ASN.1-Definition nach [ISO7816-15] basieren und um zusätzliche benötigte Informationen erweitert sind.

Die USCCM-Bibliothek basiert auf einer ebenfalls in diesem Projekt entwickelten, objektorientierten Abstraktion des Cryptoki-Interfaces, die es vereinfachen soll, die kryptografischen Funktionen zu implementieren. Cryptoki (Cryptographic Token Interface) ist die in PKCS#11 spezifizierte Schnittstelle zum Austausch kryptografischer Daten. Unter einem Token versteht man ein kryptografisches Gerät wie z. B. eine Signaturkarte.

Universal Cryptoki Framework

Slot-Abstraktion
Bei einem „Slot“ handelt es sich um die Repräsentation der Verbindung zu einem Token, dem eigentlichen Kryptografie-Modul. Eine Instanz der „CSlot“-Klasse, bzw. einer davon abgeleiteten Klasse, bildet jeweils genau einen solchen Slot ab. Die „CSlot“-Klasse selbst bietet noch keinerlei kryptografische Funktionalität, es lassen sich jedoch „Dummy“-Instanzen von ihr bilden, die sich lediglich als „Datenspeicher“ nutzen lassen, indem man Session-Objekte erzeugt, die (im Gegensatz zu Tokenobjekten) nicht persistent gespeichert werden.

Um einen funktionsfähigen Slot zu erstellen, implementiert man eine neue Klasse, die von der Interfaceklasse „CSlot“ erbt. Je nach Verwendungszweck müssen die entsprechenden, virtuellen Methoden der Klasse überschrieben werden; für eine elektronische Signatur wären dies z. B. „C_SignInit()“, „C_Sign()“, „C_SignUpdate()“ und „C_SignFinal()“. Aber auch die grundlegenden Informationsfunktionen, die Auskunft über die Fähigkeiten („Mechanismen“) des Tokens geben, sind in diesem Fall zwingend erforderlich.

Bei beliebig aktivierbaren Hardware-Token, wie den in diesem Projekt verwendeten Signaturkarten, ist die eigentliche Verwaltung der Vorgänge (Einstecken, Auslesen der Fähigkeiten des Tokens, Deaktivieren bei Herausziehen) eine zusätzliche Aufgabe der Klasse. Dies kann z. B. durch das Hinzuerben eines Thread-Interfaces erfolgen, dessen Implementation die verschiedenen Zustände des Slots überwacht; je nach Zustand müssen die Funktionen entsprechende Rückmeldungen liefern.

Mit den Initialisierungsmethoden, über die jede Klasse verfügen muss, werden beim Initialisieren der Cryptoki-Bibliothek automatisch Instanzen der Klasse erzeugt. Die Anzahl und Konfiguration der Klasse obliegt dabei ausschließlich dieser Initialisierungsfunktion; so kann

diese auf eine Konfigurationsdatei oder die Registry zugreifen, oder dynamisch z.B. das PC/SC-Framework nach Kartenlesern durchsuchen. Die so instanziierten Klassen sind automatisch an der Bibliothekschnittstelle verfügbar (C_GetSlotList(), C_GetSlotInfo()).

Cryptoki-Objekte
Die Cryptoki-Objekte werden von einem weiteren Klassensystem verwaltet, das aus der Klasse „CCryptokiObject“, die exakt ein Objekt repräsentiert, und dem „Consistency-Checker“, der die Datenkonsistenz dieser Objekte sicherstellt, besteht. In diesem „ConsistencyChecker“ sind die dazu notwendigen Regeln nach PKCS#11 vordefiniert, können jedoch jederzeit vom Slot erweitert (für neue, spezifische Objekte) oder ersetzt werden.

Wird durch den Benutzer ein neues Objekt erstellt, so wird das angegebene Template zunächst vom Standard-Konsistenzprüfer des Frameworks überprüft, wobei fehlende Werte mit Defaultwerten soweit möglich aufgefüllt werden. Anschließend wird es von der Slot-Implementation geprüft und u.U. abgelehnt (z.B. bei einem Versuch der Erzeugung eines Objekts auf einem schreibgeschützten Token). Diese Funktion ist außerdem für die weitere Verarbeitung zuständig, wie z.B. das Schreiben des Objekts auf das Token.

Bei der Suche nach einem Objekt kommt die generische Implementation der Objekte zur Wirkung: jedes Objekt besitzt eine Funktion, die das Objekt auf eine Übereinstimmung mit einem übergebenen Suchmuster prüft. Auf diese Weise kann durch ein einfaches Iterieren über den „Objektspeicher“ des Slots die Suchfunktion implementiert werden, die außerdem den Zustand der Session (öffentlich oder privat) berücksichtigt. Diese Funktionalität ist bereits in der Grundklasse „CSlot“ enthalten und muss nicht neu implementiert werden.

Ebenso sind die anderen Objektfunktionen bereits implementiert und müssen ggf. nur noch geringfügig erweitert werden; Session-Objekte verwalten sie in der Grundimplementation bereits eigenständig.

Sessions
Jede Session eines Slots wird durch eine eigene Klasseninstanz einer CSession- oder davon abgeleiteten Klasse repräsentiert, die direkt einem Slot zugeordnet ist. Sie enthält außerdem alle Informationen der CK_SESSION_INFO-Struktur sowie den „Speicherplatz“ für den aktuellen Zustand einer Operation (dies kann mit C_Get/SetOperationState() verwendet werden, um eine kryptografische Operation auszusetzen, um sie später an dieser Stelle fortsetzen zu können; dies hängt jedoch vom Token ab).

Abhängigkeiten zu anderer Software

Um die USCCM-Bibliothek möglichst flexibel und portierbar zu halten, wurde, soweit möglich, für systemspezifische Operationen und die grafische Benutzeroberfläche das wxWidgets-Abstraktionsframework verwendet [wxWid], welches Microsoft Windows. Mac-

OS X sowie verschiedene Linux und BSD Systeme unterstützt. Es wurde jedoch bisher nur eine Microsoft-Windows-Implementation genutzt.

Unterstützte Cryptoki-Funktionen

Die USCCM-Bibliothek unterstützt die in Tabelle 1 mit einem ✓ versehenen Cryptoki-Funktionen, die mit einem ✘ gekennzeichneten Funktionen werden nicht unterstützt:

Tabelle 1 Unterstützte Cryptoki-Funktionen

Funktionsname	Unterstützung	Beschreibung
Allgemeine Funktionen		
`C_Initialize`	✓	Initialisiert die Bibliothek, startet Threads etc.
`C_Finalize`	✓	Stoppt alle noch aktiven Vorgänge und bereitet die Bibliothek zum Entladen vor
`C_GetInfo`	✓	Gibt allgemeine Funktionen zurück (Name, Versionen von Soft- und Hardware)
`C_GetFunctionList`	✓	Gibt eine Struktur mit den Einsprungszeigern aller Funktionen zurück
Slot- und Token-Managementfunktionen		
`C_GetSlotList`	✓	Gibt die Anzahl bzw. die ID-Nummern der verfügbaren Slots zurück
`C_GetSlotInfo`	✓	Gibt Informationen über einen bestimmten Slot zurück (Name, Hersteller, Typ, Versionen)
`C_GetTokenInfo`	✓	Gibt Informationen über ein Token (hier eine Signaturkarte) zurück (Bezeichnung, Hersteller, Modell, Seriennummer, ...)
`C_WaitForSlotEvent`	✘	Zeigt einen aktiven Slot (z. B. beim Einstecken einer Signaturkarte) an; blockiert auch auf Wunsch
`C_GetMechanismList`	✓	Gibt eine Liste aller Mechanismus-IDs zurück, die von einem Token unterstützt werden
`C_GetMechanismInfo`	✓	Gibt genauere Informationen über einen bestimmten Mechanismus preis.
`C_InitToken`	✘	Initialisiert, d.h. richtet ein Token neu ein
`C_InitPIN`	✘	Erzeugt die (Benutzer-) PIN
`C_SetPIN`	✘	Ändert die (Benutzer-) PIN
Session-Managementfunktionen		
`C_OpenSession`	✓	Öffnet eine Session zu einem bestimmten Token
`C_CloseSession`	✓	Schließt eine bestimmte Session
`C_CloseAllSession`	✓	Schließt *alle* Sessions zu einem bestimmten Token
`C_GetSessionInfo`	✓	Gibt Informationen zu einer bestimmten Session zurück (wie Benutzerstatus und den letzten aufgetretenen Fehler
`C_GetOperationState`	✘	Gibt den Zustand eines Vorganges (Entschlüsselung, Signatur, …)

		mit allen notwendigen Daten zurück, um die Operation später mit `C_SetOperationState` fortzuführen
`C_SetOperationState`	✘	Restauriert einen zuvor begonnenen und mit `C_GetOperation-State` gesicherten Vorgang
`C_Login`	✓	Setzt die Rechte einer Session neu (Benutzer- oder Supervisor-Rechte)
`C_Logout`	✓	Löscht die Rechte einer Session zurück zu "public"
Objekt-Managementfunktionen		
`C_CreateObject`	✓	Erzeugt ein neues PKCS#11-Objekt
`C_CopyObject`	✓	Kopiert ein Objekt
`C_DestroyObject`	✓	Löscht ein Objekt
`C_GetObjectSize`	✓	Gibt die benötigte Speichergröße eines Objekts in Byte zurück
`C_GetAttributeValue`	✓	Gibt – sofern es die Session-Rechte zulassen – ein oder mehrere Attribute eines bestimmten Objekts zurück
`C_SetAttributeValue`	✓	Setzt – sofern es die Session-Rechte zulassen – ein oder mehrere Attribute eines bestimmten Objekts
`C_FindObjectsInit`	✓	Startet eine Suche nach Objekten, die einem bestimmten Muster entsprechen
`C_FindObjects`	✓	Liefert bei Aufruf eine Liste der gefundenen Objekte zurück; kann mehrmals aufgerufen werden, um weitere zu finden
`C_FindObjectsFinal`	✓	Beendet eine Suchoperation
Verschlüsselungsfunktionen		
`C_EncryptInit`	✘	Initialisiert einen Verschlüsselungsvorgang
`C_Encrypt`	✘	Verschlüsselt ein Datenpaket und beendet die Operation mit der Rückgabe der verschlüsselten Daten
`C_EncryptUpdate`	✘	Verschlüsselt die übergebenen Daten
`C_EncryptFinal`	✘	Beendet die Operation und liefert die letzten, verschlüsselten Daten zurück.
Entschlüsselungsfunktionen		
`C_DecryptInit`	✘	Initialisiert einen Entschlüsselungsvorgang
`C_Decrypt`	✘	Entschlüsselt ein Datenpaket und beendet die Operation mit der Rückgabe der entschlüsselten Daten
`C_DecryptUpdate`	✘	Entschlüsselt die übergebenen Daten
`C_DecryptFinal`	✘	Beendet die Operation und liefert die letzten, entschlüsselten Daten zurück.
Hashfunktionen		
`C_DigestInit`	✘	Initialisiert einen Hashvorgang
`C_Digest`	✘	Hasht ein Datenpaket und beendet die Operation mit der Rückgabe des Hashwertes
`C_DigestUpdate`	✘	Übernimmt die übergebenen Daten in die Hashberechnung
`C_DigestKey`	✘	Übernimmt die Byte-Darstellung des angegeben Schlüssels in die Hashwertberechnung
`C_DigestFinal`	✘	Beendet die Operation und liefert den Hash zurück.

Signatur- und MAC-Funktionen		
C_SignInit	✓	Initialisiert eine Signaturerzeugung
C_Sign	✓	Transformiert die übergebenen Daten je nach Einstellung zu einer Signatur und beendet den Signaturvorgang
C_SignUpdate	✓	Übernimmt die übergebenen Daten je nach Einstellung zur Signaturerstellung
C_SignFinal	✓	Beendet den Signaturvorgang und gibt die Signatur zurück
C_SignRecoverInit	✓	Initialisiert eine Signaturerzeugung, bei der die Nachricht aus der Signatur selbst rekonstruiert wird
C_SignRecover	✓	Transformiert die übergebenen Daten direkt in eine Signatur
Funktionen zur Verifizierung von Signaturen und MACs		
C_VerifyInit	✘	Initialisiert eine Signaturüberprüfung
C_Verify	✘	Verifiziert die übergebene Signatur mit den ebenfalls übergebenen Daten und beendet die Operation
C_VerifyUpdate	✘	Übernimmt Daten zur Signaturberechnung
C_VerifyFinal	✘	Verifiziert die übergebene Signatur und beendet die Operation
C_VerifyRecoverInit	✘	Initialisiert eine Verifikation, bei der die Nachricht aus der Signatur selbst wiedergewonnen wird
C_VerifyRecover	✘	Verifiziert die Signatur und gibt die enthaltene Nachricht zurück
kryptografische Doppel-Funktionen		
C_DigestEncryptUpdate	✘	Führt zugleich eine Hash- sowie eine Verschlüsselungsoperation auf den übergebenen Daten durch; beide müssen zuvor initialisiert worden sein.
C_DecryptDigestUpdate	✘	Führt eine Entschlüsselungs- und danach eine Hashoperation auf den entschlüsselten Daten durch; beide Operationen müssen zuvor initialisiert worden sein.
C_SignEncryptUpdate	✘	Führt zugleich eine Signatur- sowie eine Verschlüsselungsoperation auf den übergebenen Daten durch; beide müssen zuvor initialisiert worden sein.
C_DecryptVerifyUpdate	✘	Führt eine Entschlüsselungs- und danach eine Verifikationsoperation auf den entschlüsselten Daten durch; beide Operationen müssen zuvor initialisiert worden sein.
Schlüssel-Managementfunktionen		
C_GenerateKey	✘	Erzeugt einen neuen (symmetrischen) Schlüssel
C_GenerateKeyPair	✘	Erzeugt ein neues Schlüsselpaar (asymmetrische Verfahren)
C_WrapKey	✘	Verschlüsselt einen (z.B. temporären) Schlüssel mit einem anderen.
C_UnwrapKey	✘	Entschlüsselt einen Schlüssel mit Hilfe eines anderen
C_DeriveKey	✘	Leitet einen (z.B. temporären) Schlüssel von einem Basisschlüssel mit Hilfe der übergebenen Zusatzinformationen ab
Zufallszahlenfunktionen		
C_SeedRandom	✘	Nutzt die übergebenen Daten (zusätzlich) dazu, einen Pseudozufallsgenerator zu initialisieren
C_GenerateRandom	✘	Gibt eine zufällige Byte-Folge der gewünschten Länge zurück

Managementfunktionen für parallel laufende Funktionen		
`C_GetFunctionStatus`	✘	Obsolet. Liefert lediglich eine Fehlermeldung zurück.
`C_CancelFunction`	✘	Obsolet. Liefert lediglich eine Fehlermeldung zurück.
Callback-Funktionen		
„Surrender Callbacks“	✘	Während einer Operation kann die PKCS#11-Bibliothek immer wieder eine übergebene Funktion der Hauptanwendung aufrufen, um zum einen eine Art Feedback zum Benutzer zu bieten, zum anderen, es ihm zu ermöglichen die Operation abzubrechen
„herstellereigene Callbacks“	✘	Können beliebig genutzt werden; bei einem unbekannten Callback sollte die Hauptanwendung „OK“ zurückgeben

Unterstützte Algorithmen

Die USCCM-Bibliothek unterstützt die folgenden Hashalgorithmen unabhängig von der verwendeten Smartcard:

- SHA-1
- RIPEMD-160

Als Signaturalgorithmus wird derzeit nur RSA unterstützt; technikbedingt hängt dies zudem von der verwendeten Smartcard ab.

Anpassung an verschiedene Signaturkarten

Um Interoperabilität mit verschiedenen Signaturkarten bieten zu können, muss das TST an Hand von Datenobjekten, die auf der jeweils eingeführten Signaturkarte bereitgestellt werden, die besonderen Eigenschaften der Karte ermitteln (z. B. das unterstützte Benutzerauthentisierungsverfahren).

Mittlerweile gibt es mehrere deutsche Zertifizierungsdiensteanbieter, die ihre jeweiligen Signaturkarten beliebig gestalten können. Fast alle verfügbaren Signaturkarten basieren auf [DIN66291-1]. [DINV66291-1] definiert eine Hauptinformationsdatei auf der Karte, den sog. Security Service Descriptor (SSD). Darin sind die Befehle für die einzelnen Vorgänge wie Benutzerauthentisierung und Erzeugen einer elektronischen Signatur explizit aufgeführt. Auch sind zu jedem Vorgang zusätzliche Informationen wie z. B. das benutzte Zertifikat und Algorithmen aufgeführt.

Es ist davon auszugehen, dass zukünftige Signaturkarten anstatt [DIN66291-1] den internationalen Standard [ISO7816-15] unterstützen werden. [ISO7816-15] definiert Cryptographic Information Objects und gibt die Struktur von Informationen vor, die eine kryptografische Anwendung auf der Smartcard beschreiben. Zu diesen Informationen gehören z. B. die unterstützten Algorithmen, Speicherorte von Zertifikaten, Referenzierung von Schlüsseln und das Format der PIN/Passwort oder biometrischen Authentisierung..

Die Bibliothek zur Ansteuerung der Signaturkarten muss mindestens mit den beiden Standards [DINV66291-1] und [ISO7816-15] umgehen können und sollte auf möglichst einfache Weise um andere Normen oder proprietäre Systeme erweitert werden können.

Leider unterscheiden sich die Karten zudem in der Befehlsfolge einer Operation, was nicht in den Cryptographic Information Objects dargestellt werden kann. Daher wurde eine Erweiterung entwickelt [Sch02], die einen weiteren Datensatz vorsieht, in dem die Kartenbefehle (APDUs) bis auf Byte-Ebene genau spezifiziert werden können. Außerdem wird deren genaue Reihenfolge der APDUs festgelegt.

Diese Zusatzinformationen werden nicht von der Karte bereitgestellt, sondern für bestimmte Signaturkarten erstellt und im TST als „Kartenprofil“ gespeichert. Beim Einstecken einer Smartcard identifiziert das TST die Karte und lädt das dazugehörige Profil.

Benutzerauthentisierung

Die Bibliothek unterstützt zwei Arten der Benutzerauthentisierung gegenüber dem Token (SmartCard):

- PIN/Passwort
- Fingerabdruck-Biometrie

Dabei verlangt die Bibliothek zunächst keinerlei Authentisierung; der zum Nutzen der kryptografischen Operationen notwendige Login-Vorgang wird ohne Angabe eines Passwortes ausgeführt (es sind zu diesem Zeitpunkt jedoch keine Zugriffe auf geschützte Funktionen der Karte möglich).

Erst zum Zeitpunkt der Signaturerstellung (Aufruf von „C_Sign()“ oder „C_SignFinal()“ an der Bibliotheksschnittstelle) wird die eigentliche Überprüfung des Benutzers vorgenommen.

Bei der PIN/Passwort-Authentisierung wird der Nutzer über einen Dialog nach seiner PIN/seinem Passwort gefragt. Seine Antwort wird nach einer Typüberprüfung entsprechend des unterstützten PIN-Formats der Karte formatiert und der Karte übergeben.

Bei der biometrischen Authentisierung hingegen, die jedoch nur mit einer entsprechend befähigten Smartcard möglich ist, wird zunächst ein Dialog gestartet, in dem der Benutzer aufgefordert wird, seinen Finger auf den Sensor des Terminals zu legen. Dieser wird erfasst, verarbeitet und in ein zu [DIN66400] konformes Format transformiert. Dieses Template wird daraufhin an die Karte gesendet, die es eigenständig überprüft.

Um die Authentizität des Datentemplates sicherzustellen und z. B. Replay- und Datenakquisitions-Angriffe zu verhindern, wird zur Übertragung ein gesicherter Kanal zwischen TST und der Signaturkarte aufgebaut (siehe Kapitel 3).

2.5.6 Kryptografische Bibliothek Crypto++

Die USCCM-Bibliothek greift auf Funktionen der kryptografischen Bibliothek Crypto++ zurück. Die Crypto++-Klassenbibliothek ist eine freie Implementation verschiedener, kryptografischer Algorithmen, die als Public-Domain-Quellcode (nicht urheberrechtsgeschützt) zur Verfügung stehen.

2.5.7 Chipkartenübertragungsmodul

Die Chipkarten werden über die CT-API-DLL der Chipkarten-Kontaktiereinheiten von Omnikey angesprochen.

2.5.8 Fingerabdruck-Merkmalsextraktion

Prinzip

Für die automatische Erkennung von Fingerabdrücken wird der Verlauf der Papillarlinien der Fingerkuppen als biometrisches Merkmal verwendet. Diese Linien sind bei jeder Person einmalig und nicht vererbbar. Sie sind darüber hinaus – von Verletzungen abgesehen – auf Lebenszeit invariant; bei älteren Personen verhärtet sich lediglich die Haut der Fingerkuppe. Der weltweite Verbreitungsgrad des biometrischen Merkmals Fingerabdruck ist sehr groß. Außerdem ist in der Regel ein Ausweichen auf einen anderen Finger möglich, wenn der von einem Erkennungssystem primär geforderte Finger nicht vorhanden ist (der linke Zeigefinger beispielsweise fehlt bei 0,62% der Weltbevölkerung).

Aus dem Linienmuster der Fingerkuppe können nun sowohl lokale als auch globale Merkmale extrahiert werden. Lokale Merkmale, die sogenannten Minutien, werden aus den Verzweigungen und Endungen der Linien abgeleitet. Globale Merkmale beschreiben das charakteristische Muster der Papillarlinien. Es existieren die sechs Grundmuster flacher und spitzer Bogen, linke und rechte Schleife, Doppelschleife und Wirbel sowie aus diesen Grundtypen zusammengesetzte Muster.

Die allgemeine Funktionsweise aller Fingerabdrucksensoren ist unabhängig vom zugrundeliegenden physikalischen Prinzip ähnlich. Jeder Sensor erfasst das zweidimensionale Abbild der dreidimensionalen Objektstruktur der Fingerlinien. Je ausgeprägter diese 3D-Struktur ist, desto kontrastreicher ist das 2D-Abbild. Physikalische Wirkprinzipien derzeit gängiger Sensoren sind:

- optisch (prismen- oder chipbasiert, auch berührungslos)
- kapazitiv
- e-Feld
- drucksensitiv
- thermisch

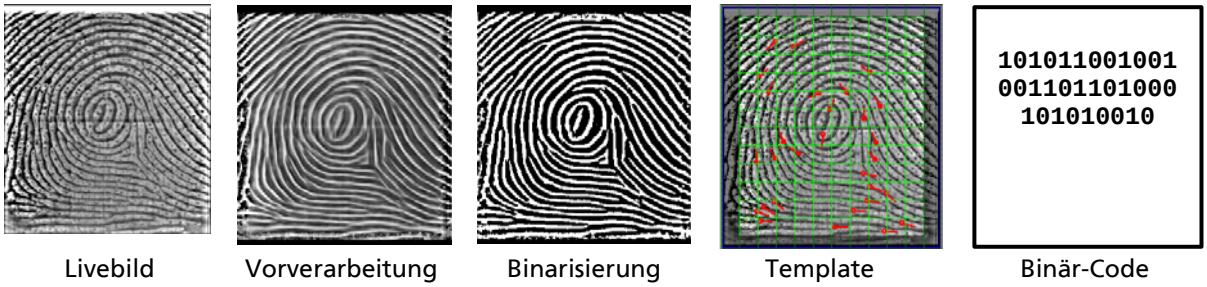


Abbildung 3 Fingerabdruck-Merkmalsextraktion

Sensoren mit den vier letztgenannten Wirkprinzipien sind chipbasiert. Ein CCD- oder CMOS-Chip wirkt dabei direkt als aktive Sensorfläche, d. h. es ist kein Linsensystem erforderlich.

Fingerabdruck-Merkmalsextraktion im TST

Aufbauend auf den am IPK entwickelten Verfahren zur Fingerabdruck-Merkmalsextraktion und -Erkennung wurde während der Projektlaufzeit eine Version entwickelt, die in der Smartcard-Umgebung des TST einsetzbar ist. Dabei erfolgt die Merkmalserfassung sowie die Merkmalsextraktion außerhalb der Karte auf dem TST.Für die Integration des Veridicom FPS110 Fingerabdrucksensors wurde eine Software-Schnittstelle zwischen Sensor und Merkmalserfassung implementiert. Des weiteren wurde die Software zur Bildvorverarbeitung und zur Merkmalsextraktion entsprechend adaptiert.

Der Vergleich der Merkmalssätze zweier Fingerabdrücke erfolgt vollständig auf der von Giesecke & Devrient implementierten Smartcard. Dafür wurde das IPK-Fingerabdruck-Merkmalsextraktions-System in das TST eingebettet und eine Schnittstelle, die ein nachfolgendes On-Card-Matching ermöglicht, entwickelt.

Als Templateformat wurde das auf der eingesetzten Smartcard verwendete Format nach [DIN66400] gewählt. Die Implementierung der IPK-Softwarebibliothek zur Fingerabdruckerkennung wurde daher entsprechend um Routinen zur Konvertierung der proprietären IPK-Templatestruktur in das DIN-Format erweitert. Unterstützt werden derzeit die beiden Formate „Standard“ und „Compact“, im speziellen „Ordered Ascending, Cartesian x-y“.

Für die biometrische Authentisierung am TST wurde eine auf der IPK-Software zur Fingerabdruckerkennung aufsetzende Bibliothek mit grafischer Benutzeroberfläche (siehe Abbildung 4) implementiert, auf die von der TST-Rahmenapplikation über eine ANSI-C-Programmierschnittstelle zugegriffen werden kann. Neben der grafischen Oberfläche mit umfangreicher Visualisierung des aufgenommenen Fingerabdrucks sowie aller extrahierten Merkmale, besitzt die Bibliothek folgende Funktionalität:

- Fingerabdruck-Erfassung und Merkmals-Extraktion,
- Enrollment eines Benutzers,
- Verifikation eines Benutzers,
- Identifikation eines Benutzers.

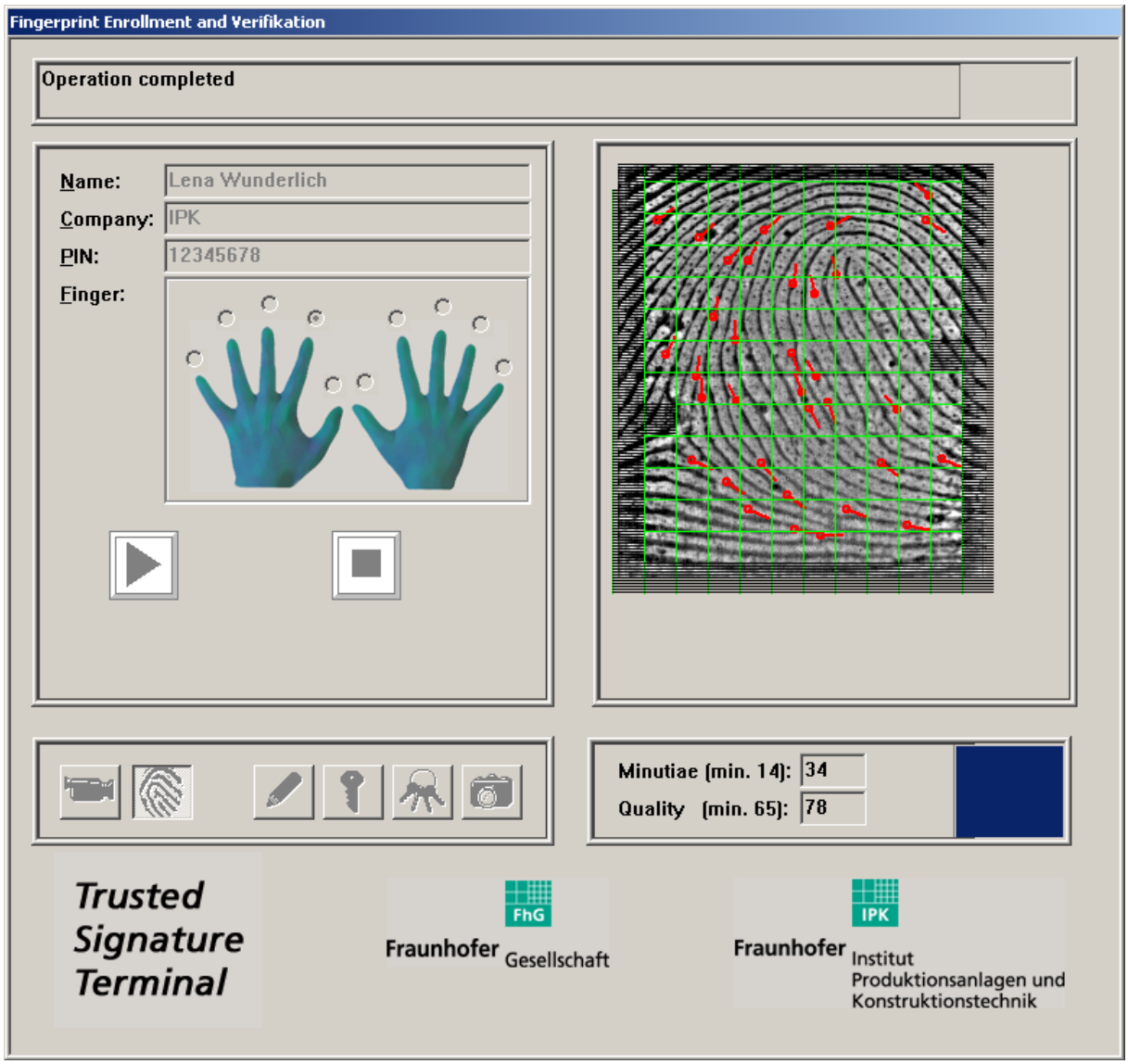


Abbildung 4 Biometrische Authentisierung am TST

Die Bibliothek besitzt eine flexibel parametrierbare und speziell auf die TST-Rahmensoftware zugeschnittene Schnittstelle. Die Merkmalsextraktion kann so konfiguriert werden, dass sowohl die IPK-eigene, proprietäre Templatestruktur als auch eine zu [DIN66400] kompatible Templatestruktur generiert werden kann. Unterstützt werden hierbei derzeit sowohl Standard- als auch Compact-Format.

Die Implementierung der Verifikation bzw. Identifikation eines Benutzers in der Bibliothek, also nicht On-Card, stellte eine kartenunabhängige Übergangslösung dar, die im finalen Prototypen des TST aber nicht mehr benötigt wird.

Die Bibliothek kann zusätzlich stand-alone, d. h. ohne TST-Rahmensoftware, betrieben werden, um DIN-kompatible Fingerabdruck-Templates aufzunehmen. Dies ist z. B. für Tests der Funktionalität des On-Card-Matching erforderlich.

3 Schutz biometrischer Daten

3.1 Einleitung

Die biometrische Benutzer-Authentisierung stellt eine Alternative zur wissensbasierten Authentisierung mittels PIN oder Passwort dar. Eine Authentisierung, sei sie wissensbasiert oder biometriebasiert, erfordert einen Vergleich von zuvor abgelegten Referenzdaten mit den aktuell erhobenen bzw. vorgelegten Daten, den sogenannten Verifikationsdaten, wie einem Passwort oder einem Fingerabdruck. Um mögliche Sicherheitsbedrohungen auszuschließen, die sich aus einer entfernten Überprüfungen der Authentifizierungsdaten ergeben können, also zum Beispiel einer Überprüfung in einem speziellen Gerät mit nachfolgender Übertragung an die Smartcard, ist der biometrische Vergleichsalgorithmus auf der Smartcard implementiert (On-Card-Matching).

Ein allgemeines Sicherheitsproblem biometrischer Verfahren, insbesondere der Fingerabdruck-Erkennung, ist der Umstand, dass biometrische Daten im Gegensatz zur PIN nicht als geheime Daten gelten können. Sie können leicht ausgespäht oder beschafft und unter Umgehung des biometrischen Sensors zu einem Missbrauch von Sicherheitsanwendungen verwendet werden. Die Verifikationsdaten sollten folglich der Smartcard kryptografisch geschützt übergeben werden, damit ihre unmittelbare Herkunft vom Sensor nachweisbar ist. Bei den bisherigen Oncard-Vergleichsverfahren mit Fingerabdruck-Daten werden die Verifikationsdaten ungeschützt der Smartcard übergeben. Als eine Sicherheitskomponente des TST wurde das im Folgenden beschriebene Konzept zur Lösung des dargestellten Problems implementiert. Die Datenübertragung vom biometrischen Sensor zur Smartcard erhält einen kryptografischen Schutz, mittels dessen die Smartcard die gegenwärtige Präsentation, die Authentizität und Integrität der Daten überprüfen kann. Das Sicherheitskonzept zum Schutz biometrischer Daten ist für firmeneigene oder private Dienstleistungssysteme und Smartcards mit sicherheitsrelevanten Anwendungen (z. B. zur Erzeugung elektronischer Signaturen), die mit biometrischer Benutzerauthentisierung geschützt werden, geeignet.

3.2 Rechtliche Aspekte

In letzter Zeit sind einige neue Rechtsvorschriften geschaffen worden, die den Einsatz von Biometrie vorsehen. In Deutschland sind mit dem neuen Signaturgesetz [SigG01] und der neuen Signaturverordnung [SigV01] in Bezug auf die Authentisierung von Personen erstmals die rechtlichen Voraussetzungen für die Gleichstellung biometrischer Methoden gegenüber den wissensbasierten Verfahren geschaffen worden. Die Biometrie ist danach als vollwertige Alternative zur wissensbasierten Authentisierung zulässig, falls die biometrischen Komponenten nach den ITSEC-Evaluierungskriterien die Mechanismenstärke „hoch“ erreichen. Wird nur die Mechanismenstärke „mittel“ erreicht, so ist die Biometrie – wie bisher in der

alten Signaturverordnung – nur als sekundäre Methode (nach anfänglicher einmaliger wissensbasierter Authentisierung) zur Freischaltung weiterer Signaturen zulässig.

3.3 Sicherheitsstandards

In Bezug auf Sicherheit gibt es neben allgemeinen Standards für IT-Sicherheit (z.B. [ITSEC], [ISO15408]) und für spezielle Sicherheitsanwendungen (z. B. Signaturgesetze) keine technischen Sicherheitsstandards für biometrische Verifikationssysteme und keine standardisierten Testmethoden, um die Systeme nach den Sicherheitskriterien zu evaluieren. Prinzipiell ist zwar festgelegt, was die Mechanismenstärke „hoch“ hinsichtlich der biometrischen Benutzer-Authentisierung bedeutet; es gibt jedoch keine standardisierten Methoden zur verlässlichen Bestimmung von Mechanismenstärken. Zudem existieren noch keine Standards für biometrische Vergleichsalgorithmen und ihre jeweiligen Toleranzgrenzen, d.h. es gibt keine eindeutig festgelegten mathematischen Kriterien für eine Übereinstimmung biometrischer Daten.

3.4 Sicherheitsprobleme bei biometrischen Daten

3.4.1 Annahmen

Der biometrische Sensor stellt den ersten Angriffspunkt eines biometrischen Systems dar, das in ein Dienstleistungssystem, integriert ist. Viele der heute verbreiteten optischen und kapazitiven Fingerabdrucksensoren können durch Täuschung mit einer Attrappe hintergangen werden [MMJH02]. Die Verhinderung einer direkten Täuschung des Sensors war jedoch nicht Gegenstand des ZAVIR-Projektes, sondern es wurde von der Annahme ausgegangen, dass überwindungssichere Fingerabdruck-Sensoren mit Lebenderkennung verfügbar werden. Im Folgenden sind Sicherheitsprobleme beschrieben, bei denen ein Angreifer das biometrische Modul umgeht (so genanntes Bypassing), um einer fremden Benutzerkarte eine authentische biometrische Benutzerauthentisierung vorzutäuschen.

3.4.2 Datenakquisitionsangriff

Fingerabdrucksdaten sind als öffentlich anzusehen, weil sie von Personen meist unwillentlich und unmerklich auf berührten Gegenständen hinterlassen werden. Beim so genannten Datenakquisitionsangriff wird ein solcher Fingerabdruck, beispielsweise von einem benutzten Glas wie in Abbildung 5, aufgezeichnet. Nach dem Übertragen des Abdrucks etwa auf Papier, können die Daten eingescannt und digitalisiert werden. Es wird angenommen, dass ein professioneller Angreifer die Datenformate des Systems kennt, auch wenn diese proprietär sind und vom Hersteller nicht veröffentlich werden, und aus den Rohdaten entsprechende Verifikationsdaten generieren kann.

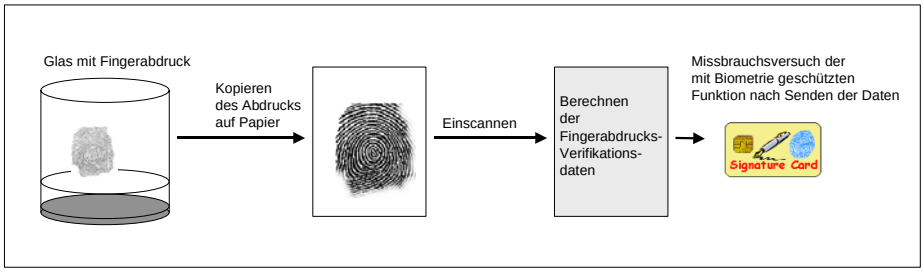


Abbildung 5 Datenakquisitions-Angriff

Wenn kein Sicherheitsmechanismus vorhanden ist, um die direkte Herkunft der Verifikationsdaten vom Sensor nachzuweisen, wird die Benutzerkarte sich beim Empfang der Daten in derselben Weise verhalten, als ob die Daten aktuell vom Sensor aufgenommen und von der Merkmalsextraktion verarbeitet wurden. Der Sicherheitsstatus einer erfolgreichen biometrischen Benutzerverifikation würde auf der Karte gesetzt werden und damit den Missbrauch einer geschützten Funktion ermöglichen.

3.4.3 Replay-Angriff

Eine weitere Angriffsmöglichkeit besteht, wenn die Systemkonfiguration einen Zugang zur Datenleitung ermöglicht, auf der die biometrischen Verifikationsdaten übertragen werden. In diesem Fall kann ein Angreifer die Daten einer erfolgreichen Benutzerverifikation aufzeichnen und sie für eine so genannte Replay-Angriff verwenden, nachdem er die zugehörige Benutzerkarte dem rechtmäßigen Besitzer entwendet oder zumindest zeitweise in Besitz genommen hat. Bekanntlich lassen viele Benutzer ihre Smartcard den ganzen Tag im Kartenleser, um nicht jedes Mal aufs Neue die Karte bei einem Authentisierungsvorgang einführen zu müssen. Der Angreifer sendet die aufgezeichneten Daten unter Umgehung des biometrischen Sensors erneut an die Benutzerkarte und könnte anschließend ebenfalls die Kartenfunktion missbrauchen.

3.4.4 Man-in-the-Middle-Angriff

Bei einem so genannten Man-in-the-Middle-Angriff sind die Aktionen des Angreifers zugleich an die Systemumgebung und an die Benutzerkarte gerichtet. Der Angreifer manipuliert die Kommunikation zwischen den beiden Systemkomponenten Kartenterminal und Benutzerkarte, indem er den Datenstrom zwischen den Komponenten abfängt und zu beiden Seiten manipulierte oder ganz neu generierte Befehle bzw. Antworten weitersendet. Möglicherweise wird auf keiner Seite bemerkt, dass die Kommunikation in Wirklichkeit mit einem falschen Partner stattfindet. Gerade bei der Etablierung von Sicherheitsvorkehrungen muss ein solcher Angriff ausgeschlossen werden können. Auch die Manipulation des Verifikationsergebnisses auf dem Weg von der Benutzerkarte zum Terminal fällt in diese Kategorie von Angriffen.

3.4.5 Systemarchitektur und Angriffspunkte

Das Sicherheitskonzept gewährleistet die Authentizität und Integrität der öffentlichen Fingerabdruck-Daten. Ein Angreifer im Besitz einer fremden Benutzerkarte, eines Terminals mit biometrischem Modul und der fremden Verifikationsdaten kann der Benutzerkarte diese Verifikationsdaten nicht mehr erfolgreich präsentieren. Abbildung 6 zeigt ein Beispiel der Systemarchitektur eines geschützten Dienstleistungssystems. Ein solches System stellt Benutzern eine Anwendung zur Verfügung, die durch biometrische Benutzerauthentisierung geschützt ist. Beispiele wären Systeme in Apotheken oder Arztpraxen, bei denen nur berechtigte Personen Lese- und Schreibzugriff auf bestimmte Daten haben sollen. Ein weiteres Beispiel ist das TST, an dem ein privater Benutzer Dokumente in elektronischer Form signieren kann.

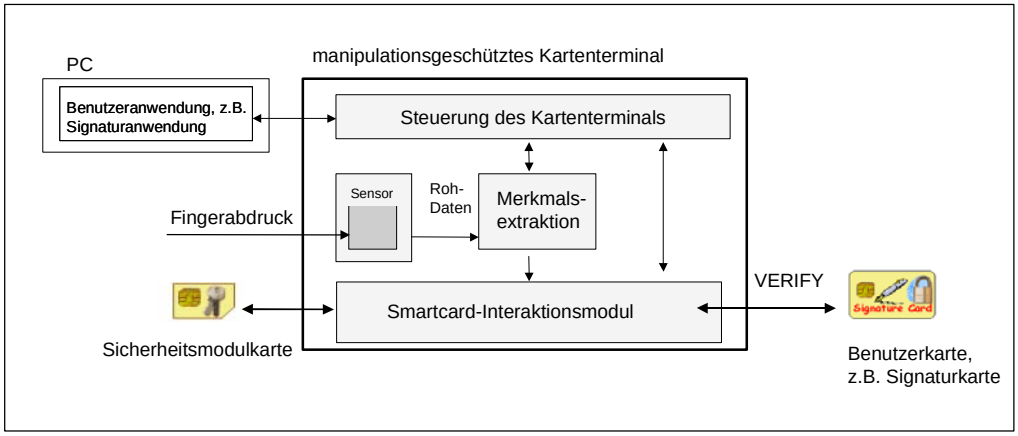


Abbildung 6 Systemarchitektur des Dienstleistungssystems mit Smartcard-Schnittstellen (Beispiel)

Das System besteht aus einem PC mit der eigentlichen Dienstleistungsanwendung und einem manipulationsgeschützten Kartenterminal. Im Terminal sind der Fingerabdruck-Sensor und die Merkmalsextraktion als biometrisches Modul integriert. Der Benutzer muss sich vor der Nutzung der Dienstleistung mit einem Fingerabdruck authentisieren. Auf der Benutzerkarte befindet sich eine wesentliche Teilfunktion der Dienstleistung (z. B. das Signieren eines Hashwertes für eine Signatur-Dienstleistung), die nur nach einer erfolgreicher Benutzerauthentisierung ausgeführt werden kann. Das Terminal enthält ein Smartcard-Interaktionsmodul, über das ein Sicherheitsprotokoll zwischen einer Sicherheitsmodulkarte (Security Module Card, SMC) und der Benutzerkarte abläuft.

Die SMC ist als PlugIn-Karte konzipiert, um dem Hersteller des Dienstleistungssystems einen flexiblen Umgang mit dieser Komponente zu ermöglichen. Es wäre auch denkbar, die für die SMC vorgesehene Anwendung zum Schutz der biometrischen Daten vollständig im Terminal zu integrieren. Die vorgeschlagene Konzeption ermöglicht den Herstellern aber, je nach Bedarf zusätzliche Sicherheitsapplikationen oder Neuerungen in die SMC zu integrieren. Eine SMC kann beispielsweise leicht ausgewechselt werden, wenn ein neues Zertifikat genutzt werden soll. Die SMC und die Benutzerkarte besitzen jeweils ein eigenes Authentisierungs-Zertifikat, wodurch eine dynamische Erzeugung von kryptografischen Schlüsseln möglich ist.

Beide Karten haben kryptografische Funktionen zur Berechnung und Prüfung der SM-Datenobjekte, die in den geschützten Befehlen an die Benutzerkarte und in deren Antworten verwendet werden. Das Kartenterminal hat dagegen konzeptuell nur statische Komponenten, die zur Erhöhung der Sicherheit möglichst in Hardware realisiert sein sollten.

Die biometrischen Daten werden im Dienstleistungssystem zu Verifikationsdaten verarbeitet. An der Schnittstelle zur Benutzerkarte werden diese Verifikationsdaten verschlüsselt und mit einer kryptografischen Prüfsumme versehen in einem VERIFY-Kommando der Benutzerkarte übergeben. Der kryptografische Schutz der Verifikationsdaten wird in der SMC berechnet. Im Gegensatz zur Benutzerkarten-Schnittstelle, an der Interoperabilität erforderlich ist, handelt es sich bei der Schnittstelle des Smartcard-Interaktionsmoduls zur SMC um eine interne herstellerspezifische Schnittstelle. Zusammenfassend zeigt die Abbildung 7 die Schnittstelle zur SMC und zur Benutzerkarte mit den oben genannten möglichen Angriffen, deren Problematik durch das Sicherheitskonzept gelöst wird.

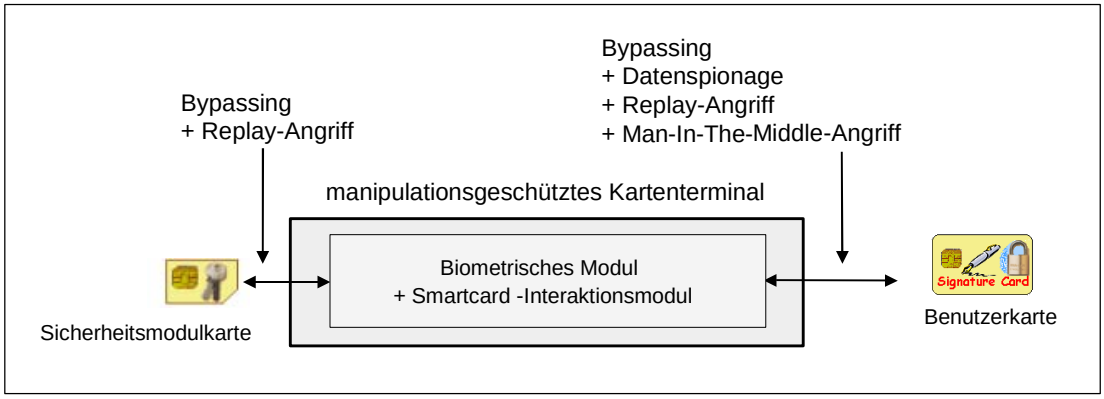


Abbildung 7 Mögliche Angriffsversuche an den Smartcard-Schnittstellen des Dienstleistungssystems

3.5 Realisierung eines Trusted Channels als Sicherheitslösung

3.5.1 Eigenschaften des Trusted Channels

Aus Sicht der Sicherheitsanforderungen für die Anwendung des Dienstleistungssystems ist ein „Trusted Channel“ zwischen Sicherheitsmodulkarte und Benutzerkarte erforderlich, der folgende Eigenschaften haben muss:

- die Kommandos an die Benutzerkarte müssen authentisch sein, d.h. die Karte muss sicher sein können, dass die Kommandos aus autorisierter Quelle stammen
- die Kommandos an die Benutzerkarte dürfen nicht verändert worden sein, d. h. die Integrität der Kommandos muss verifizierbar sein
- es dürfen keine Kommandos unbemerkt einfügbar sein, d. h. sowohl neu konstruierte als auch in einer Session zuvor gesendete (gesicherte) Kommandos müssen als unzulässig erkannt werden können
- das Unterdrücken von Kommandos muss feststellbar sein.

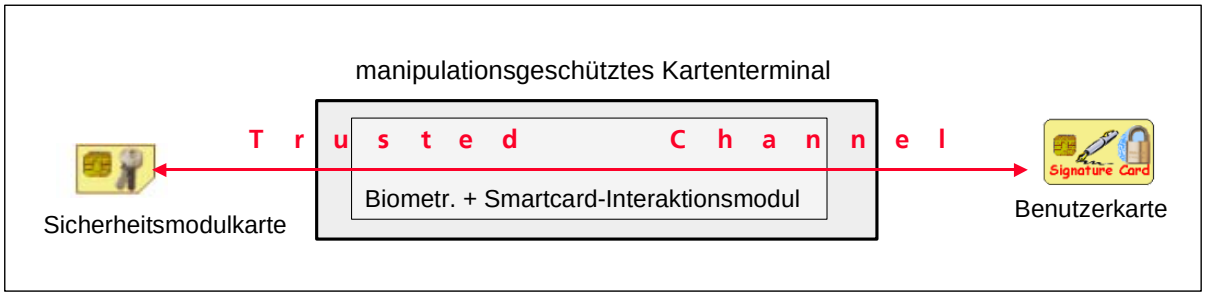


Abbildung 8 Trusted Channel zwischen Sicherheitsmodulkarte und Benutzerkarte

3.5.2 Realisierung des Trusted Channels mittels Secure Messaging

Die Realisierung von Trusted Channels wird technisch mit dem in [ISO7816-4] definierten Konzept des „Secure Messaging (SM)“ umgesetzt. Ein SM-Kommando hat einen Aufbau wie in Abbildung 9 dargestellt.

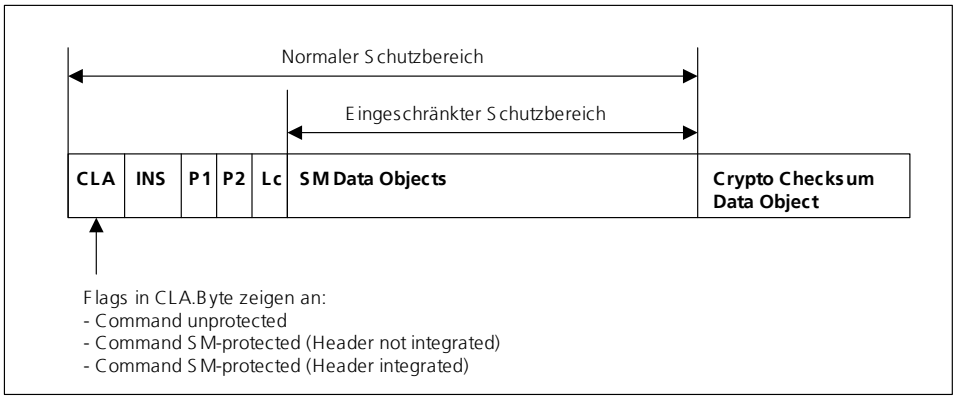


Abbildung 9 SM-geschütztes Kommando

Wie aus Abbildung 9 ersichtlich, wird ein SM-Kommando durch eine kryptografische Prüfsumme geschützt, wobei der Schutzbereich sich nach ISO/IEC 7816-4 entweder über das gesamte Kommando oder nur über das Datenfeld desselben erstreckt. Für die Signaturkarte des TST werden SM-Kommandos verwendet, in denen das gesamte Kommando geschützt ist. Die SM-Data-Objects bestehen im wesentlichen entweder aus einem Plain Value (PV) Data Object, das die Daten im Klartext enthält, die sonst ungeschützt gesendet werden, oder einem Cryptogram Data Object, das diese Daten dann in verschlüsselter Form enthält.

Zur Berechnung der kryptografischen Prüfsumme und des Kryptogramms werden zwei verschiedene SM-Schlüssel, so genannte Session Keys, verwendet, die zuvor in der Regel im Rahmen eines gegenseitigen Authentisierungsverfahrens ausgehandelt werden. Hierzu werden nach ISO/IEC 7816-4 entweder symmetrische oder asymmetrische Authentisierungsverfahren verwendet. Im TST wird das asymmetrische Verfahren RSA mit einer Schlüssellänge von 1024 bit verwendet. Die ausgehandelten Session Keys sind symmetrische Schlüssel, als Verschlüsselungsverfahren wird DES-3 benutzt.

Bestandteil des Authentisierungsverfahrens ist, wie in ISO/IEC 7816-4 vorgesehen, die Vereinbarung des Startwerts für den „Send Sequence Counter“, der vor jeder Berechnung einer Prüfsumme um 1 erhöht wird und der in deren Berechnung mit eingeht, so dass die Unterdrückung von Kommandos erkannt werden kann. Erkennt die Smartcard einen Fehler, d. h., wurde ein Kommando manipuliert, eingefügt oder unterdrückt, dann werden die SM-Schlüssel in der Karte gelöscht. Die mittels biometrischer Benutzerauthentisierung geschützte Funktion der Benutzerkarte (z. B. die Signaturfunktion) kann daraufhin innerhalb der Session nicht mehr ausgeführt werden.

Das SM-Konzept, also insbesondere der Aufbau eines SM-Kommandos und einer SM-Response sind in [ISO7816-4] und [ISO7816-8] klar beschrieben, die Erzeugung eines SM-Kommandos und die Behandlung der SM-Response im Kartenterminal durch ein geeignetes Sicherheitsmodul werden jedoch nur sehr rudimentär angedeutet. Im TST übernimmt die Sicherheitsmodulkarte diese Aufgaben, in ähnlicher Weise wie die in [HPC04] spezifizierte SMC. Für die Erzeugung von SM-Kommandos und der Prüfung der SM-Response werden die PSO-Kommandos COMPUTE CC, VERIFY CC, ENCIPHER und DECIPHER verwendet. Das Verfahren braucht für die Erzeugung der Befehle an die SMC spezielle Konstruktionsmittel, die über die übliche Nutzung der betreffenden PERFORM SECURITY OPERATIONS (PSO-Kommandos) hinausgehen. Abbildung 10 und Abbildung 11 zeigen das mehrstufige PSO-Verfahren.

3.6 Verfahren der gegenseitigen Authentisierung

Zu Beginn jeder Sitzung vor der eigentlichen Benutzer-Authentisierung findet eine kryptografische Prüfung an den beiden Smartcard-Schnittstellen statt. Die SMC (auf Seiten des Terminals mit dem biometrischen Modul) und die Benutzerkarte führen auf RSA-Basis eine gegenseitige Geräte-Authentisierung nach [DIN66291-1] durch. Beide Seiten tauschen kartenverifizierbare Authentisierungs-Zertifikate aus, die ihre öffentlichen RSA-Schlüssel enthalten. Die zugehörigen privaten RSA-Schlüssel bleiben auf den Karten sicher verwahrt. Die Authentisierungsdaten enthalten u. a. eine Zufallszahl der prüfenden Karte und eine Zufallszahl der sich authentisierenden Karte. Die Daten werden einschließlich ihres Hashwerts mit dem privaten RSA-Schlüssel der sich authentisierenden Karte signiert und mit dem öffentlichen RSA-Schlüssel der prüfenden Karte verschlüsselt. Nach einer erfolgreichen Prüfung der Authentisierungsdaten berechnet die jeweils prüfende Karte aus diesen und den eigenen Authentisierungsdaten

- den symmetrischen Session Key SK.CG für die Berechnung von Kryptogrammen,
- den symmetrischen Session Key SK.CC für die Berechnung kryptografischer Prüfsummen,
- und den Initialwert des Send Sequence Counters, der in die Berechnung von kryptografischen Prüfsummen eingeht.

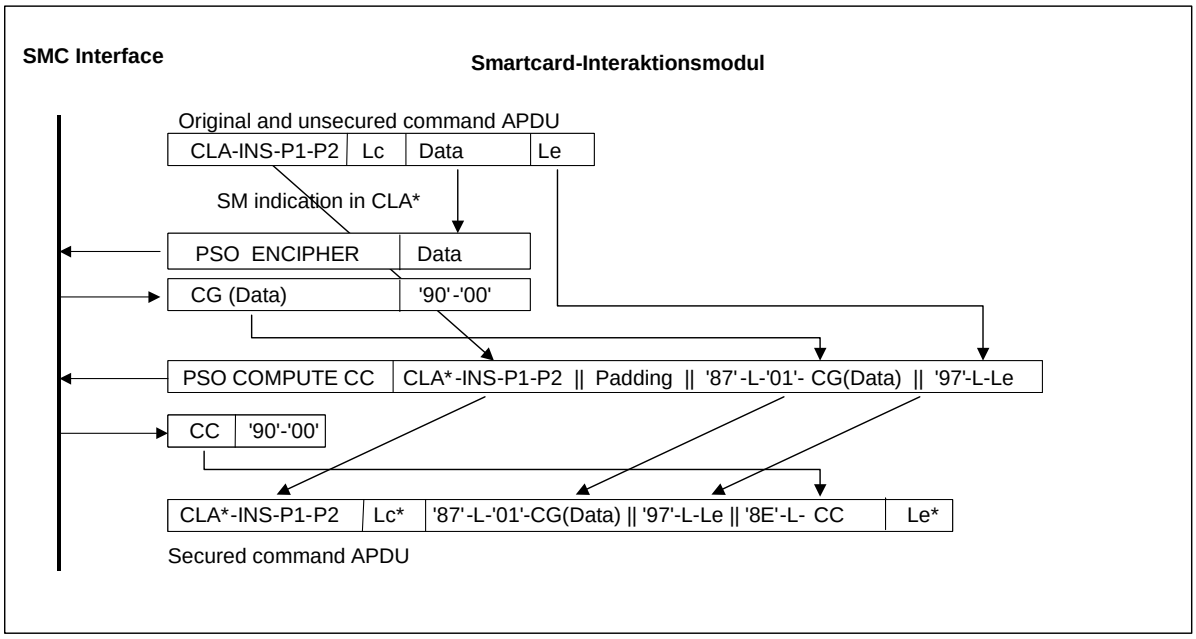


Abbildung 10 Beispiel zur Erzeugung eines SM-Kommandos

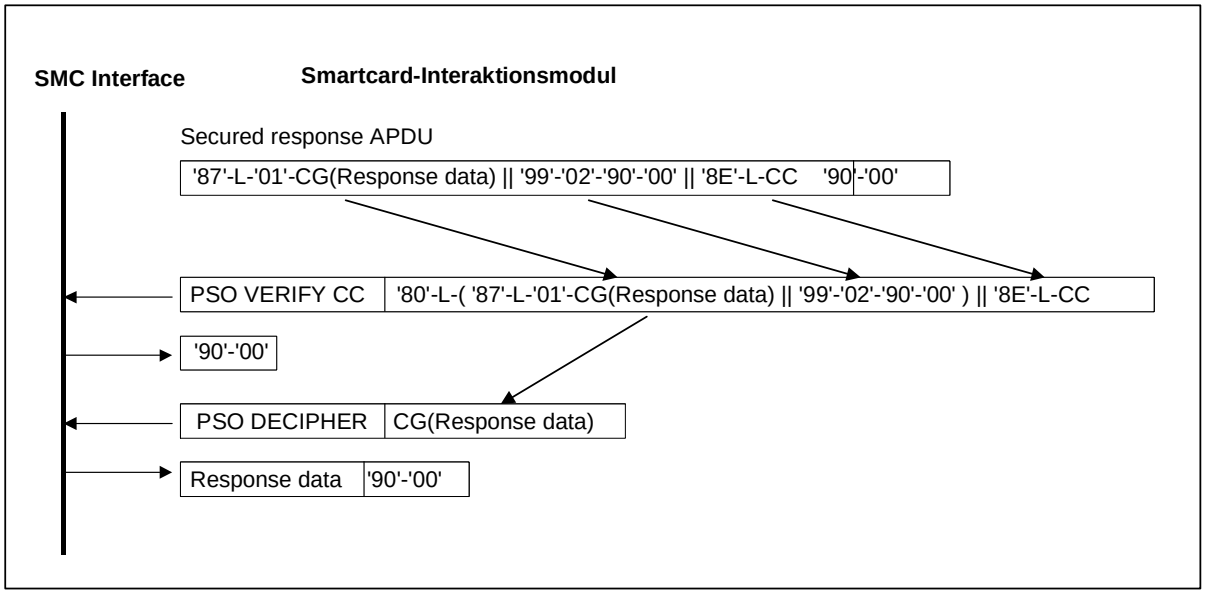


Abbildung 11 Beispiel zur Behandlung einer SM-Response

Diese Schlüssel und der Sequenzzähler sind damit am Ende der Authentisierungsprozedur auf der SMC und der Benutzerkarte verfügbar und dienen der Bildung und Prüfung von SM-Kommandos an die Benutzerkarte. Als kryptografische Prüfsumme dient ein Retail-MAC, der mit dem Sequenzzähler initialisiert und auf der Basis von DES-3 mit den Sitzungsschlüsseln gebildet wird. Auf Seiten der Benutzerkarte erfordert das SM für Kommandos und Antworten immer die Berechnung der kryptografischen Prüfsumme.

3.7 Authentisierungsschlüssel-Management

Da das Dienstleistungssystem ein zur Benutzerkarte offenes System ist, wird ein geeignetes Schlüsselmanagement gebraucht, damit Benutzerkarten unterschiedlicher Hersteller Verwendung finden können. Damit das einstufige Zertifikats-Prüfverfahren angewendet werden kann, besitzen die SMC und die Benutzerkarten Zertifikate, die von derselben Zertifizierungsstelle (CA) ausgestellt wurden und deren Schlüssel mit dem privaten Schlüssel der CA signiert wurden. Der zugehörige öffentliche RSA-Schlüssel der CA ist sowohl auf der SMC als auch auf der Benutzerkarte vorhanden und wird zur Prüfung eines Zertifikats verwendet. Konnte ein Zertifikat erfolgreich geprüft werden, so wird der im Zertifikat enthaltene öffentliche Schlüssel des Zertifikatsinhabers mit seiner Referenz temporär auf der Karte gespeichert.

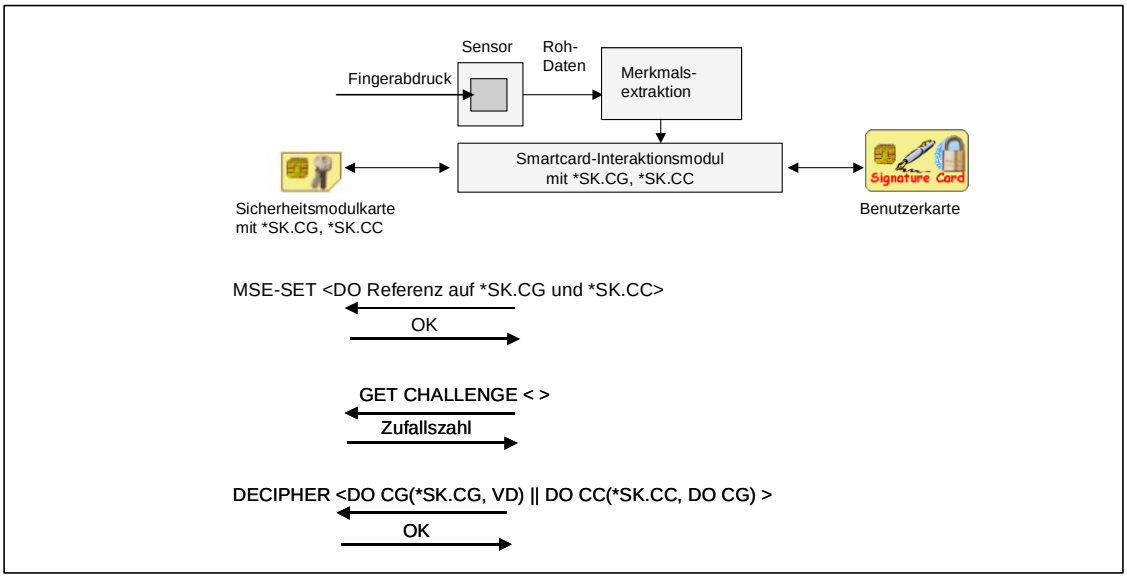
3.8 Übergabe der Verifikationsdaten an die Sicherheitsmodulkarte

Zur Berechnung der kryptografischen Prüfsumme müssen die biometrischen Verifikationsdaten auf sichere Weise an die SMC gesendet werden. Die Verifikationsdaten sollen innerhalb des Terminals im biometrischen Modul und dem Smartcard-Interaktionsmodul gegen Ausforschung und Abhörung gesichert sein. Zu diesem Zweck ist das Terminal in einem manipulationsgeschützten Gehäuse eingebettet. Vom Terminal wird außerdem erwartet, dass die biometrischen Verifikationsdaten bei der Übergabe vom Smartcard-Interaktionsmodul an die SMC authentisch sind und nachweisbar aus einer Live-Präsentation stammen. Die aktuellen Verifikationsdaten werden deshalb mit einer Prüfsumme an die SMC übertragen. Zur Absicherung der Live-Präsentation gegen Replay-Angriffe geht eine Zufallszahl in die Berechnung der Prüfsumme ein.

Zur Erzeugung der Prüfsumme wird im biometrischen Modul des Dienstleistungssystems ein symmetrisches Verschlüsselungsverfahren wie DES-3 verwendet. Da es sich bei der Schnittstelle zwischen Dienstleistungssystem und SMC um eine interne Schnittstelle in einem geschlossenen System handelt, genügt hier der einfache Einsatz symmetrischer Kryptografie. Die zugehörigen statischen Schlüssel *SK.CG und *SK.CC sind vom Hersteller in den manipuliergeschützten Umgebungen des Terminals und des Sicherheitsmoduls abgelegt. Die Befehle an die SMC werden vollständig vom Terminal kontrolliert, d. h. das Terminal berechnet mit den statischen Schlüsseln die entsprechenden SM-Datenobjekte und sendet sie an die SMC.

Die Abbildung 12 zeigt die Übergabe der biometrischen Verifikationsdaten als Kryptogramm mit kryptografischer Prüfsumme an die SMC. Die dargestellten Befehle werden nicht im SM-Modus ausgeführt, weil die zugehörigen Sitzungsschlüssel SK.CG und SK.CC nur der SMC und der Benutzerkarte, nicht aber dem Terminal bekannt sind. Mit dem MSE-SET-Kommando werden auf der SMC die statischen Schlüssel *SK.CG und *SK.CC, die dem Terminal und der SMC gemeinsam sind, als aktuelle Komponenten der SMC-Sicherheitsumgebung gesetzt. Im zweiten Kommando wird von der SMC eine Zufallszahl angefordert, welche als

Initialisierungsvektor in die Berechnung der Prüfsumme eingeht. Mit einem DECIPHER-Kommando gelangen die verschlüsselten und mit einer Prüfsumme versehenen Verifikationsdaten zum Sicherheitsmodul. Die statischen Schlüssel dürfen auf der SMC nur zum Umschlüsseln verwendet werden, d.h. das DECIPHER-Kommando darf keinen ungeschützten Klartext zurückgeben. Diese Erweiterung der Befehlsfunktion ist nun auch in der Revision von [ISO7816-8] vorgesehen worden. Wenn die Prüfsumme mit der Zufallszahl verifiziert werden konnte, werden die Verifikationsdaten in der SMC entschlüsselt und als Klartext temporär gespeichert.



*SK.CG, *SK.CC = statische symmetrische Schlüssel, VD = biometrische Verifikationsdaten, DO = Datenobjekt, CG = Cryptogram, CC = kryptografische Prüfsumme

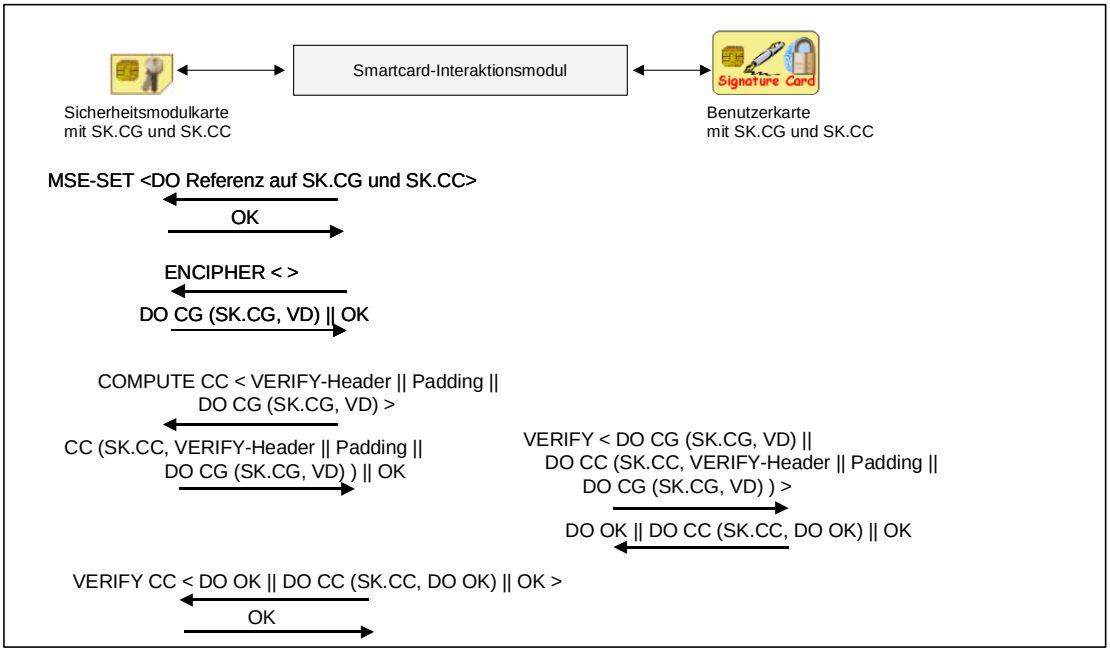
Abbildung 12 Übergabe der biometrischen Verifikationsdaten an die Sicherheitsmodulkarte

Weil von der Verschlüsselung öffentlicher Daten keine nennenswerte Erhöhung der Sicherheit erwartet wird, könnten die biometrischen Daten theoretisch auch als Klartext mit der entsprechenden Prüfsumme übertragen werden. Allerdings bietet der DECIPHER-Befehl mit Kryptogramm und Prüfsumme die Erweiterungsmöglichkeit zur Übertragung von echten Geheimnissen wie PIN oder Resetting-Code, bei denen nicht die Authentizität und Integrität, sondern die Geheimhaltung im Vordergrund stünde.

3.9 Ablauf einer biometrischen Benutzerauthentisierung

Abbildung 13 zeigt eine Befehlssequenz, die das Ziel hat, die biometrischen Verifikationsdaten eines Benutzers sicher an die Benutzerkarte zu senden und die Antwort der Verifikation sicher an das Terminal zu übermitteln. Die PSO-Kommandos ENCIPHER und COMPUTE CC an die SMC werden dazu verwendet, auf der SMC mit den Session-Keys die SM-Datenobjekten Kryptogramm und Prüfsumme der biometrischen Daten zu berechnen. Das ENCIPHER-Kommando hat im Gegensatz zum üblichen Gebrauch ein leeres Datenfeld, da sich die Daten bereits in der SMC befinden. Diese Erweiterung der Befehlsfunktion hat inzwischen auch in den revidierten Standard ISO/IEC 7816-8 Eingang gefunden.

Keiner der Befehle wird im SM-Modus an die SMC gesendet. Das Terminal konstruiert mit den erhaltenen SM-Objekten einen VERIFY-Befehl und sendet ihn an die Benutzerkarte. In der Benutzerkarte wird anhand des Session Keys und des Send Sequence Counters die Prüfsumme des Befehls geprüft. Damit ist gewährleistet, dass die Benutzerkarte die aktuelle Herkunft der biometrischen Daten vom Sensor selbst überprüft. Bei einer Benutzerauthentisierung muss der Fingerabdruck also direkt dem Sensor präsentiert worden sein und kann nicht aus einer Datenakquisitions-Attacke oder aus früheren Authentisierungsbefehlen stammen. Ist die Prüfsumme korrekt, werden die Verifikationsdaten entschlüsselt und der eigentliche Vergleich der biometrischen Verifikationsdaten mit den Referenzdaten kann stattfinden. Ein positives Verifikationsergebnis führt auf der Benutzerkarte zum Sicherheitsstatus „Benutzerauthentisierung erfolgreich“, womit die geschützte Funktion (d. h. die Signaturfunktion) zur Nutzung freigegeben ist. Die geschützte Antwort wird im Datenfeld eines VERIFY CC-Befehls an die SMC weitergereicht, wo die Prüfsumme verifiziert wird. Das Ergebnis könnte danach vom Terminal auch an andere sicherheitsrelevante Anwendungen weitergegeben werden.



SK.CG und SK.CC = symmetrische Sitzungsschlüssel, VD = biometrische Verifikationsdaten,
DO = Datenobjekt, CG = Kryptogramm, CC = kryptografische Prüfsumme, SW = Statuswort

Abbildung 13 Kryptografisch geschützte biometrische Benutzer-Authentisierung

Alle weiteren Befehle an die Signaturkarte, insbesondere der COMPUTE DS-Befehl, werden ebenfalls im SM-Modus an die Signaturkarte gesendet, wobei die Verwendung eines Kryptogramms optional, die Verwendung der Prüfsumme in jedem Befehl jedoch obligatorisch ist. Die SM-Datenobjekte werden in gleicher Weise in der SMC berechnet. Jeder Berechnung einer Prüfsumme, ungeachtet ob sie in der SMC oder in der Signaturkarte erfolgt, folgt eine prüfende Berechnung in der jeweils anderen Karte, so dass der Send Sequence Counter auf beiden Seiten gleichermaßen inkrementiert denselben Wert hat.

4 Mitteilung des biometrischen Benutzerauthentisierungsmodus

4.1 Problem

Der TST-Prototyp lässt sowohl die biometrische Benutzerauthentisierung mit Fingerabdruck als auch die wissensbasierte Benutzerauthentisierung mit einer PIN zu. Unter der Voraussetzung, dass ihre Mechanismenstärke ausreichend hoch ist (siehe die Anforderungen in [SigV01]), können biometrische Benutzerauthentisierungsverfahren die Zurechenbarkeit elektronischer Signaturen zu Personen erhöhen, da biometrische Merkmale an eine bestimmte Person gebunden sind. Dem Empfänger eines signierten Dokuments sollte auf gesicherte Weise der Benutzerauthentisierungsmodus (biometrisch oder wissensbasiert) mitgeteilt werden. Wenn ein biometrisches Benutzerauthentisierungsverfahren verwendet wurde, sollte der Signierer dies nicht ableugnen können, und wenn kein biometrisches Verfahren verwendet wurde, sollte der Signierer dies auch nicht vortäuschen können. Wenn dem Empfänger eines signierten Dokuments glaubhaft mitgeteilt wird, dass beim Signieren ein biometrisches Verfahren zur Benutzerauthentisierung verwendet wurde, und die Mechanismenstärke des Verfahrens ausreichend hoch ist, kann der Empfänger sicher sein, dass die elektronische Signatur tatsächlich vom rechtmäßigen Inhaber der Signaturkarte erzeugt wurde.

Im Rahmen des ZAVIR-Projekts wurde eine Lösung entwickelt, die diese Anforderungen erfüllt und darüber hinaus mit dem deutschen Signaturgesetz [SigG01] und der Europäischen Richtlinie für elektronische Signaturen sowie den Standard-Signaturformaten vereinbar ist und die Verifizierung der erzeugten elektronischen Signaturen nicht behindert.

4.2 Lösungsansatz

Eine STARCOS-Karte von Giesecke & Devrient mit On-Card-Matching für Fingerabdrücke [G&D03a] dient als Signaturkarte und eine Javacard von Giesecke & Devrient ist als SMC in das TST integriert. Die Antwort der Signaturkarte auf einen COMPUTE DS-Befehl nach einer biometrischen Benutzerauthentisierung ist ein Signaturblock, der neben der Dokument-Signatur die Zusatzinformation über die verwendete Benutzerauthentisierungsmethode enthält. Um diesen von der Signaturkarte gelieferten Signaturblock authentisch an den Empfänger des signierten Dokuments weiterzugeben, wird der Signaturblock mit Zusatzinformation auf der SMC mit einer Zusatz-Signatur versehen. Diese Zusatz-Signatur wird mit dem Karten-Authentisierungsschlüssel PrK.SMC.AUT erzeugt, auf dessen Nutzung der Benutzer keinen Einfluss hat. Die Lösung ist fälschungssicher, weil das Signieren der Zusatzinformation ausschließlich von der SMC kontrolliert wird, also unter Verwendung eines von außen vorgegebenen Wertes nicht möglich ist. Die Verwendung von PrK.SMC.AUT, die

bisher auf den Befehl INTERNAL AUTHENTICATE beschränkt war, ist damit auf die Signatur der Zusatzinformation ausgeweitet worden. Dies muss durch die entsprechende Zertifizierungsinstanz bestätigt werden. Die SMC signiert den Signaturblock mit Zusatzinformation beim Ausführen des VERIFY CC nach einem entsprechenden mit Biometrie freigeschalteten COMPUTE DS der Signaturkarte:

(1) Verifikation der Prüfsumme im Datenfeld des COMPUTE CC
(2) Wenn ein Signaturblock mit Zusatzinformation im Datenfeld des COMPUTE CC vorhanden ist:

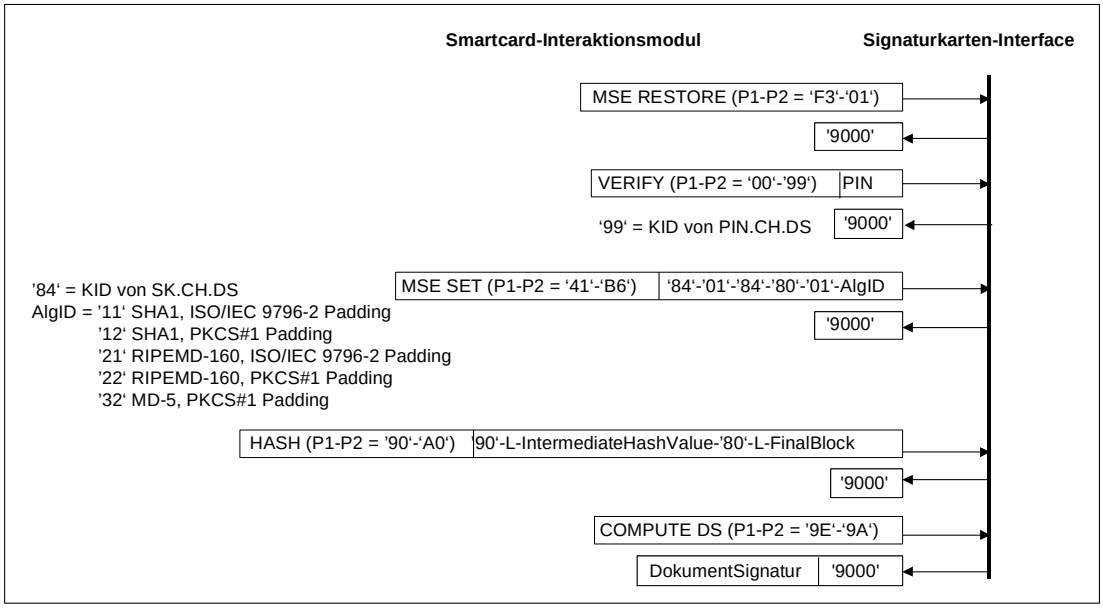
    a) Berechnung der Zusatz-Signatur über diesen Signaturblock unter Verwendung des Karten-Authentisierungsschlüssels PrK.SMC.AUT
    b) Speicherung des Signaturblocks mit Zusatz-Signatur in einer lesbaren Log-Datei

Die gespeicherte Logging-Information wird anschließend ausgelesen und dem Empfänger zusammen mit dem signierten Dokument und den X.509-Zertifikaten beider Schlüssel mitgeteilt. Durch Prüfen der Dokument-Signatur und der Zusatzsignatur und durch Prüfen, ob die an das Dokument angehängte Dokument-Signatur mit der Dokument-Signatur in der Zusatzinformation identisch ist, kann auf Empfängerseite sowohl die Echtheit des Dokuments als auch die bei der Signaturerstellung verwendete biometrische Benutzerauthentisierung nachgewiesen werden.

4.3 Realisierung der Signaturphasen auf der Signaturkarte

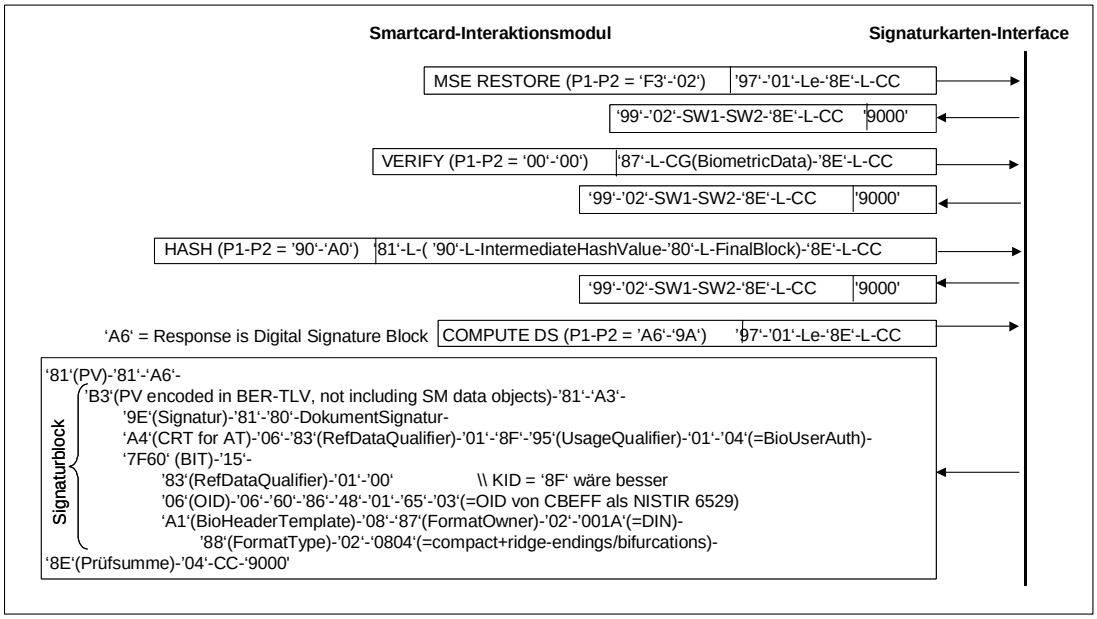
Jede Benutzer-Authentisierungsmethode läuft auf der STARCOS SOK 2.4-Signaturkarte in ihrer eigenen Sicherheitsumgebung ab, d. h. die PIN-Authentisierung im Security Environment SE#1 (siehe Abbildung 14), die biometrische Authentisierung im SE#2 (siehe Abbildung 15). Mit MSE RESTORE kann die aktuelle Sicherheitsumgebung gewechselt werden. In Abhängigkeit von der Sicherheitsumgebung sind bestimmte Kommandos nur in der geeigneten Ausprägung erlaubt (speziell das VERIFY und das COMPUTE DS). Beispielsweise ist es nicht möglich, in der Sicherheitsumgebung der biometrischen Benutzerauthentisierung ein wissensbasiertes VERIFY auszuführen. Dadurch kann nicht zunächst eine Authentisierung mit der PIN erfolgen und unmittelbar danach eine Signatur erzeugt werden, welche sinngemäß die Zusatzinformation „Benutzer-Authentisierung erfolgte mit Biometrie“ trägt. Sobald eine andere Sicherheitsumgebung gesetzt wird, wird der lokale (applikationsgebundene) Sicherheitszustand (z. B. „Benutzer-Authentisierung erfolgreich“) gelöscht. SE#2 lässt keinen MSE SET-Befehl zu, sondern erhält die nötigen Informationen aus dem ersten (und einzigen) Record von EF_SE (siehe Abbildung 16).

Vor der biometrischen Authentisierung sind Informationen über die biometrischen Daten (Biometric, Information Template, BIT) auslesbar. Ein GET DATA mit P1-P2 = '7F60' (Tag des BIT) ist nicht möglich. Stattdessen kann EF_SE selektiert und mit einem READ RECORD ausgelesen werden (nach [ISO7816-11]). Die Referenzdaten werden anhand des zweiten Parameters des VERIFY-Befehls identifiziert. VERIFY funktioniert in SE#1 mit P2 = '00' (no information given) und P2 = '99' (KID von PIN.CH.DS), in SE#2 dagegen nur



(Befehle nicht im SM-Modus)

Abbildung 14 Signaturphase im SE#1 (PIN-Modus)



(alle Befehle im SM-Modus)

Abbildung 15 Signaturphase im SE#2 (Biometrie-Modus)

mit P2 = '00'. Der Retry-Counter für die Verifikation der PIN.CH.DS hat den Initialwert 3, der Retry-Counter für die Verifikation der biometrischen Daten den Initialwert 14.

```
7B 2E                          -- Security Environment Template
   80 01 02                        -- Security Environment ID (= SE#2)
   A4 06                           -- CRT for Authentication
      83 01 8F                        -- Reference Data Qualifier (biometric reference)
      95 01 04                        -- Usage Qualifier (= Biometric User Authentication)
   B6 09                           -- CRT for Digital Signature
      84 01 84                        -- Private Key Reference (= SK.CH.DS)
      80 01 12                        -- Algorithm Reference (= SHA1, PKCS#1 Padding)
      95 01 40                        -- Usage Qualifier (= Computation)
   7F 60 15                        -- Biometric Information Template (BIT)
      83 01 00                        -- Reference Data Qualifier for VERIFY
      06 06 60 86 48 01 65 03         -- OID (NISTIR 6529 in Comp. Sec. Objects Reg.)
      A1 08                           -- Biometric Information Data Objects
         87 02 00 1A                     -- Format Owner (= DIN)
            88 02 08 04                     -- compact, ridge endings, ridge bifurcations)
```

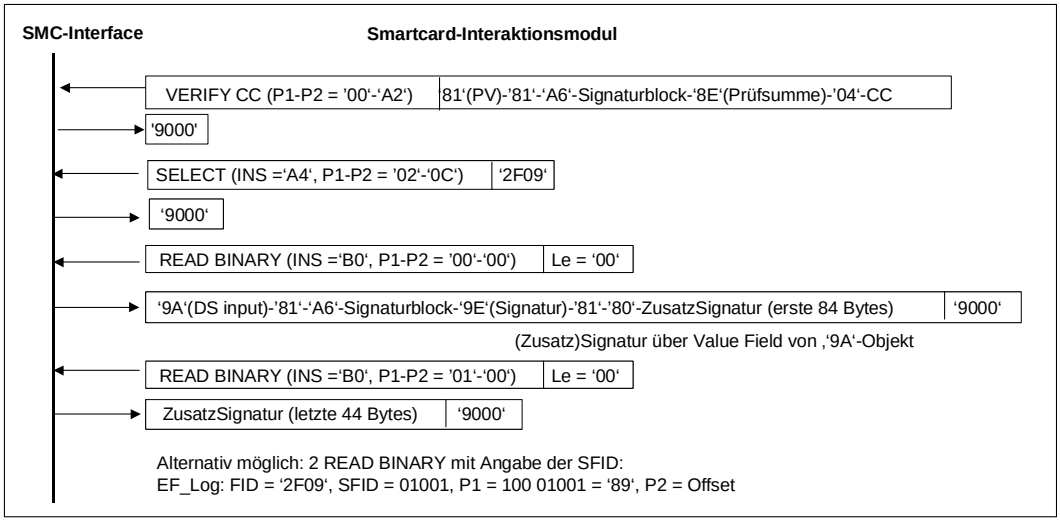
Abbildung 16 Daten der Datei EF_SE auf der Signaturkarte

4.4 Realisierung der Zusatzsignatur auf der Sicherheitsmodulkarte

Die als SMC dienende Sm@rtcafé 2.0 Javacard verwendet zur Bildung SM-geschützter Befehle an die Signaturkarte die PSO-Kommandos DECIPHER, ENCIPHER und COMPUTE CC. Die SM-geschützten Antworten der Signaturkarte werden mit VERIFY CC, DECIPHER und ENCIPHER analysiert. Die Antwort des COMPUTE DS wird wie jede mit SM-geschützte Signaturkarten-Antwort im Datenfeld eines VERIFY CC zur SMC gesendet und geprüft. Handelt es sich dabei aufgrund des Biometrie-Modus der Signaturkarte um den Signaturblock mit Zusatzinformationen (erkennbar am Tag 'B3' und der Länge) führt COMPUTE CC die folgende zusätzlich implementierte Funktion aus: Die SMC signiert den Signaturblock mit dem Karten-Authentisierungsschlüssel PrK.SMC.AUT. Der Signatur-Algorithmus ist RSA/SHA mit PKCS#1-Padding. Die Zusatzinformation wird mit der Zusatz-Signatur konkateniert und in die Log-Datei EF_LOG (FID = '2F09') eingetragen. Die Datei kann anschließend (bei gleichem Sicherheitsstatus „USERCARD_AUTH_SUCCESSFUL“) über ihren FID selektiert und mit zwei READ BINARY (256 + 44 Antwort-Bytes) gelesen werden (siehe Abbildung 17). Alternativ können auch direkt zwei READ BINARY mit Angabe der SFID ('9' = 01001, P1 = 1000 1001 = '89') gesendet werden, wobei allerdings Offsets nur zwischen 0 und 255 möglich sind. Die Datei EF_C_X509_SMC_AUT (FID = '2F0A') enthält ein X.509-Zertifikat mit dem öffentlichen Schlüssel PuK.SMC.AUT und ist immer lesbar. Bei einer Zertifikatslänge von beispielsweise 835 Bytes sind 4 READ BINARY (256 + 256 + 256 + 67 Antwort-Bytes) nötig. Alle Befehle an die SMC werden vom Secure-Messaging-Modul des TST gesendet. Dieses Modul stellt an der internen Schnittstelle zum TST zusätzlich zu den SM-Funktionen eine Funktion zum Lesen der Log-Datei bereit (siehe Abbildung 17).

4.5 Senden des signierten Dokuments an den Empfänger

Das signierte Dokument wird vom TST zu einer PKCS#7-Message vom Typ „SignedData“ verarbeitet, um es anschließend an den Empfänger zu versenden. „SignedData“ setzt sich



(Befehle nicht im SM-Modus)

Abbildung 17 Senden des Signaturblocks an die Sicherheitsmodulkarte und Lesen der Log-Datei

u. a. aus dem Dokument („contentInfo"), dem X.509-Zertifikat mit dem öffentlichen Signaturschlüssel PuK.CH.DS und „signerInfo" zusammen. Als „signerInfo" fügt das TST den Hashwert des Dokuments, sowie zusätzliche Attribute wie z.B. die Signaturzeit als „authenticatedAttributes" zusammen. Die eigentliche Signatur wird aus dem Hashwert (dem Input für COMPUTE DS) dieses Attributblockes gebildet, wobei die letzte „Runde" der Hashbildung in der Smartcard erfolgt. Die Dokument-Signatur wird als „encryptedDigest" ebenfalls in die „signerInfo" übernommen. Zusätzlich zu diesen notwendigen Angaben können noch beliebige „unauthenticatedAttributes" angegeben werden. Diese Angaben müssen rein informativ sein und bedürfen daher keiner (elektronischen) Unterschrift. In diesen Block wird u. a. der Nachweis eingefügt, auf welche Art der Signierende sich gegenüber der Karte authentisiert hat.

Befindet sich die Signaturkarte im Biometrie-Modus, sendet sie als Antwort des COMPUTE DS einen Signaturblock mit der Dokument-Signatur und den Zusatzinformationen der biometrischen Authentisierung. Da dies über den Trusted Channel und somit über die SMC erfolgt, hat diese die Möglichkeit, die Zusatzinformation über die biometrische Authentisierung des Nutzers elektronisch zu signieren und somit eine gewisse Nachweissicherheit zu bieten; in diesem Signaturblock ist nicht nur die Information über die Art der Authentisierung enthalten, sondern auch die Dokumentsignatur, um diese Zusatzinformation an den jeweiligen Signaturvorgang zu binden. Während zunächst nur die eigentliche Dokumentsignatur an die Signaturanwendung zurück gesendet wird, liest die Signaturbibliothek noch den Signaturblock mit der Zusatz-Signatur von der SMC aus. Die Bibliothek fügt die Dokument-Signatur in „contentInfo", dem Signaturblock, das X.509-Zertifikat der SMC mit dem Publikkey PuK.SMC.AUT und die erzeugte Signatur in eine PKCS#7-Message (ebenfalls vom Typ „Signed Data") ein. Diese signierte, eigenständige PKCS#7-Block wird als Objekt in den „unauthenticatedAttributes" der Dokument-PKCS#7-Nachricht eingefügt.

Das TST überträgt nach dem Signaturvorgang das signierte Dokument an den PC des Signierers (siehe Abbildung 18). Dieser kann nun mit Hilfe seines Mailprogramms das Dokument über das Internet versenden. Das Dokument kann von jedem Programm geprüft werden, das PKCS#7-Messages versteht (z. B. das E-Mail-Programm Outlook) und die notwendigen CA-Zertifikate kennt. Für die „unauthenticatedAttributes“ ist allerdings ein Zusatzprogramm nötig. Das Mailprogramm prüft auf Empfängerseite lediglich das erhaltene Dokument mit seiner Signatur:

(1) Prüfen des X.509-Zertifikats von PuK.CH.DS (Root/CA-Zertifikat des Zertifikatsaustellers des Signierers erforderlich)
(2) Bilden des Dokument-Hashwerts und Verifizieren der Dokument-Signatur mit PuK.CH.DS
(3) Anzeigen des Verifikationsergebnisses

Das Zusatzprogramm, dass die Zusatzinformation der Benutzerauthentisierung anzeigt, hat folgende Funktionalität:

(1) Prüfen des X.509-Zertifikats von PuK.SMC.AUT (Root/CA-Zertifikat des SMC-Ausgebers nötig)
(2) Prüfen der Zusatz-Signatur mit PuK.SMC.AUT
(3) Prüfen der Dokument-Signatur im „encryptedDigest“ und im Signaturblock auf Gleichheit
(4) Anzeigen des verwendeten biometrischen Authentisierungsmodus

Wurde dem Empfänger des signierten Dokuments keine Zusatzinformation mitgesendet, so

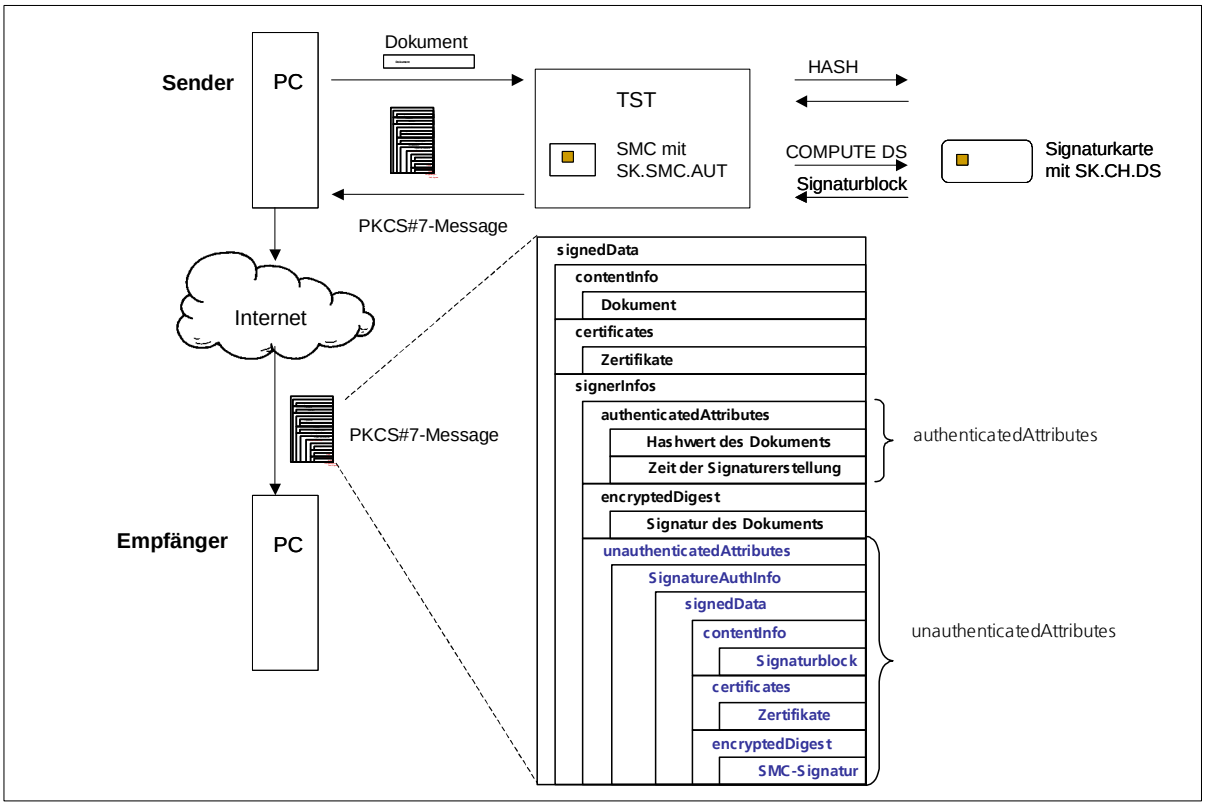


Abbildung 18 Mitteilung des Authentisierungsmodus an den Empfänger

gilt als Default-Einstellung, dass die Signaturfunktion durch einen wissensbasierten Authentisierungsmodus freigegeben wurde.

4.6 Anzeige/Verifikation des Benutzerauthentisierungsmodus auf der Empfängerseite

Da es sich bei dem signierten Dokument um eine übliche PKCS#7-Nachricht [RSA93] handelt, ist die Verifizierung der elektronischern Signatur mit heute üblicher Software möglich. Diese Programme bieten jedoch keine oder eine sehr eingeschränkte Ansicht der (Zusatz-) Attribute, insbesondere können sie die spezielle Datenstruktur zur Mitteilung des Benutzerauthentisierungsmodus nicht anzeigen. Daher ist eine gesonderte Software nötig, die in Form einer Explorer-Erweiterung implementiert wurde. Durch das Aktivieren des Kontext-Menüs (Rechtsklick) einer vom TST erzeugten signierten E-Mail-Datei kann durch den neuen Menüpunkt „Signer's authentication info...“ eine entsprechende Information abgerufen werden, siehe Abbildung 19.

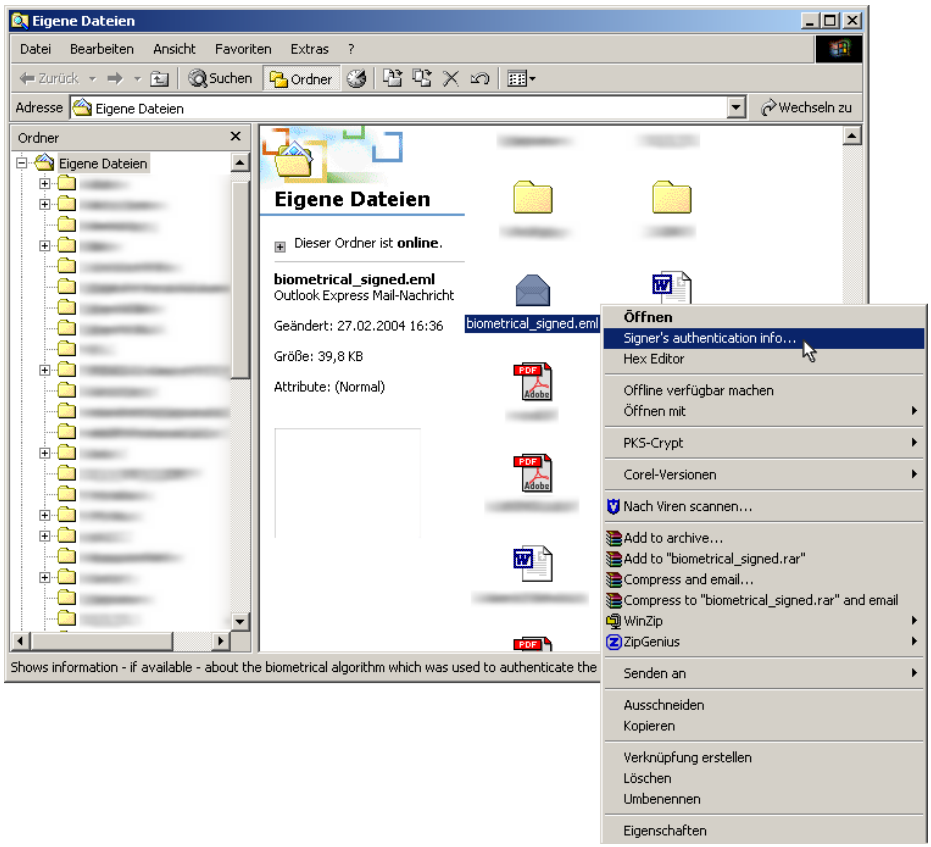


Abbildung 19 Explorer-Erweiterung

Das Plugin durchsucht die in der E-Mail enthaltene PKCS#7-Struktur nach den in den „unauthenticatedAttributes“ verzeichneten Authentisierungsinformationen. Werden sie gefunden, so müssen die Signaturen des eigentlichen Dokuments mit der im Attribut gesicherten Version auf Übereinstimmung gestestet werden (diese Information stellt die Verankerung des Attributs mit dem Dokument dar), danach die Signatur des Attributes selbst. Ist diese korrekt, so kann die Information im „Signaturdatensatz“ entsprechend ausgewertet werden.

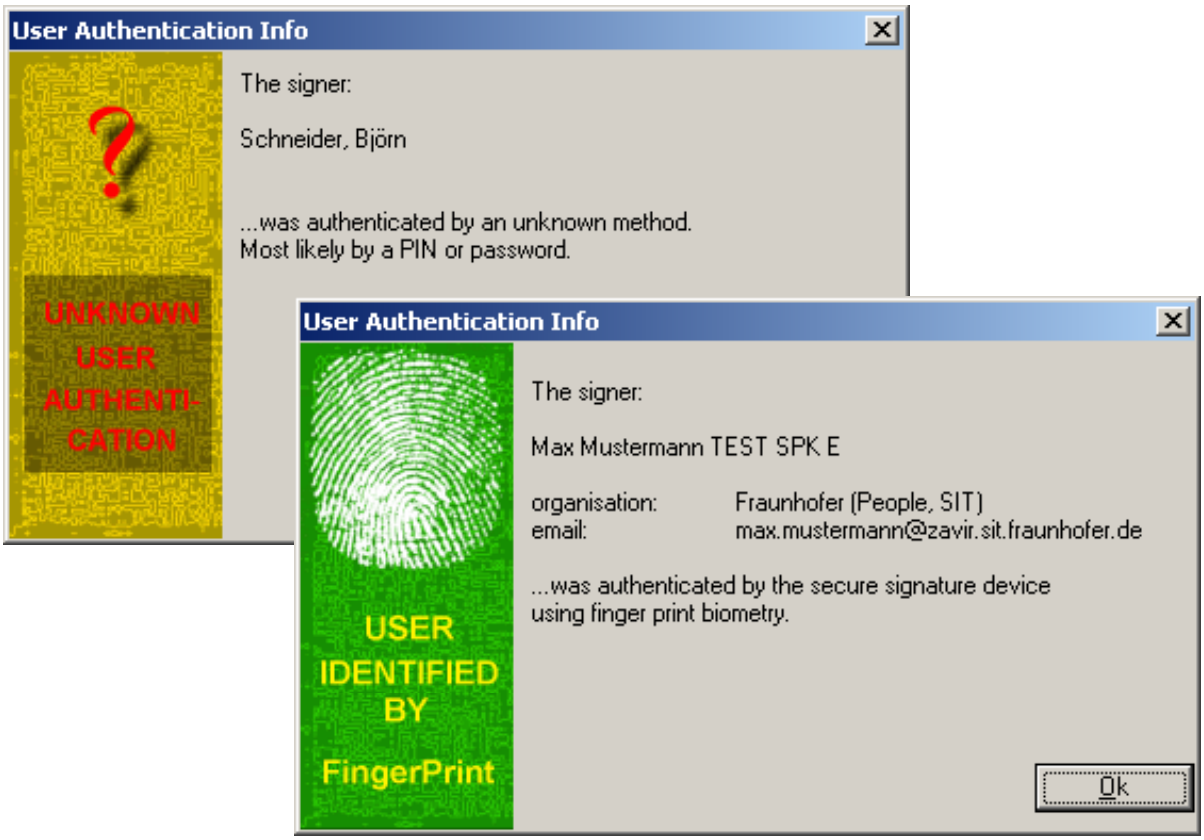


Abbildung 20 Mögliche Informationsfenster mit User Authentication Info

Diese Vorgehensweise ergibt derzeit zwei mögliche Informationsfenster (bei einer PIN-Authentisierung wird kein Informationsdatensatz den „unauthenticatedAttributes“ hinzugefügt), siehe Abbildung 20. Die Informationsfenster zeigen zusätzlich noch die Standard-Informationen des Zertifikats des Unterzeichners an, um eine genaue Zuordnung durch den Betrachter zu ermöglichen.

5 Biometrische Benutzerauthentisierung auf Smartcards mittels handschriftlicher Unterschriften

5.1 Einführung

Eine wesentliche Grundlage für die Sicherheit von Smartcards ist die verlässliche Authentisierung der Benutzer. Die Identität einer Person, die im Besitz einer Smartcard ist und diese benutzen will, kann an Hand der Kenntnis einer geheimen PIN (Personal Identification Number) oder an Hand von biometrischen Merkmalen (physiologische Merkmale, z. B. Fingerabdruck, oder Verhaltensmerkmale, z. B. handgeschriebene Unterschrift) überprüft werden. Da der Karteninhaber bei biometrischer Benutzerauthentisierung keine PIN einzugeben braucht, gelten biometrische Verfahren als benutzerfreundlicher als die bisher üblichen wissensbasierten Verfahren. Bei wissensbasierter Benutzerauthentisierung besteht darüber hinaus die Gefahr, dass die PIN in die Hände unberechtigter Personen gelangt und missbraucht wird. Da biometrische Merkmale an eine bestimmte Person gebunden sind, können biometrische Benutzerauthentisierungsverfahren auch eine höhere Sicherheit als wissensbasierte Verfahren bieten, vorausgesetzt die Überwindungssicherheit und Erkennungsleistung der biometrischen Verfahren sind ausreichend hoch.

Um festzustellen, ob ein zu authentisierender Benutzer der rechtmäßige Karteninhaber ist, erfolgt bei der biometrischen Benutzerauthentisierung ein Eins-zu-Eins-Vergleich seiner biometrischen Merkmale mit den biometrischen Merkmalen des rechtmäßigen Karteninhabers, die zuvor beim „Enrollment“ als Referenzdaten in der Smartcard abgespeichert wurden.

Aus den zeitlichen Verläufen der Stiftposition und gegebenenfalls des Schreibdrucks und der Stiftneigung, die beim handschriftlichen Unterschreiben auf einem grafischen Tablett (Online-Unterschrift) aufgenommen werden, können nach einer Vorverarbeitung biometrische Merkmale (Unterschriftsdynamik) extrahiert werden, die zur Benutzerauthentisierung verwendet werden können. Der Vorteil der Unterschriftsdynamik gegenüber anderen biometrischen Merkmalen ist, dass handschriftliche Unterschriften als Mittel zur Authentisierung von Personen vielerorts seit langem akzeptiert sind. Darüber hinaus werden handschriftliche Unterschriften als Ausdruck einer willentlichen Entscheidung des Schreibers angesehen, da sie i. allg. nicht zufällig oder unbeabsichtigt abgegeben werden.

Der Vorgang des Unterschreibens selbst läuft i. allg. in hohem Maße automatisch ab und bedarf keiner oder nur einer geringen bewußten Kontrolle durch den Schreiber. Die Unterschriftsdynamik einer Person ist jedoch nicht unveränderlich, sondern variiert von Unterschrift zu Unterschrift innerhalb eines gewissen Toleranzbereichs. Damit ein berechtigter Be-

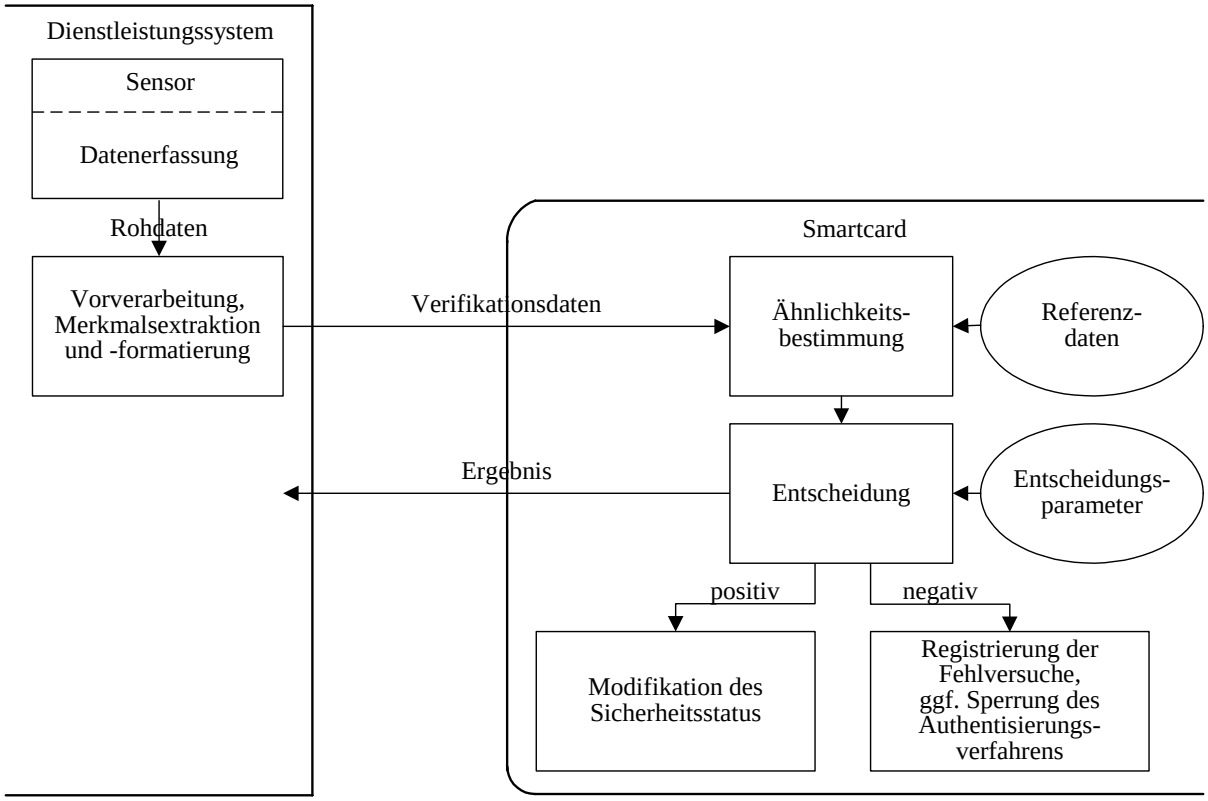


Abbildung 21 On-Card-Matching zur biometrischen Benutzerauthentisierung

nutzer nicht zu oft fälschlicherweise abgewiesen wird, muss bei der Benutzerauthentisierung ebenfalls ein gewisser Toleranzbereich zugelassen werden. Der Toleranzbereich darf jedoch nicht zu groß sein, damit die Chancen eines Angreifers, der sich als berechtigter Benutzer ausgibt, angenommen zu werden, möglichst klein sind.

Wenn eine Smartcard sicherheitsrelevante Funktionen bereitstellt oder Werte oder sicherheitssensitive Daten trägt, dann sollte der Vergleich von Verifikations- und Referenzdaten in der Smartcard selbst erfolgen (On-Card-Matching). Dies verhindert, dass der Karte ein positives Ergebnis nur vorgetäuscht wird. Die durch das On-Card-Matching geschützte Funktion auf der Smartcard kann erst nach erfolgreicher Benutzerauthentisierung verwendet werden [Str01]. Abbildung 21 illustriert die Arbeitsteilung zwischen Dienstleistungssystem und Smartcard.

Es gibt eine Vielzahl von Methoden zur Analyse von On-line-Unterschriften. Ein Überblick ist in [Pla89, Sch98] zu finden. Da Smartcards nur eine begrenzte Speicherkapazität besitzen und ihre Rechenleistung um Potenzen geringer ist als die üblicher PCs, müssen Algorithmen für das On-Card-Matching sorgfältig ausgewählt und an diese Bedingungen angepasst werden. Es wurde daher ein geeigneter Ansatz ausgewählt und im Sinne einer Machbarkeitsuntersuchung auf Java-Karten implementiert.

5.2 Anforderungen

5.2.1 Qualitätsanforderungen

Biometrische Benutzerauthentisierungsverfahren können entweder als vollwertige Alternative oder als Zusatz zu wissensbasierten Verfahren zum Einsatz kommen. Die Signaturverordnung [SigV01] erlaubt den Einsatz biometrischer Benutzerauthentisierungsverfahren auch in Produkten für qualifizierte elektronische Signaturen, mit denen wie mit papiergebundenen handschriftlichen Unterschriften rechtsverbindliche Erklärungen möglich sind. Für den Fall, dass ein biometrisches Verfahren als vollwertige Alternative zum wissensbasierten Verfahren zum Schutz von geheimen Signaturschlüsseln (6stellige PIN) eingesetzt werden soll, wird in [SigV01] gefordert, dass eine dem wissensbasierten Verfahren gleichwertige Sicherheit, d. h. die Mechanismenstärke „hoch“, zu erreichen ist. Mit der Mechanismenstärke „hoch“ bewertete Sicherheitsfunktionen müssen ausreichenden Schutz bieten vor einem geplanten, organisierten Brechen der Sicherheit durch Angreifer, die über ein hohes Angriffspotential verfügen [ISO15408]. Für den Fall, dass ein biometrisches Verfahren nur zusätzlich zur wissensbasierten Benutzerauthentisierung eingesetzt werden soll, reicht es aus, wenn die Mechanismenstärke „mittel“ erreicht wird. Mit der Mechanismenstärke „mittel“ bewertete Sicherheitsfunktionen müssen ausreichenden Schutz bieten vor einem absichtlichen Brechen der Sicherheit durch Angreifer, die über ein mittleres Angriffspotential verfügen [ISO15408]. Eine allgemeine Definition des Angriffspotentials ist in [CEM99] zu finden.

Ob biometrische Benutzerauthentisierungsverfahren diese hohen Anforderungen an die Mechanismenstärke erfüllen oder nicht, muss durch Tests ihrer Überwindungssicherheit und ihrer Falschakzeptanzrate (False Acceptance Rate, FAR) überprüft werden. Die Falschakzeptanzrate ist die Wahrscheinlichkeit, dass ein biometrisches System fälschlicherweise einen unberechtigten Benutzer zulässt. Die Falschakzeptanzrate ist ein Maß für die Sicherheit eines biometrischen Benutzerauthentisierungsverfahrens, die Falschrückweisungsrate (False Rejection Rate, FRR) hingegen ist ein Maß für die Benutzerfreundlichkeit des Verfahrens. Die Falschrückweisungsrate ist die Wahrscheinlichkeit, dass ein biometrisches System einen berechtigten Benutzer fälschlicherweise zurückweist. Die Falschakzeptanz- und Falschrückweisungsrate eines biometrischen Systems hängen i. allg. von einem einstellbaren Schwellenwert ab, der den geforderten Grad der Ähnlichkeit von Verifikations- und Referenzdaten bestimmt. Es wäre wünschenswert, den Schwellenwert so einzustellen, dass sowohl die Falschakzeptanzrate als auch die Falschrückweisungsrate möglichst gering sind, diese beiden Anforderungen laufen jedoch einander zuwider: Je geringer die Falschakzeptanzrate, d. h. je weniger Fälschungen akzeptiert werden, desto höher ist die Falschrückweisungsrate, d. h. desto weniger authentische Unterschriften werden auch akzeptiert, und umgekehrt. Bei der Wahl des Schwellenwertes müssen Sicherheit (geringe Falschakzeptanzrate) und Benutzerfreundlichkeit (geringe Falschrückweisungsrate) in Abhängigkeit von den Anforderungen an die konkrete Anwendung gegeneinander abgewogen werden.

Die Unterschriftsdynamik kann weniger leicht gefälscht werden als die geometrische Form einer Unterschrift, da Informationen über die Unterschriftsdynamik einem potentiellen Fälscher weniger leicht zugänglich sind als Informationen über das Aussehen einer Unterschrift [Nal99]. Aber auch die Unterschriftsdynamik ist sicher nicht unfälschbar. Wenn ein poten-

tieller Fälscher weiß, dass es auf die Unterschriftsdynamik ankommt, und er den Prozeß des Unterschreibens bei seinem Opfer beobachten kann, dann kann er auch mit gewissem Aufwand die Unterschriftsdynamik seines Opfers erlernen oder mit technischen Hilfsmitteln nachahmen.

Falschakzeptanz- und Falschrückweisungsrate können mit Hilfe umfangreicher Testdatenbanken experimentell ermittelt werden (mit einer gewissen statistischen Unsicherheit) [Man02, Phi00]. Ob es überhaupt möglich ist, die hohen Anforderungen an die Mechanismenstärke durch On-line-Unterschriftenanalyse zu erfüllen, d. h., ob die Unterschriftsdynamik berechtigter Benutzer sich hinreichend von der Schreibdynamik geschickter Fälscher unterscheidet, muss noch weiter untersucht werden.

5.2.2 Standardisierbarkeit

Für das On-Card-Matching sollen solche Unterschriftsmerkmale ausgewählt und solche Datenformate für die Übertragung über die Chipkartenschnittstelle entwickelt werden, die sich für eine spätere Standardisierung eignen. Die Standardisierung ist wichtig, da Smartcards unterschiedlicher Hersteller mit Merkmalsextraktionsverfahren verschiedener Hersteller zusammenarbeiten sollen.

5.2.3 Mindestanforderungen an die eingesetzten grafischen Tabletts

Graphische Tabletts liefern Abtastfolgen für die Stiftposition (x- und y-Koordinaten), in einigen Fällen auch Zusatzinformationen wie Abtastfolgen für den Schreibdruck oder für die Stiftneigung. Das On-Card-Matching-Verfahren für On-line-Unterschriften soll vom verwendeten grafischen Tablett weitgehend unabhängig sein, damit es in unterschiedlichen Umgebungen zum Einsatz kommen kann. Unterschiede zwischen den grafischen Tabletts sind, soweit möglich, durch eine geeignete Vorverarbeitung der aufgenommenen Rohdaten (siehe Abschnitt 5.3.2) auszugleichen. Die Unabhängigkeit vom eingesetzten Tablett wird durch Kompatibilität des Tabletts zur gängigen Programmierschnittstelle Wintab-API für grafische Tabletts unter Microsoft Windows [Poy94] unterstützt.

Zusatzinformationen wie Schreibdruck und Stiftneigungswinkel erleichtern den Unterschriftenvergleich. Da jedoch nicht jedes Tablett diese Zusatzinformationen liefert, werden sie in dem hier entwickelten On-Card-Matching-Verfahren nicht berücksichtigt. Selbst wenn verschiedene Tabletts Druckwerte liefern, ist deren einheitliche Kalibrierung schwierig, zumal während des Unterschreibens bisweilen der vom jeweiligen Tablett aufnehmbare maximale Druckwert erreicht wird.

Da die Unterschriftsdynamik einer Person von Unterschrift zu Unterschrift geringfügig variiert und Unterschriften innerhalb gewisser Toleranzbereiche akzeptiert werden müssen, ist es nicht notwendig, die Unterschriftsdynamik so genau wie möglich aufzunehmen. Wenn die zulässige Toleranz beispielsweise im Bereich von 0,1 mm liegt, ist eine Auflösung im Bereich von 0,001 mm nicht erforderlich. Damit die Qualitätsanforderungen an das On-Card-

Matching-Verfahren erfüllt werden können, müssen die eingesetzten grafischen Tabletts jedoch gewisse Mindestanforderungen erfüllen. Die Daten müssen möglichst reproduzierbar erfasst werden und sollen möglichst wenig von externen Parametern wie z. B. der Stiftneigung abhängen. Wie groß die Toleranzbereiche bei handschriftlichen Unterschriften sein dürfen, ist noch genauer zu untersuchen. Wir gehen davon aus, dass eine physikalische Auflösung des grafischen Tabletts von mindestens 1000 dpi (Punkte bzw. Pixel pro Zoll), d. h. höchstens 0,0254 mm Pixel-Abstand, auf jeden Fall ausreicht.

Eine Abtastrate von 100 Hz wird als ausreichend angesehen (10 ms Abstand zwischen den Abtastzeitpunkten). Damit nicht für jeden Abtastzeitpunkt eine Zeitangabe aufgezeichnet werden muss, sondern nur einmalig der für jeweils benachbarte Abtastzeitpunkte gleiche Zeitabstand, sollte das grafische Tablett äquidistante Abtastfolgen liefern.

Bei einigen Tabletts wird die Stiftposition nicht nur dann gemeldet, wenn der Stift das Tablett berührt, sondern auch wenn sich der Stift innerhalb eines „Nahbereichs“ über dem Tablett befindet. Die Erfassung der Stiftbewegungen im Nahbereich des Tabletts liefert individuelle Schreiberinformationen, die ein Fälscher kaum nachvollziehen kann, und sollte deshalb vom eingesetzten grafischen Tablett geleistet werden.

Damit das Schreiben auf einem grafischen Tablett möglichst den gewohnten Schreibbedingungen entspricht, muss das Tablett einige ergonomische Anforderungen erfüllen. Position und Schreibfläche des Tabletts sollten den Schreibvorgang nicht einschränken. Für ein besseres Schreibgefühl sollte der verwendete Stift traditionellen Schreibgeräten nachempfunden sein. Stifte mit einer Mine oder grafische Tabletts kombiniert mit LCD-Bildschirm liefern dem Unterschreibenden visuellen Feedback und erhöhen so die Sicherheit und fördern den gewohnten Automatismus während des Unterschreibens.

5.2.4 Implementierungsplattform

Java-Karten

Java-Karten sind Smartcards mit einem Interpreter (Java Card Virtual Machine) für die Ausführung von Bytecode, d. h. von prozessorunabhängigem Objektkode. Ihre Funktionalität ist in der Java-Card-2.1.1-Spezifikation [Sun00a, Sun00b, Sun00c] beschrieben. Da ihre Programmierung auf einer Untermenge von Java, einer Java Card Virtual Machine als Ausführungsplattform und Java-Entwicklungswerkzeugen beruht, eignen sich Java-Karten zur prototypischen Implementierung des On-Card-Matching-Verfahrens.

Der Speicherplatz, die Rechengeschwindigkeit und der Java-Sprachumfang auf einer Java-Karte sind beschränkt. Auf Java-Karten werden die Datentypen *boolean*, *byte*, *short* und optional *int* unterstützt, *char*, *double*, *float* und *long* jedoch nicht. Es werden nur eindimensionale Felder unterstützt. Es stehen nur Grundrechenarten und i. allg. keine mathematischen Bibliotheken zur Verfügung. Es gibt keine Garbage-Collection und kein *finalize*(). Einmal erzeugte Objekte und Felder können nicht wieder entfernt werden, ihr Speicherplatz bleibt belegt. Aus diesem Grund sind alle notwendigen Objekte und Felder bei der Installation eines Java Card

Applets zu erzeugen und später immer wieder zu verwenden. Dynamisches Laden von Klassen wird nicht unterstützt. Alle notwendigen Klassen müssen bei der Herstellung der Java-Karte oder bei der Installation eines Java Card Applets auf die Java-Karte gebracht werden.

Um die Qualitätsanforderungen an das On-Card-Matching-Verfahren erfüllen zu können, wurden für die Implementierung leistungsfähige Java-Karten mit den folgenden technischen Daten ausgewählt [G&D01], die jedoch bei weitem nicht an die Rechenleistung heutiger PCs heranreichen:

- Wortlänge der CPU: 16-Bit
- RAM: 2 kByte (als Arbeitsspeicher)
- ROM: 64 kByte (für Betriebssystem und Java Card Virtual Machine)
- EEPROM: 32 kByte (für Java Card Applets und Daten)
- Übertragungsrate: 9,6 kBit/s

Probleme bereitet insbesondere die Knappheit an Arbeitsspeicher. Es ist zwar ausreichend EEPROM-Speicherplatz vorhanden, der auch zum Speichern von Zwischenergebnissen verwendet werden könnte, das Schreiben in EEPROM-Speicherzellen ist jedoch wesentlich langsamer als das Schreiben in RAM, zudem ist technikbedingt nur eine relativ geringe Anzahl von Schreibvorgängen auf EEPROM-Speicherzellen möglich, bevor diese dadurch irreversibel zerstört werden [Chen00].

Native-Code-Karten

Auf Grund der interpretierenden Arbeitsweise ist die Programmausführung auf Java-Karten langsamer als auf Karten, die prozessorspezifischem Native Code des eingebetteten Mikroprozessors ausführen. Daher wurde der Unterschriftserkennungsalgorithmus auch als Prototyp auf einer Smartcard implementiert, die Native Code ausführt, und zwar auf einer Funcard 2 [Atm01].

5.3 Entwurf des On-Card-Matching-Verfahrens

5.3.1 Auswahl einer Analysemethode

In [Sch98] werden verschiedene Methoden zur Analyse von On-line-Unterschriften diskutiert, die auch in Kombination eingesetzt werden können:

- Statistische Analyse: Die statistische Analyse betrachtet nur die allgemeine Ausprägung einer On-line-Unterschrift und abstrahiert von deren geometrischer Form.
- Strukturelle Analyse im Ortsbereich: Bei der strukturellen Analyse werden Eigenschaften, die in der geometrischen Form einer zu prüfenden On-line-Unterschrift wahrnehmbar sind, mit berücksichtigt. Die Unterschrift kann hierzu z. B. an Hand markanter Schreibpunkte in verschiedene Unterschriftssegmente segmentiert werden.

- Analyse im Zeitbereich: Bei der Analyse im Zeitbereich werden Abtastfolgen, d. h. Folgen von Werten zu den Abtastzeitpunkten, als zu vergleichende Merkmale verwendet. Mittels dynamischer Optimierung wird eine nichtlineare Zeitanpassung der Abtastfolgen der zu prüfenden On-line-Unterschrift an die Referenzdaten vorgenommen und dabei ihr Abstand ermittelt. Die Merkmalsextraktion besteht hauptsächlich nur in der geeigneten Formatierung der aufgenommenen Abtastfolgen.
- Analyse im Frequenzbereich: Die Zeitfunktionen für die zu prüfenden On-line-Unterschrift können in den Frequenzbereich transformiert und die Koeffizienten im Frequenzbereich als zu vergleichende Merkmale verwendet werden. Hierfür wird eine zeitdiskrete Wavelet-Transformation angewendet. Im Gegensatz zur FOURIER-Transformation, die eine Darstellung der betrachteten Größe in Abhängigkeit von der Frequenz, aber unabhängig von der Zeit liefert, liefert eine Wavelet-Transformation eine Darstellung der betrachteten Größe in Abhängigkeit von Frequenz und Zeit.

Wegen ihrer hohen Fehlerraten reicht die statistische Analyse allein nicht aus. Auf Grund der beschränkten Rechenleistung erscheint eine Analyse im Frequenzbereich derzeit nicht auf einer Smartcard machbar zu sein. Die strukturelle Analyse im Ortsbereich und die Analyse im Zeitbereich kämen für eine Realisierung auf einer Smartcard in Betracht. Für eine spätere Standardisierung des Formats der an der Chipkartenschnittstelle zu übertragenden Daten bietet es sich an, direkt die vom Tablett aufgenommenen und vorverarbeiteten Abtastfolgen für x und y, eventuell ergänzt um Folgen für Geschwindigkeit, Beschleunigung, Krümmung u. ä., als biometrische Merkmale anzusehen. Aus diesen Gründen wird für die Realisierung als On-Card-Matching-Algorithmus die Analyse im Zeitbereich ausgewählt.

5.3.2 Vorverarbeitung der aufgenommenen On-line-Unterschriften

Ziel der Vorverarbeitung der aufgenommenen On-line-Unterschriften ist es, unwesentliche Informationen, die Ausdruck zufälliger Schwankungen beim Unterschreiben, jedoch nicht Ausdruck der individuellen Unterschriftsdynamik sind, zu unterdrücken sowie Unterschiede zwischen den mit verschiedenen grafischen Tabletts aufgenommenen Daten auszugleichen. Zum Beispiel sind die Größe einer Unterschrift sowie ihre Position und Ausrichtung auf dem grafischen Tablett nicht schreibertypisch, sondern variieren zufällig. Unterschiede zwischen den mit verschiedenen grafischen Tabletts aufgenommenen Daten entstehen z. B. durch unterschiedliche Abtastraten und Auflösungen der Tabletts. Abbildung 22 zeigt die von einem grafischen Tablett mit einer Abtastrate von 200 Hz gelieferten Folgen von x- und y-Koordinaten einer Muster-Unterschrift. Zur Vorverarbeitung sind die folgenden Schritte erforderlich [Sch98, Wir98]:

- Bestimmen des Schreibanfangs und -endes: Der Schreibanfang wird auf den Zeitpunkt gesetzt, zu dem der Stift das erste Mal das grafische Tablett berührt. Das Schreibende wird auf den Zeitpunkt gesetzt, zu dem der Stift das letzte Mal das grafische Tablett berührt. Stiftbewegungen im Nahbereich des Tabletts vor bzw. nach diesen Zeitpunkten werden nicht als identitätstragend angesehen und ignoriert.
- Füllen kleiner zeitlicher Lücken in den Abtastfolgen: Durch Interpolation und anschließende äquidistante Abtastung werden kleine zeitliche Lücken in den Abtastfolgen überbrückt, die entstehen, falls der Stift während des Unterschreibens kurzzeitig den Nahbereich des grafischen Tabletts verlässt. Solche Lücken treten bei der gleichen Person nicht in jeder On-line-Unterschrift auf und werden daher nicht als identitätstragend angesehen.
- Tiefpassfilterung zur Eliminierung des Rauschens
- Translation und Rotation des Schriftbilds: Die Position einer On-line-Unterschrift auf einem grafischen Tablett und der Winkel zwischen ihrer Hauptrichtung und der *x*-Achse können von Unterschrift zu Unterschrift der gleichen Person variieren und werden nicht als identitätstragend angesehen. Die Unterschriften sind so zu verschieben und zu drehen, dass der Mittelwert der *y*-Koordinaten gleich 0 ist.
- Entfernen des Linearanteils: Die auf Grund des üblichen Schreibens von links nach rechts in der *x*-Folge enthaltene waagerechte Schreibbewegung (d. h. der Linearanteil) wird entfernt. Dadurch wird auch der Mittelwert der *x*-Koordinaten gleich 0 und die *x*-Koordinaten werden auf den gleichen Wertebereich wie die *y*-Koordinaten abgebildet.
- Normierung: Die Größe von On-line-Unterschriften kann geringfügig variieren und wird nicht als identitätstragend angesehen. Darum erfolgt eine Normierung der *x*- und *y*-Folgen unter Beibehaltung des Verhältnisses zwischen *x*- und *y*-Bewegung. Durch die Normierung werden unterschiedlich große On-line-Unterschriften einer Person vergleichbar. Außerdem werden unterschiedliche *x*- und *y*-Auflösungen unterschiedlicher grafischer Tabletts ausgeglichen.

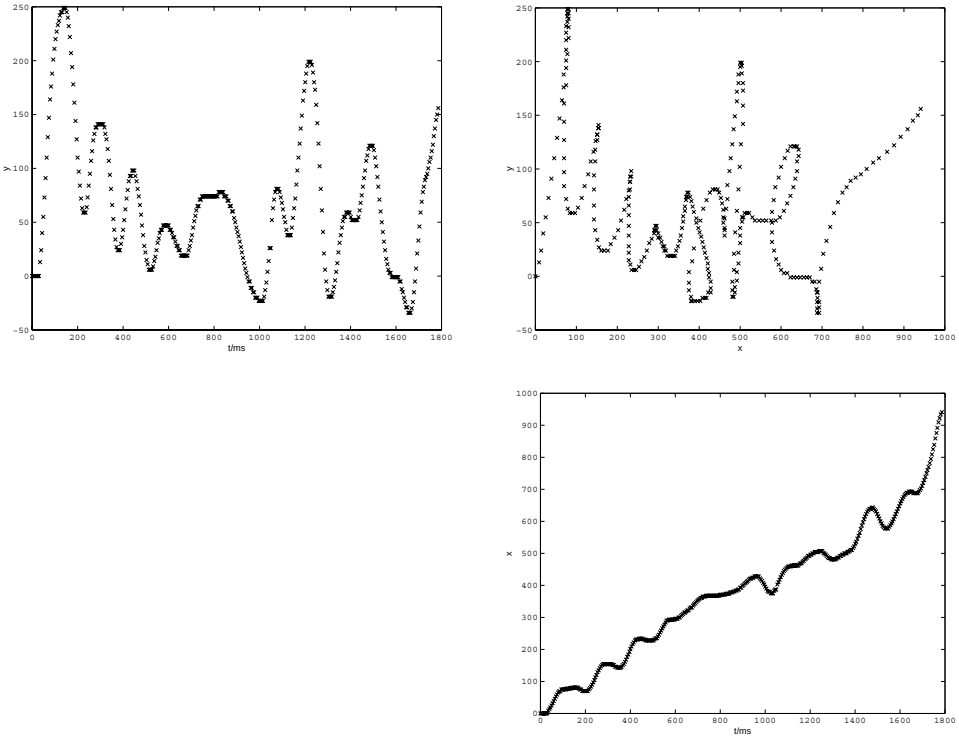


Abbildung 22 *y*,*t*-, *x*,*y*- und *x*,*t*-Diagramm einer aufgenommenen On-line-Unterschrift

5.3.3 Merkmalsextraktion

Die zu vergleichenden Merkmale sind zum einen die direkt mit dem grafischen Tablett aufgenommenen und vorverarbeiteten Abtastfolgen für die *x*- und *y*-Koordinaten, zum anderen lassen sich aus diesen Abtastfolgen durch Differenzenbildung Folgen für die *x*- und *y*-Geschwindigkeit ableiten. Die Berechnung der Folgen für die *x*- und *y*-Geschwindigkeit auf der Smartcard kostet weniger Zeit als ihre Berechnung außerhalb der Smartcard und ihre anschließende Übertragung über die langsame Chipkartenschnittstelle. Daher erfolgt die Berechnung der Geschwindigkeiten nicht in der Merkmalsextraktionskomponente im Dienstleistungssystem, sondern auf der Smartcard.

Die Abtastrate sollte einerseits möglichst gering sein, damit möglichst wenige Daten an die Smartcard übertragen und auf ihr gespeichert und verarbeitet werden brauchen. Andererseits ist bei der Wahl der Abtastrate das NYQUIST/SHANNONsche Abtasttheorem zu beachten: Will man ein Signal digitalisieren, dessen Frequenzspektrum bei Zerlegung in harmonische Schwingungen (FOURIER-Transformation) als höchste Frequenz die Frequenz f enthält, so muss man mit einer Abtastrate von mindestens $2f$ abtasten, um das Signal rekonstruieren zu können. Wir gehen davon aus, dass eine Abtastung mit 100 Hz ausreichend ist, um keine wesentlichen Informationen über eine aufgenommene On-line-Unterschrift zu verlieren.

Angenommen, die *x*- und *y*-Folgen wären normalverteilt [Bro99], dann lägen etwa 70% der normierten *x*- und *y*-Koordinaten im Intervall zwischen –1 und 1. Gleitkommazahlen sind jedoch auf Grund ihres Platzbedarfs und der fehlenden Gleitkomma-Arithmetik auf Java-Karten nicht für die Übertragung zur Smartcard geeignet. Für die Übertragung zur Smartcard ist es zweckmäßig, die *x*- und *y*-Koordinaten so zu skalieren, dass sie jeweils in einem vorzeichenbehafteten Byte kodiert werden können, also ganze Zahlen im Intervall zwischen –128 und 127 sind. Wiederum angenommen, die *x*- und *y*-Folgen wären normalverteilt, dann lägen nahezu 100% der normierten *x*- und *y*-Koordinaten im Intervall zwischen –4 und 4. Damit nahezu alle Werte im gewünschten Intervall zwischen –128 und 127 liegen, erfolgt eine Skalierung der normierten *x*- und *y*-Koordinaten durch Multiplikation mit 128/4 = 32 und anschließendes Runden auf die nächstgelegene ganze Zahl. Sollte einmal ein Wert nach der Skalierung außerhalb des gewünschten Intervalls liegen (was zwar möglich, aber sehr unwahrscheinlich ist, falls die Werte tatsächlich annähernd normalverteilt sind), wird er durch –128 bzw. 127 ersetzt. Durch das Runden auf ganze Zahlen tritt ein Quantisierungsfehler auf. Wir gehen davon aus, dass der Quantisierungsfehler, gemessen an den zufälligen Schwankungen von Unterschrift zu Unterschrift einer Person, klein ist und daher vernachlässigt werden kann.

Abbildung 23 zeigt das mit geringerer Abtastrate (100 Hz) abgetastete, verschobene, gedrehte, von der waagerechten Bewegung befreite, normierte und skalierte Abbild der Muster-Unterschrift aus Abbildung 22.

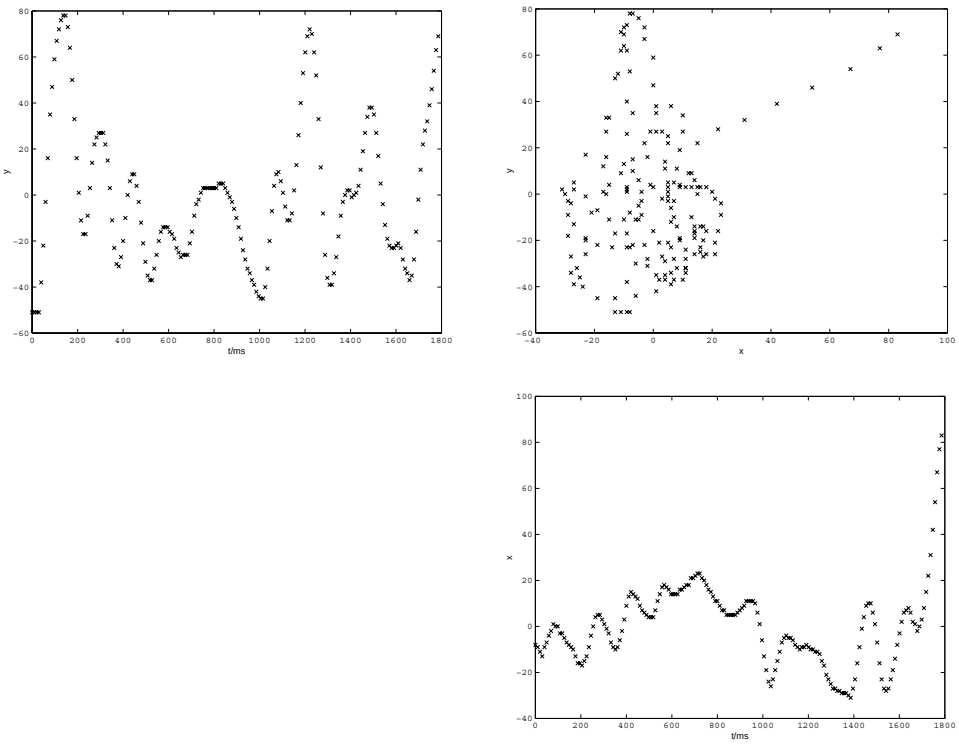


Abbildung 23 *y*,*t*-, *x*,*y*- und *x*,*t*-Diagramm der On-line-Unterschrift aus Abbildung 22 nach Translation und Rotation des Schriftbilds, Entfernung des Linearanteils, Normierung und Skalierung sowie Anpassung der Abtastrate

5.3.4 Merkmalsvergleich

Beim Merkmalsvergleich wird der „Abstand“ der zu prüfenden On-line-Unterschrift zur Referenzunterschrift als Maß für die Ähnlichkeit der beiden Unterschriften ermittelt. Wenn der Abstand der zu prüfenden On-line-Unterschrift zur Referenzunterschrift größer ist als ein Schwellenwert, wird die Unterschrift als gefälscht angesehen, andernfalls als authentisch. Als Referenzunterschrift wird eine von mehreren in der Enrollment-Phase aufgenommenen On-line-Unterschriften einer zu authentisierenden Person ausgewählt, und zwar die, die die geringsten Abstände zu den übrigen Unterschriften besitzt.

Die beiden zu vergleichenden On-line-Unterschriften haben i. allg. unterschiedliche Längen. Die Schreibgeschwindigkeit einer Person variiert geringfügig von Unterschrift zu Unterschrift und zwischen den einzelnen Schriftzügen können größere oder kleinere Pausen eingelegt werden. Um zu vermeiden, unterschiedliche Unterschriftsabschnitte miteinander zu vergleichen, wird eine nichtlineare Zeitanpassung der Verifikationsdaten an die Referenzdaten vorgenommen [Sch98, Wir98]. Im Gegensatz zur linearen Zeitanpassung, bei der die Zeitachse der Verifikationsdaten als Ganzes auf die Länge der Referenzdaten gedehnt oder gestaucht würde, wird bei der nichtlinearen Zeitanpassung die Zeitachse der Verifikationsdaten in einigen Abschnitten gedehnt, in anderen Abschnitten gestaucht oder unverändert gelassen.

Das Verfahren, das die nichtlineare Zeitanpassung leistet, ist die dynamische Optimierung [Bro99]. Ziel ist, eine Abbildung zwischen den Zeitachsen der zu vergleichenden On-line-

Unterschriften (Verzerrungspfad) zu ermitteln, die den Abstand der zu vergleichenden Unterschriften unter Beachtung bestimmter Nebenbedingungen minimiert. Nebenbedingungen sind, dass Anfangs- und Endpunkte der Verifikations- und Referenzdaten aufeinander abgebildet werden müssen, dass die Reihenfolge der Zeitpunkte in der Abbildung beibehalten werden muss und dass bei der Abbildung kein Zeitpunkt ausgelassen werden darf. Der Abstand von Verifikations- und Referenzdaten ist die Summe der lokalen Abstände entlang des optimalen Verzerrungspfads. Der lokale Abstand zwischen dem Merkmalsvektor an einem Punkt der Verifikationsdaten (bestehend aus *x*- und *y*-Koordinate sowie Geschwindigkeit in *x*- und *y*-Richtung zu diesem Zeitpunkt) und dem Merkmalsvektor an einem Punkt der Referenzdaten kann auf verschiedene Weise definiert werden. Auf der Smartcard verwenden wir als möglichst einfach zu berechnendes Abstandsmaß die Absolutbetragsnorm.

Algorithmen zur dynamischen Optimierung sind in der Literatur beschrieben (z. B. [Wir98]) und sollen hier nicht wiederholt werden. Die dynamische Optimierung ist mit hohem Rechenaufwand verbunden. Der Rechenaufwand wird verringert, indem der optimale Verzerrungspfad nur in einem eingeschränkten Bereich gesucht wird (Sakoe-Chiba-Band [SC80]).

5.4 Testergebnisse

Wir haben eine erste Technologiebewertung unserer Prototypimplementierungen durchgeführt, indem wir sie gegen eine Unterschriftsdatenbank getestet haben, die innerhalb unserer Forschungsgruppe gesammelt wurde. Es wurden Unterschriften von 5 Kollegen gesammelt. Die Unterschriften wurden auf einem WACOM-Tablett aufgenommen. Die Datenbank enthält insgesamt 50 echten Unterschriften und 69 Fälschungsversuche. Für eine statistisch signifikantere Technologiebewertung ist eine größere Datenbank erforderlich.

Für die Fälschungsversuche hatten die Angreifer die zu fälschenden Original-Unterschiften auf Papier vorliegen (ein ziemlich realistisches Szenario) und konnten nach dem Besten ihrer Fähigkeit Fälschungen produzieren. Ihnen wurde erlaubt, die zu fälschenden Unterschriften zu üben, beim Fälschen auf die Vorlage zu schauen oder die Vorlage nachzuzeichnen.

Abbildung 24 fasst die Testergebnisse in einer Receiver Operating Characteristic zusammen [Man02]. Die Equal Error Rate (FRR gleich FAR) liegt bei ungefähr 8%. Die Rechenzeit, die zur Übertragung der Verifikationsdaten an die Karte und für das On-Card-Matching erforderlich ist, wächst linear mit der Größe der Verifikationsdaten. Für eine On-line-Unterschrift, die zu schreiben 1 s gedauert hat, beträgt die zur Unterschriftserkennung erforderliche Rechenzeit auf der Sm@rtCafé-Java-Karte etwa 10 s und auf der Native-Code-Karte etwa 3,5 s. Für einen praktischen Einsatz ist eine weitere Verringerung der Fehlerraten und der benötigten Rechenzeit wünschenswert und auch machbar [HF03, HF04].

5.5 Zusammenfassung und Ausblick

Der Vorteil einer Benutzerauthentisierung mittels handschriftlicher Unterschriften ist ihre hohe Benutzerakzeptanz. Handschriftliche Unterschriften sind als Mittel zur Authentisierung

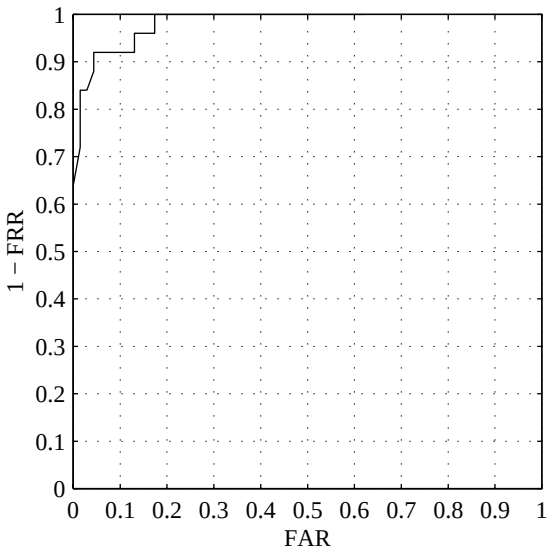


Abbildung 24 Receiver Operating Characteristic

von Personen vielerorts seit langem akzeptiert und werden als Ausdruck einer willentlichen Entscheidung des Schreibers angesehen.

Für die Implementierung auf einer Smartcard wurde die Analyse der On-line-Unterschriften im Zeitbereich ausgewählt. Die zu vergleichenden Merkmale sind die vom grafischen Tablett gelieferten Abtastfolgen der *x*- und *y*-Koordinaten und die aus ihnen abgeleiteten Folgen für die Geschwindigkeit in *x*- und *y*-Richtung. Mittels dynamischer Optimierung wird eine nicht-lineare Zeitanpassung der zu vergleichenden Unterschriften vorgenommen und dabei ihr Abstand als Maß ihrer Ähnlichkeit ermittelt.

Im Rahmen einer Machbarkeitsstudie wurde ein erster Prototyp des On-Card-Matching-Verfahrens für On-line-Unterschriften auf einer Java-Karte implementiert. Die Herausforderung bestand darin, den Merkmalsvergleich mit den in der Smartcard verfügbaren, beschränkten Ressourcen zu bewerkstelligen.

Eine statistische Bewertung der mit dem Verfahren erreichbaren Falschakzeptanz- und Falschrückweisungsraten an Hand umfangreicher Unterschriftsdatenbanken steht noch aus. Eine Verbesserung der Erkennungsleistung und des Zeitverhaltens wird von der in [Wir98] diskutierten Zerlegung der On-line-Unterschriften in Tablettschriftzüge (bei denen der Stift das grafische Tablett berührt) und Nahbereichsschriftzüge (bei denen der Stift das Tablett nicht berührt) erwartet, dies ist jedoch noch nicht im ersten Prototyp implementiert.

6 Untersuchung zur Überwindungssicherheit biometrischer Sensoren

6.1 Einführung

Im Gegensatz zur PIN oder dem Passwort als Authentisierungsmerkmal ist die Abgrenzung zwischen korrekter und falscher Identifikation bei der Biometrie deutlich schwieriger. Eine PIN kann bei der Eingabe jedes Mal 100% gleich reproduziert werden, während bei biometrischen Verfahren die Erfassung mit Toleranzen erfolgen muss, da z.B. ein Finger nicht immer gleich aufgelegt werden kann, sodass Rotationswinkel, Anpressdruck, abgebildete Fläche usw. übereinstimmen. Dieser Zwang zur Toleranz bei der Aufnahme ermöglicht es natürlich einem Angreifer zu versuchen, ebenfalls eine gute Fälschung zu erzeugen, die so nahe am Original liegt, dass diese innerhalb der Toleranz fälschlich akzeptiert wird.

Neben der reinen Überwindungssicherheit spielt auch die Datenübertragung und die Speicherung der biometrischen Merkmale eine Rolle, wenn auch gegenüber der Überwindungssicherheit untergeordnet: Denn auch bei anderen Authentisierungsmethoden müssen Daten übertragen und gespeichert werden, sodass bei biometrischen Verfahren dieselben Methoden (z.B. Ruhestromüberwachung, verschlüsselte Speicherung, dezentrale Speicherung auf Smartcards) Anwendung finden können.

Die Datenübertragung oder -speicherung – wenn ungeschützt – kann jedoch biometrische Daten preisgeben, die wiederum für Angriffe mit Fälschung auf den Sensor verwendet werden können.

6.2 Untersuchung der Überwindungssicherheit von Fingerabdrucksensoren

Im Rahmen des Projektes wurden zehn unterschiedliche Fingerabdrucksensoren auf Überwindungssicherheit untersucht. Diese umfassten alle derzeit gängigen Erfassungsmethoden: Neben den weit verbreiteten optischen und kapazitiven Sensoren, die den Markt zu großen Teilen abdecken, wurden auch ein Sensor mit Dermis-Erkennung und ein Sensor, der auf Druck reagiert untersucht. Hierfür wurden sowohl Fälschungen auf Basis des Fingers direkt wie auch von latenten Fingerabdrücken erstellt. Durch Versuchsreihen mit unterschiedlichen Materialien wurde die für den jeweiligen Sensor optimale Zusammensetzung ermittelt und entsprechende Angriffe durchgeführt. Die Ergebnisse hierfür werden in einer Fraunhofer-Publikation zusammengefasst (siehe Kapitel 6.4).

6.3 Untersuchung der Übertragungssicherheit eines Grafiktabletts für Unterschriftenerfassung

Als zweite Biometrie neben der Fingerabdruckerkennung kam im Rahmen des Projektes ZAVIR die Unterschriftenerkennung zum Einsatz. Hier ist eine reproduzierbare und nachvollziehbare Art und Weise des Angriffs schwierig. Ein geübter „Fälscher“ wäre bei bestimmten Unterschriften u.U. in der Lage eine Falscherkennung herbeizuführen. Dies ist jedoch nicht immer wiederholbar. Außerdem kommt es darauf an, ob die zu fälschende Unterschrift dem Angreifer „liegt“, z.B. weil sie seiner Handschrift ähnelt.

Eine andere Möglichkeit ist die Reproduktion mittels eines Industrieroboters, der den Stift des Tabletts führt. Da jedoch sehr hohe Beschleunigungen und schnelle Bewegungswechsel (z.B. bei Schleifen) bei hoher räumlicher Genauigkeit notwendig sind, ist dies nur mittels eines komplexeren Aufbaus möglich.

Aus diesen Gründen wurde von einem direkten Angriff auf die Erfassung abgesehen, da ein anderer Punkt erfolgversprechender erschien: die Datenübertragung.

Bei dem per USB angeschlossenen Tablett wurde die Datenübertragung belauscht und untersucht, ob das Protokoll für die Stiftbewegungen daraus extrahiert werden kann. Die Ergebnisse dieser Untersuchung werden in einer Publikation der Fraunhofer-Gesellschaft zur Verfügung gestellt (siehe Kapitel 6.4).

6.4 Verfügbarkeit der Ergebnisse

Die innerhalb der Untersuchung gesammelten Ergebnisse bergen offensichtlich eine gewisse Brisanz in punkto Sicherheit. Die Erstellung der Fingerabdruckfälschungen wird detailliert beschrieben, sodass diese bei Veröffentlichung eine Anleitung auch für weniger erfahrene Angreifer darstellt. Gleiches gilt für die Protokoll-Ergebnisse aus der Untersuchung der Datenübertragung des Grafiktabletts. Um eine Reproduzierbarkeit und Vergleichbarkeit der Ergebnisse auch bei anderen Gruppen, die sich mit Überwindungssicherheit beschäftigen, zu ermöglichen, werden die Hersteller der Sensoren genannt. Da die Intention der Untersuchung nicht eine Schwächung der Biometrie durch Anleitung zur Überwindung ist, wird an dieser Stelle von der Darstellung der Untersuchungsergebnisse abgesehen. Um dennoch einen kontrollierten Zugriff darauf zu ermöglichen, wird der Untersuchungsbericht als Publikation der Fraunhofer-Gesellschaft verfügbar gemacht. Diese ermöglicht eine genauere Kontrolle und Verwertung der Ergebnisse.

7 Zusammenfassung und Ausblick

Nur wenn Manipulationen an der Signaturanwendung ausgeschlossen sind und die zu signierenden Dokumente in eindeutig darstellbaren Dateiformaten vorliegen, kann man sicher sein, dass tatsächlich das signiert wird, was auf dem Bildschirm angezeigt wird. Außerdem muss die Benutzerauthentisierung einen Nachweis der Urheberschaft erlauben. Die übliche PIN gestattet nur einen schwachen Nachweis, da sie in die Hände Unberechtigter gelangen kann. Ziel des Projektes war es daher, einerseits eine vertrauenswürdige, vor Manipulationen geschützte Signierumgebung (Trusted Signature Terminal, TST) zu entwickeln, und andererseits Beiträge zur Nutzbarmachung biometrischer Verfahren (Fingerabdruck- und Unterschriftserkennung) für die Benutzerauthentisierung auf Signaturkarten zu leisten, um eine verlässliche Zurechenbarkeit elektronischer Signaturen zu Personen zu erreichen und diese auch für die Empfänger signierter Dokumente nachvollziehbar zu machen.

Die wichtigsten Ergebnisse des Projekts sind

- Prototyp eines Trusted Signature Terminals als vertrauenswürdige Signierumgebung
- Prototyp einer Signaturkarte mit Fingerabdruck-On-Card-Matching, die in der Antwort auf Signaturbefehle anzeigt, ob das biometrische Verfahren zur Benutzerauthentisierung eingesetzt wurde, um die Zurechenbarkeit der erzeugten elektronischen Signaturen zum rechtmäßigen Karteninhaber zu erhöhen. Die Fingerabdruckdaten genügen einem standardisierten Datenformat [DIN66400] und werden an der Kartenschnittstelle nach wechselseitiger Authentisierung der Signaturkarte und einer in das Terminal eingebauten Sicherheitsmodulkarte durch Secure Messaging geschützt.
- Prototyp und Funktionsdemonstrator für Unterschrifts-On-Card-Matching
- Untersuchung zur Überwindungssicherheit biometrischer Sensoren

Der ZAVIR-Lösungsansatz bietet die Chance, die elektronische Signatur für höherwertige Geschäftsprozesse besser nutzen zu können. Es werden Partner für die Weiterentwicklung des Prototyps des Trusted Signature Terminals gesucht. Das Know-How und Komponenten zur sicheren Signaturerzeugung und biometrischen Benutzerauthentisierung, die im ZAVIR-Projekt entstanden sind, können in verschiedenen neuen Projekten genutzt werden.

Literaturverzeichnis

[Atm01] Atmel 8-bit AVR microcontroller with 8 Kbytes in-system programmable flash – AT90S8515. Datasheet, 2001

[BaWo99] Baltus, R. ; Woop, M.-B.: Elektronische Verträge, rechtlich abgesichert mit der „digitalen Signatur“, oder: Wird die eigenhändige Unterschrift durch eine 8stellige PIN ersetzt? http://edvgt.jura.uni-sb.de/Tagung99/ak99/authentifikation.htm, September 1999

[BD02] Busch, C. ; Daum, H.: Frei von Zweifel? Biometrische Erkennung: Grundlagen, Verfahren, Sicherheit. *c't Magazin für Computertechnik*, Heft 05/2002, S. 156–161

[Bro99] Bronstein, I.N. ; Semendjadjew, K.A. ; Musiol, G. ; Mühlig, H.: Taschenbuch der Mathematik. 4., überarbeitete und erweiterte Aufl. der Neubearbeitung. Verlag Harri Deutsch, 1999

[CEM99] Common evaluation methodology for information technology security (CEM). Version 1.0, August 1999

[Chen00] Chen, Z.Q.: Java Card Technology for Smart Cards – Architecture and Programmer’s Guide. Addison-Wesley, 2000

[DIN66291-1] Chipkarten mit Digitaler Signatur-Anwendung/Funktion nach SigG und SigV – Teil 1: Anwendungsschnittstelle. DIN V 66291-1, 2000

[DIN66400] Finger Minutiae Encoding Format and Parameters for On-Card-Matching. Normentwurf DIN V 66400. November 2002

[Fre03] FreeImage – a free, open source graphics library. Documentation Library version 3.3.0. http://freeimage.sourceforge.net/

[G&D01] Reference Manual of the Java Card Operating System Sm@rtCafé 2.0 / Giesecke & Devrient. Edition 12/01, 2001

[G&D03a] Specification Signature Application StarCert for STARCOS SPK 2.3 version 7.0. Giesecke & Devrient. Final Version 2.93/Status 17. Februar 2003

[G&D03b] Specification Card Life Cycle StarCert v2.2 on STARCOS SPK 2.3 v. 7.0. Giesecke & Devrient. Version 1.92/Status 21. Februar 2003

[G&D03c] Security Target STARCOS SPK 2.4 with ZAVIR application, V1.0. Giesecke & Devrient. Version 1.1/Stand 18. Dezember 2003

[Har02] Hariwati, D.: Merkmalsvergleich in einer Smartcard für „Signature Dynamics“-Verfahren. Diplomarbeit, Technische Universität Darmstadt, Oktober 2002

[HF03] Henniger, O. ; Franke, K.: Biometrische Benutzerauthentisierung auf Smartcards mittels handschriftlicher Unterschriften. In: *Tagungsband des Schwerpunkts "Sicherheit – Schutz und Zuverlässigkeit" der Jahrestagung der Gesellschaft für Informatik* (Frankfurt/Main, September 2003)

[HF04] Henniger, O. ; Franke, K.: Biometric user authentication on smart cards by means of handwritten signatures. In: *Proceedings of the International Conference on Biometric Authentication* (Hong Kong, Juli 2004)

[HPC04] German Health Professional Card and Security Module Card Specification, Version 2.0 Revision 1, Mai 2004

[HSFU03] Henniger, O. ; Struif, B. ; Franke, K. ; Ulrich, R.: Trusted Signature Terminal – Eine vertrauenswürdige Signierumgebung. In: P. Horster (Hrsg.): *Tagungsband der Arbeitskonferenz D-A-CH Security* (Erfurt, März 2003)

[ISO15408] Information technology – Security techniques – Evaluation criteria for IT security (Common Criteria). International Standard ISO/IEC 15408. Dezember 1999

[ISO7816-4] Information Technology – Identification Cards – Integrated Circuit Cards – Part 4: Organization, Security and Commands for Interchange. Final Draft International Standard ISO/IEC 7816-4, 2004

[ISO7816-6] Information Technology – Identification Cards – Integrated Circuit Cards – Part 6: Interindustry Data Elements for Interchange. International Standard ISO/IEC 7816-6, 2004

[ISO7816-8] Information Technology – Identification Cards – Integrated Circuit Cards – Part 8: Commands for Security Operations. International Standard ISO/IEC 7816-8, 2004

[ISO7816-11] Information Technology – Identification Cards – Integrated Circuit Cards – Part 11: Personal Verification through Biometric Methods. International Standard ISO 7816-11, 2004

[ISO7816-15] Information Technology – Identification Cards – Integrated Circuit Cards – Part 15: Cryptographic Information Application. International Standard ISO 7816-15, 2004

[ITSEC91] Information Technology Security Evaluation Criteria (ITSEC); Provisional Harmonized Criteria. Juni 1991

[Man02] Mansfield, A.T. ; Wayman, J.L.: Best practices in testing and reporting performance of biometric devices. Version 2.0, August 2002

[Mas01] Mascolo, G. ; Neumann, C.: Trojanische Pferde. *Spiegel* 24/2001

[MKT99-3] Multifunktionale Kartenterminals (MKT) für das Gesundheitswesen und andere Anwendungsgebiete – Teil 3: CT-API 1.1 – Anwendungsunabhängiges Card-Terminal Application Programming Interface. April 1999.

[MKT99-4] Multifunktionale Kartenterminals (MKT) für das Gesundheitswesen und andere Anwendungsgebiete – Teil 4: CT-BCS – Anwendungsunabhängiger Card-Terminal Basic Command Set. Version 1.0, April 1999.

[MMJH02] T. Matsumoto, H. Matsumoto, K. Yamada, S. Hoshino: Impact of artificial “gummy” fingers on fingerprint systems. In: *Proceedings of SPIE* Vol. 4677, Januar 2002.

[Nal99] Nalwa, V.S.: Automatic on-line signature verification. In: Jain, A. ; Bolle, R. ; Pankanti, S.: *Biometrics: Personal identification in networked society*. Kluwer Academic Publishers, 1999

[OBEX03] Infrared Data Association (IrDA): Object Exchange Protocol. Version 1.3, 2003

[Phi00] Philips, P.J. ; Martin, A. ; Wilson, C.L. ; Przybocki, M.: An introduction to evaluating biometric systems. In: *IEEE Computer*, Februar 2000, S. 56–63

[Pla89] Plamondon, R. ; Lorette, G.: Automatic signature verification and writer identification – The state of the art. In: *Pattern Recognition* 22 (1989), Nr. 2, S. 107–131

[Poy94] Poyner, R.: Wintab Interface Specification 1.0: 16- and 32-bit API Reference. Dezember 1994

[Ran99] Rankl, W. ; Effing, W.: *Handbuch der Chipkarten*. Hanser, 1999

[RSA01] Public-Key Cryptography Standards (PKCS) #11, Cryptographic Token Interface Standard v2.11, RSA Laboratories, Bedford/Maine/USA, June 2001

[RSA93] Public-Key Cryptography Standards (PKCS) #7, Cryptographic Message Syntax Standard v1.5, RSA Laboratories, Bedford/Maine/USA, November 1993

[SC80] Sakoe, H. ; Chiba, S.: Dynamic programming optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech and Signal Processing* 26 (1980), pp. 623–625

[Sch01] Scheibelhofer, K.: Signing XML documents and the concept of "what you see is what you sign". TU Graz, Master's Thesis, 2001

[Sch02] Schneider, B.: Spezifikation eines erweiterten ISO7816-CIO-Konzepts zur Erstellung von Signaturkartenprofilen. Bachelor-Abschlussarbeit, Fachhochschule Darmstadt, April 2002

[Sch98] Schmidt, C.: On-line Unterschriftenanalyse zur Benutzerverifikation. RWTH Aachen, Dissertation, 1998

[ScSS00] Scheuermann, D. ; Schwiderski-Grosche, S. ; Struif, B.: Usability of biometrics in relation to electronic signatures. GMD Report 118, November 2000

[SigG01] Gesetz über Rahmenbedingungen für elektronische Signaturen (Signaturgesetz – SigG). 16. Mai 2001

[SigV01] Verordnung zur elektronischen Signatur (Signaturverordnung – SigV). 16. November 2001

[Str01] Struif, B.: Use of biometrics for user verification in electronic signature smart cards. In: *Smart Card Programming and Security – Proceedings of the International Conference on Research in Smart Cards* (Cannes, Frankreich, September 2001)

[Sun00a] Java Card 2.1.1 Application Programming Interface. Sun Microsystems, Revision 1.0, Mai 2000

[Sun00b] Java Card 2.1.1 Runtime Environment (JCRE) Specification. Sun Microsystems, Revision 1.0, Mai 2000

[Sun00c] Java Card 2.1.1 Virtual Machine Specification. Sun Microsystems, Revision 1.0, Mai 2000

[Tak01] Tak, M.: Versetzung in Klasse 3: nicht gefährdet! Card-Forum 01/2001

[Tang03] Tang, H.: Trusted Viewer for Electronically Signed Documents. Master Thesis, Fachhochschule Darmstadt, Juni 2003

[Wal02] Waldmann, U.: Sicherung biometrischer Daten durch kryptographische Verfahren. Diplomarbeit, Technische Universität Darmstadt, September 2002

[Wir98] Wirtz, B.: Segmentorientierte Analyse und nichtlineare Auswertung für die dynamische Unterschriftsverifikation. TU München, Dissertation, 1998

[WSE03] Waldmann, U. ; Scheuermann, D. ; Eckert, C.: Schutz biometrischer Daten bei Authentisierung auf Smartcards. In: P. Horster (Hrsg.): *Tagungsband der Arbeitskonferenz D-A-CH Security* (Erfurt, März 2003)

[WSE04] Waldmann, U. ; Scheuermann, D. ; Eckert, C.: Protected Transmission of Biometric User Authentication Data for On-card Matching. In: *Proceedings of ACM Symposium on Applied Computing* (Nikosia, Zypern, März 2004)

[wxWid] wxWidgets (formely wxWindows), an open-source, cross-platform native (UI) framework, http://www.wxwidgets.org/