

Trusted Pocket Signer

Anwendungen elektronischer Signaturen mit einem persönlichen Signiergerät

Detlef Kraus und Christoph Sesterhenn

{detlef.kraus, christoph.sesterhenn}@src-gmbh.com

SRC Security Research & Consulting GmbH

BMW Förderkennzeichen 01MS201

1 Einleitung

Die Nutzung digitaler Signaturen insbesondere im geschäftlichen Bereich hat in den letzten Jahren weiter zugenommen. Beispielhaft seien an dieser Stelle die umfangreichen Möglichkeiten hierzu genannt, die sich im Bereich des eGovernment bieten, wie die Vielzahl von identifizierten, potenziell für den Einsatz digitale Signaturen geeigneten, elektronischen Geschäftsprozessen der Initiative ‚BundOnline 2005‘ oder die Ergebnisse der [media@Komm](#) Pilotprojekte zeigen. Alle Anwendungen machen deutlich, dass es eine Vielzahl von Einsatzmöglichkeiten gibt, rechtsverbindliche digitale Erklärungen abzugeben: sie werden häufig nicht in der eigenen Wohnung, sondern außer Haus in Geschäften, in Behörden, in Reisebüros, in Autowerkstätten oder in Arzt- und Anwaltspraxen abgegeben. Selbständige oder (leitende) Mitarbeiter mit Unterschriftsberechtigung signieren am Arbeitsplatz. In Zukunft ist auch verstärkt mit elektronischen Signaturen für e-commerce-Anwendungen über das Internet zu rechnen, die elektronische Signaturen z.B. bei Bestellvorgängen mit höherem Kaufwert erfordern.

Nach dem am 16. Mai 2001 in Kraft getretenen neuen Signaturgesetz [2] entfalten digitale Signaturen die gleiche Rechtswirkung wie eine handschriftliche Unterschrift. Das Gesetz regelt die erforderliche Sicherheitsinfrastruktur im Einklang mit der EU-Richtlinie. Insbesondere fordert das Signaturgesetz, dass Signaturanwendungskomponenten einzusetzen sind, die „feststellen lassen, auf welche Daten sich die Signatur bezieht“ (§17 Abs.2). Geht man davon aus, dass elektronische Erklärungen heute oft auf PCs erstellt und abgegeben werden, die sich zumeist nicht in einer vom Unterschreibenden kontrollierten Umgebung befinden, dann lässt sich diese Bedingung nur schwer erfüllen. Selbst die Erfahrung mit evaluierten - also bezüglich der Einhaltung von Sicherheitsbedingungen, wie sie das Signaturgesetz vorsieht, begutachteten Hard- und Softwarekomponenten - hat gezeigt, dass es schwer ist, in frei programmierbaren, online-fähigen PC-Umgebungen solche Eigenschaften zu garantieren [1].

Zur Signaturerzeugung wird ein persönliches Sicherheitsmodul (in der Terminologie der EU Direktive auch **Secure Signature Creation Device**, kurz SSCD genannt) genutzt, das mit Hilfe eines geheimen Signaturschlüssels eine digitale Unterschrift über die aus den Dokumentendaten erzeugten Hashwert generiert. Als SSCD wird eine nach den Common Criteria (CC) bzw. ITSEC evaluierte Chipkarte verwendet. Durch die Evaluation wird u.a. sichergestellt, dass der eindeutig einer Person zugeordnete Signaturschlüssel nicht aus der Chipkarte kopiert und von einer anderen Person benutzt werden kann. Vor der Verwendung des geheimen Signaturschlüssels erfolgt eine Benutzer-Authentikation über eine Geheimzahl (PIN) oder ein biometrisches Merkmal.

Die Verwendung eines Chipkartenlesers mit Benutzer-Authentisierungsfunktion (z.B. PIN-Pad) garantiert darüber hinaus, dass die zur Freischaltung einer Signaturerzeugung notwendige Persönliche Identifikations-Nummer oder - falls biometrische Benutzer-Authentisierung unterstützt - die biometrischen Verifikationsdaten nicht über den PC geleitet werden und daher vom PC her nicht ausforschbar sind.

Es ist das Ziel des vom BMWi (Bundesministerium für Wirtschaft und Technologie) geförderten TruPoSign-Projektes, ein signaturgesetzkonformes mobiles persönliches Signiergerät zu konzipieren, realisieren und in Pilotanwendungen zu erproben. Dieser Trusted Pocket Signer (kurz TPS) soll dabei als mobiler Chipkartenleser und sichere Anzeigekomponente für Dokumentinhalte eingesetzt werden und die Funktion der Freischaltung ("willful act") einer Signaturerzeugung bieten. Letztere erfolgt durch die Verifikation einer auf dem Touch Screen des TPS mit dem Stift ausgeführten Unterschrift bzw. Paraphe. Die Übertragung der Dokumente zwischen stationärem System und TPS erfolgt drahtlos z.B. über Bluetooth oder Infrarot. Mit der Entwicklung eines TPS kann die Sicherheit der Signaturerzeugung zuhause, am Arbeitsplatz und vor allem in Fremдумgebungen entscheidend verbessert werden.

2 TPS Basiskonzept

Der Trusted Pocket Signer (TPS) als mobiles, persönliches Signatur-Erzeugungssystem ist vom Aussehen her vergleichbar mit einem PDA. Da das Hauptanwendungsgebiet jedoch darin besteht, in den verschiedensten Einsatzumgebungen (z.B. Behörden, Geschäften, Anwaltskanzleien, Arztpraxen, Unternehmen und Heimumgebungen) rechtsgültige elektronische Signaturen unter Nutzung einer Signaturkarte zu erzeugen, sind Erweiterungen zu einem herkömmlichen PDA notwendig.

In der Grundausbaustufe weist der TPS die folgenden Hardware-Komponenten auf:

- eine CPU mit einem RISC Prozessorkern mit Cache für Instruktionen und Daten sowie einer unabhängigen MMU, damit hochkomplexe Rechenanforderungen wie Hashing, Dokumentenanzeige, Durchführung der Erfassung, Erkennung und Verifikation biometrischer Merkmale, mit ausreichender Performance durchgeführt werden können
- eine optimale, mobile Energieversorgung mit Powermanagementfunktionen
- einen Chip für Datum und Uhrzeit
- ein Farbdisplay für die Datenausgabe
- einen Touch Screen für die Dateneingabe
- einen Eingabestift
- eine Tastatur (ggf. nur auf dem Display angezeigt), wobei für die TPS Anwendung lediglich einige wenige Tasten erforderlich sind
- einen Ein-/Ausschalter und eine Reset-Taste
- einen akustischen Signalgeber
- geeignete Schnittstellen für die sichere Kommunikation mit dem PC

- zwei bis drei Kartenleser, wobei ein Leser in Standardgröße (mit absenkbaaren Kontakten) ausgeführt ist und zwei als SIM Modul. Eines der SIM Module dient evtl. als Sicherheitsmodul im Inneren des Gerätes

Das Betriebssystem des TPS muss insbesondere

- eine Multitasking-Funktion bieten. Hierbei sind folgende Sicherheitsanforderungen zu berücksichtigen:
 - Anwendungen und Betriebssystem müssen vor direkten Manipulationen aus anderen Anwendungen geschützt werden
 - Falls sich mehrere Benutzer einen TPS teilen, etwa in der Familie oder im Krankenhaus bei der Übergabe des Dienstes auf einer Station nach Beendigung des Schichtdienstes, muss gewährleistet sein, dass ein Benutzer nur Zugriff auf seine Daten hat.
- die Realisierung einer geeigneten Secure Download-Funktion gestatten
- Schutzmechanismen für die TPS-Signatur-Anwendung bieten, die deren Unversehrtheit garantieren
- ein Display-Interaktions-Handling unterstützen, das u.a. für das Handunterschriftenverfahren geeignet ist
- für den Ablauf von üblichen PDA-Anwendungen geeignet sein.

Aus den genannten allgemeinen Anforderungen resultiert, dass dem Sicherheitsmanagement eine besondere Bedeutung zukommt. Aus den Sicherheitsüberlegungen und Gründen der Portabilität der Anwendungen werden die TPS Module in JAVA realisiert.

Die folgenden TPS Ausbaustufen sind denkbar, wobei die einzelnen Varianten mehr oder weniger weitreichende Auswirkungen auf die Systemsicherheit des TPS haben. Man kann generell unterscheiden zwischen

- **Versiegelter TPS**

Der TPS ist nicht frei programmierbar, d.h. es können keine benutzerspezifischen Programme (nach-)geladen werden. Bei Auslieferung des Gerätes können neben der TPS-Signatur-Anwendung weitere TPS-Anwendungen (z.B. Terminkalender, Adressbuch etc.) installiert sein. Es ist aber nicht möglich, nachträglich weitere Anwendungen zu installieren. Es ist jedoch möglich, bereits installierte TPS-Komponenten zu ersetzen bzw. zu ergänzen (z.B. neuer Viewer für neuen Document Type; neues SmartCard Profil) und neue, autorisierte Versionen der bereits genutzten Anwendungen zu installieren.

- **Systemkontrollierter TPS**

Der TPS gestattet nur das Herunterladen von durch autorisierte Instanzen (Hersteller, Prüfstelle, ...) geprüften und signierten Anwendungen. Der Nutzer hat keine Möglichkeit, dies zu verändern.

- **Benutzerkontrollierter TPS**

Der TPS gestattet das Herunterladen von Software nur nach Verifikation durch den TPS-Benutzer und dessen ausdrücklicher Zustimmung mit entsprechenden Sicherheitshinweisen.

Die verschiedenen Ausbaustufen haben insbesondere Rückwirkungen auf die Evaluierbarkeit des Systems. Um die oben erwähnten Anforderungen des Signaturgesetzes zu erfüllen, wird die letztgenannte Variante im Rahmen des Projektes nicht weiter verfolgt.

Je nach Einsatzumgebung des TPS werden folgende Betriebsarten in Abhängigkeit von der Einsatzumgebung und der Anzahl der Benutzer des TPS unterschieden:

- **Single User Mode**

Der TPS wird nur von einer einzigen Person benutzt. Das Gerät gehört in der Regel dieser Person. Der TPS wird für diese Person personalisiert.

- **Multi User Mode**

Der TPS wird von mehreren Personen benutzt. Das Gerät gehört in der Regel einer Institution, die denselben TPS mehreren Mitarbeitern zur Verfügung stellen will. Der TPS wird auf die Nutzung durch diese Personen eingestellt.

- **Unknown User Mode**

Die Person, die den TPS benutzt, ist dem TPS nicht bekannt. Der TPS gehört in der Regel einem Dienstleistungsanbieter, der dem Kunden einen TPS zur Verfügung stellen will, falls dieser keinen eigenen besitzt. Bei dieser Nutzungsart erfolgt keine Personalisierung.

Die unterschiedlichen Benutzer-Modi haben Einfluss auf

- das zu erstellende Personalisierungskonzept
- die Bedienerführung und
- die Verfügbarkeit bestimmter TPS-Signatur-Anwendungs-Funktionen (z.B. Dokument-speicherungs-Funktion).

Die Zuordnung eines TPS zur Nutzung durch eine bestimmte Person, also die Personalisierung des TPS, wird im wesentlichen durch das Enrollment für die biometrische Benutzer Verifikation bestimmt, die am TPS als Signatur-Abschlußaktion („willful Act“) verwendet wird. Es ist klar, dass ein Enrollment im Unknown User Mode keinen Sinn macht. Im Multi-User Mode ist nach dem Enrollment eine (manuelle oder automatische) Selektion der Referenzdaten erforderlich. Dies ist für den Single User Mode nicht erforderlich.

Die Anzahl der beim Enrollment erforderlichen Handsignatur- bzw. Paraphen-Eingaben ist vom Ähnlichkeitsgrad abhängig. Üblicherweise sollten 3 im Sitzen und 3 im Stehen erfasst werden.

Der Handsignatur- bzw. Paraphenvergleich muss sehr schnell vonstatten gehen (nach Möglichkeit innerhalb 1 sec). Die Rate der fälschlicherweise zurückgewiesenen Benutzer (False Rejection Rate kurz FRR bezeichnet) sollte bei etwa 1% bis maximal 3% liegen. Um

die Risiken niedrig zu halten, darf es für einen möglichen Angreifer nicht möglich sein, beliebig oft die Eingabe einer Handsignatur oder Paraphe zu üben. Deshalb wird ein vom Betriebssystem des TPS verwalteter Wiederholungszähler realisiert, dessen Anfangswert z.B. den Wert 3 hat. Bei korrekter PIN-Eingabe der Signaturkarte wird der Retry Counter immer auf seinen Anfangswert gesetzt.

Für die Anzeige der zu signierenden Daten werden folgende Darstellungsmodi unterschieden:

Inhaltsdarstellung (Document Browsing)

Der eigentliche Erklärungsinhalt (Dokument, Nutzdaten) wird dargestellt. Das Konzept sieht vor, dass für das Document Browsing neben dem ASCII-Viewer als Standard-Viewer weitere anwendungsspezifische Viewer für bestimmte Dokumentarten (z.B. Bankformulare, Reiseanträge, Rezepte) existieren können, in denen die Darstellung in einer für den TPS geeigneten Form festgelegt ist. Zugänglich über diese Darstellung kann der gesamte Dokumentinhalt sein. Es kann sich aber auch um eine partielle Darstellung handeln, die dem Nutzer lediglich nur eine Stichprobenkontrolle des Inhalts ermöglicht. Alternativ können auch Konverter existieren, die das Dokument so aufbereiten, dass ein anderer Viewer (etwa der ASCII-Viewer) den gewünschten View erzeugen kann.

Abstrakte Darstellung (Abstract Viewing)

Falls das zu signierende Dokument nicht auf dem TPS Display dargestellt werden kann, entweder wegen eines zu großen Umfangs des Dokuments oder weil der TPS das Darstellungsformat nicht unterstützt, dann werden sowohl auf der PC-Seite als auch auf dem TPS-Display die zu signierenden Daten in "abstrakter Form" als schnell erfassbare graphische Symbole oder einfache Ziffernfolge dargestellt. Damit ist ein Scrollen der zu signierenden Daten auf dem TPS nicht nötig. Der Signierende sieht sein Dokument auf dem eigenen PC-Bildschirm und die Signatur-Berechnung erfolgt ohne Zeiteinbußen in der gesicherten TPS-Umgebung. Der Hashwert für den Abstract View wird nur auf dem PC berechnet und an den TPS übertragen, wo dieser dann zum Vergleichen ebenfalls über den Abstract View dargestellt wird. Da nur der übertragene Hashwert kontrolliert werden kann, sollte dieser Anzeige-Modus nur in einer PC-Systemumgebung verwendet werden, die unter Kontrolle des Unterschriftleistenden steht. Dies kann z.B. in Arztpraxen der Fall sein, wo die Umgebung als Sicherheitszone angesehen werden kann, da sonst beispielsweise die Verarbeitung personenbezogener, medizinischer Daten nicht erlaubt wäre.

Signatur-Attribute Viewing

Zusätzlich zum Dokument werden nach Bedarf auch die zu signierenden Signatur-Attribute präsentiert.

Nutzung des TPS und Einbindung in die Rechnerumgebung eines Anwenders

Der normale Ablauf bei der Nutzung des TPS zur Erstellung einer digitalen Signatur eines Dokuments ist etwa wie folgt:

1. Auf dem PC des Benutzers liegt ein Dokument vor, welches digital signiert werden soll. Hierbei kann es sich um eine E-Mail, ein mit Hilfe seines Textverarbeitungsprogramms erstelltes Dokument, ein Formular oder ein anderes beliebiges Dokument handeln.
2. Der Benutzer steckt seine Signaturkarte in den TPS und schaltet sie durch Eingabe seiner Geheimzahl frei. Danach ruft er die Signaturfunktion seines Rechners auf. Diese überträgt je nach unterstützter Anzeigefunktionalität entweder das Dokument oder den Hashwert an den TPS.
3. Der Benutzer kann sich auf dem Display des TPS das Dokument selbst ansehen oder die dargestellten Abstract Views vergleichen.
4. Nach der Autorisierung des Signaturprozesses durch den Benutzer übermittelt der TPS die Signaturdaten mit dem entsprechenden Kommando an die Chipkarte, die dann die Signatur erstellt.
5. Die Signatur wird vom TPS an den PC zurückgeschickt und dort in der laufenden Applikation weiter verarbeitet.
6. Falls die Abläufe es erfordern, dass mehrere Unterschriften nacheinander erstellt werden müssen, kann eine Signaturkarte eingesetzt werden, die nach **einmaliger** PIN-Eingabe mehrerer Signaturen erzeugen kann. Der TPS übergibt die zu signierenden Daten nur dann an die Signaturkarte, wenn durch eine mit dem Eingabestift auf dem Touch Screen ausgeführte Paraphe des Benutzers verifiziert werden konnte.

Bei dem Zusammenspiel von PC und TPS werden von der Anwendung auf dem PC bis hin zur Signatur-Anwendung auf dem TPS zahlreiche *Schichten* durchlaufen. Das grundlegende Basiskonzept ist in der nachfolgenden Abbildung dargestellt.

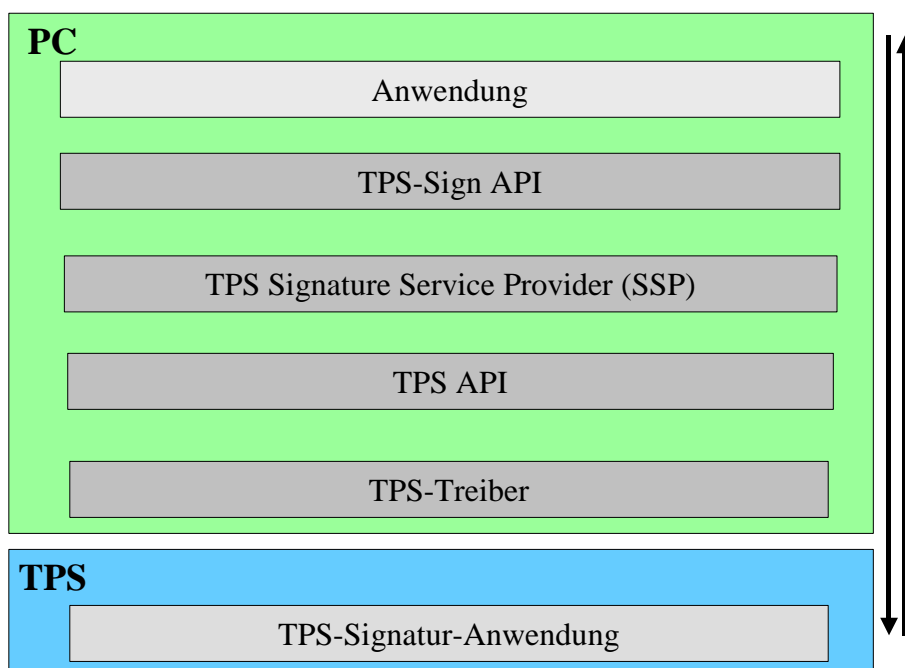


Abbildung 1: Kommunikation zwischen PC und TPS

Die TPS-Architektur bietet den Anwendungen über eine TPS-Sign-API eine Reihe von Funktionen an, welche die grundlegende Funktionalität einer Signaturkarte repräsentiert. Es werden die folgenden Basisfunktionen bereitgestellt:

- C_cs_authentication
- C_decrypt
- C_sign
- C_getCapabilities
- C_load_certificate
- C_load_certificate_reference
- C_read_certificate
- C_read_certificate_reference
- C_read_public_key
- C_icc_aut
- C_icc_ifd_init
- C_icc_ifd_aut
- C_read_public_key_ka

Diese Funktionen sind (bis auf C_getCapabilities) auch in der ZKA-SIG-API [5] umgesetzt.

Wenn weitere Funktionen einer Signaturkarte genutzt werden sollen (beispielsweise durch weitere Anwendungen auf der Karte), so werden diese nicht über die TPS-Sign-API umgesetzt. Vielmehr sind komplette Ergänzungen zu realisieren. So werden die zusätzlichen Funktionen über eine neue Schnittstelle (neben der TPS-Sign-API) auf dem PC und eine neue Anwendung auf dem TPS bereitgestellt.

Mit der folgenden Darstellung eines vereinfachten Ablaufs einer Signaturerzeugung soll ein erläuterndes Beispiel gegeben werden. Es ist dabei zu beachten, dass bei dem ausgewählten Beispiel davon ausgegangen wird, dass die PC-Anwendung die zu signierenden Daten übergeben kann. Der schematische Ablauf ist in der nachfolgenden Abbildung aufgezeigt.

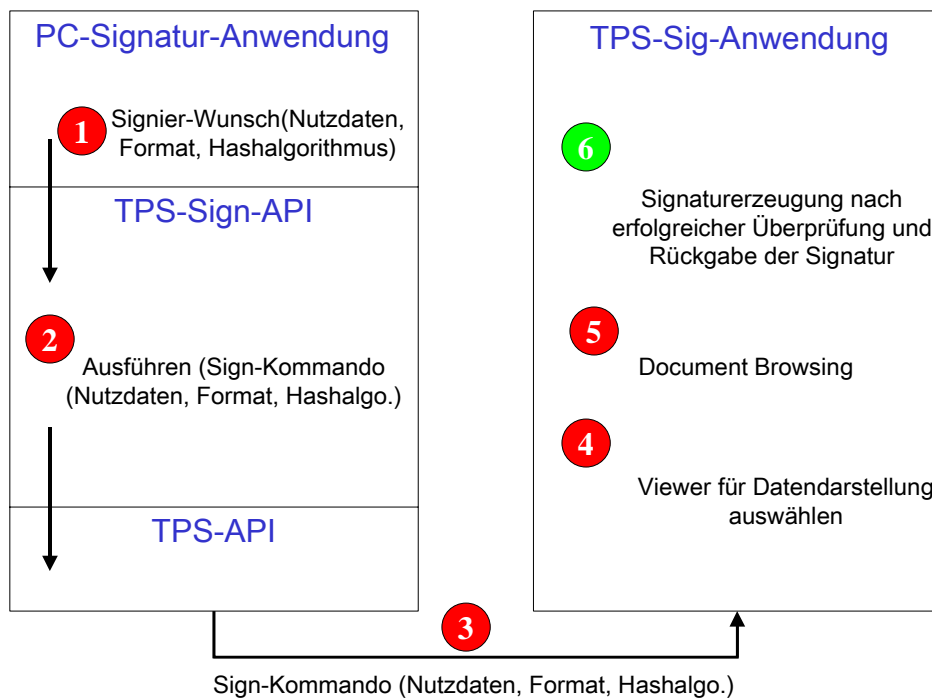


Abbildung 2: Ablauf-Beispiel Document Browsing

Bei den Abläufen gibt es zwei Varianten. Abbildung 2 zeigt das sogenannte Document Browsing. Dabei werden dem TPS neben den auf dem PC dargestellten Nutzdaten auch das entsprechende Format und der gewünschte Hashalgorithmus übergeben. Durch die Angabe des Formats können die Nutzdaten mit einem entsprechenden Viewer auf dem TPS dargestellt werden. Der Anwender kann nun direkt vergleichen, ob die zu signierenden Daten seinen Angaben auf der PC-Seite entsprechen.

Wenn die Nutzdaten auf dem TPS nicht darstellbar sind, da beispielsweise die Menge der zu signierenden Daten zu groß ist, wird auf den sogenannten Abstract View zurückgegriffen. Wie in Abbildung 3 veranschaulicht, wird auf der PC-Seite der Hashwert über die Nutzdaten errechnet und mittels des Abstract Viewers dargestellt. Dieser Hashwert wird zusammen mit dem Format der Daten und der Angabe des verwendeten Hashalgorithmus an den TPS übergeben. Auf dem TPS wird dann der übermittelte Hashwert ebenfalls mittels des Abstract Viewers visualisiert. Auf diese Art und Weise wird dem Anwender die Möglichkeit der Überprüfung gegeben, ob der Hashwert über die zu signierenden Daten auf TPS-Seite korrekt ist.

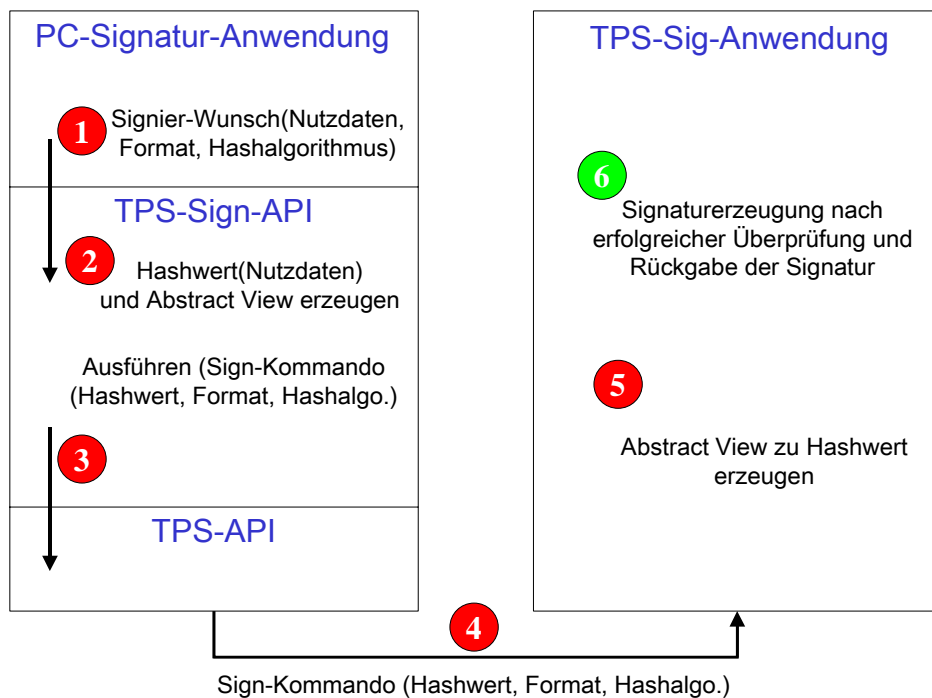


Abbildung 3: Ablauf-Beispiel – Abstract View

Anmerkung: Es muss dabei auf der PC-Seite gewährleistet sein, dass der Abstract View auch tatsächlich die am PC dargestellten Daten repräsentiert. Ein Angreifer könnte den Vorgang dahingehend manipulieren, dass an den TPS modifizierte Daten übertragen werden und der zu den modifizierten Daten gehörende Abstract View am PC angezeigt wird, indem er gegen den vermeintlichen Abstract View ausgetauscht wird. In diesem Fall würde der Anwender Nutzdaten signieren, die er nicht gesehen hat.

Zur Durchführung der dargestellten Funktionalitäten ist es erforderlich, die durch den TPS bereitgestellte Sicherheitstechnologie in bestehende bzw. noch zu entwickelnde Anwendungen zu integrieren. Hierzu wird i.d.R. eine Integration über standardisierte Schnittstellen angestrebt. Insbesondere bei Browser Technologien - wie beispielsweise bei Netscape und Microsoft – werden die Schnittstellen PKCS #11 sowie Microsoft Crypto API verwendet.

3 Einsatzszenarien

3.1 Nutzung des TPS in medizinischen Einrichtung

Eines der Szenarien, in dem der TPS eingesetzt werden soll, ist das Gesundheitswesen, speziell der Bereich, in dem intensiv medizinisch dokumentiert werden muss und diese Dokumente teilweise zur Weiterbehandlung weitergegeben werden sollen. Die Arbeitsabläufe in den verschiedenen Bereichen des Gesundheitswesens sind ihren Aufgaben entsprechend sehr unterschiedlich. Der Ansatz für das Szenario ist allgemein, seine Realisierung im Projekt fokussiert auf die Ausstattung von Praxen niedergelassener Ärzte und Abteilungen/Stationen stationärer Einrichtungen. Dort sollen erzeugte

elektronische Dokumente mit dem TPS signiert und nach einer eventuellen, verschlüsselten Übertragung entschlüsselt werden.

Grundsätzlich werden dabei Interaktionen zwischen Patient und beauftragtem Praxis- / Stationspersonal ohne eigene Unterschriften-Berechtigung für medizinische Dokumente unterschieden von Interaktionen zwischen dem Patienten und unterschriftsberechtigtem ärztlichem und pflegerischem Personal. In der nachstehenden Grafik sind diese beiden Gruppen vereinfacht als Stations-/Praxispersonal (nicht unterschriftsberechtigt im Sinn ärztlicher Verantwortung bzw. als Leistungserbringer nach SGB V) und Ärzte (unterschriftsberechtigt) dargestellt.

Die nachstehende Grafik veranschaulicht die charakteristischen Abläufe bei der ambulanten oder stationären medizinischen Betreuung eines Patienten.

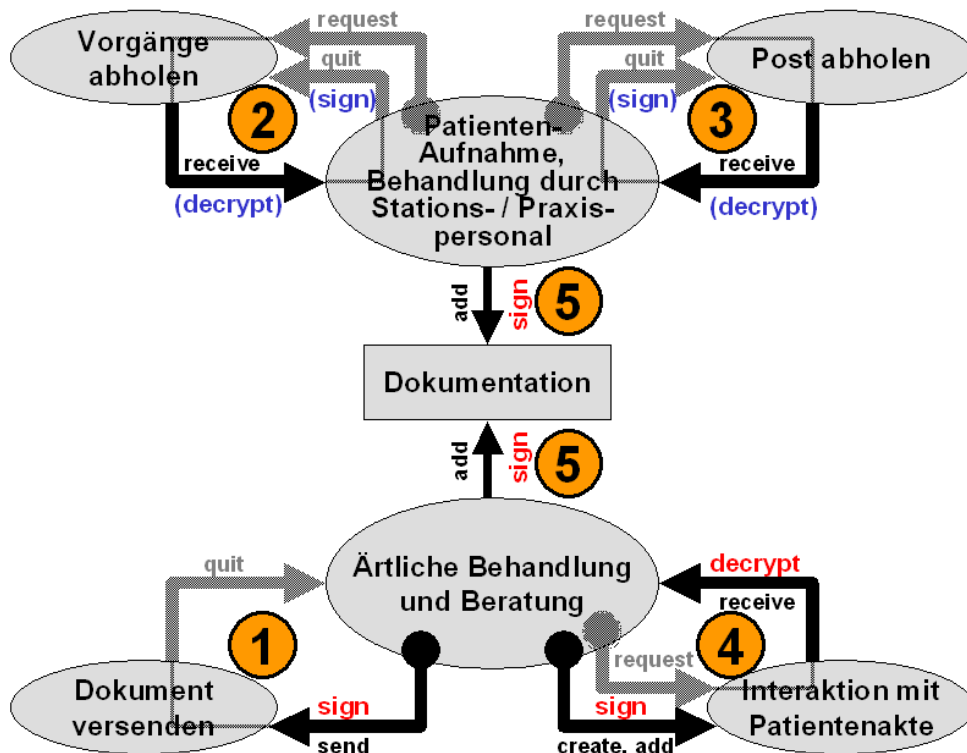


Abbildung 4: Abläufe bei medizinischen Anwendungen

Bei der Planung und Konzeption des medizinischen Einsatzszenarios sind weitere Rahmenbedingungen zu beachten. Je nach Kommunikationsmedium zwischen TPS und dem Computer ergeben sich aufgrund der räumlichen Verhältnisse spezielle Anforderungen. Typischerweise ist eine Praxis eines niedergelassenen Arztes (oder gar einer Gemeinschaftspraxis) von einer nicht unerheblichen Ausdehnung, besteht aus mehreren Behandlungszimmern, der Aufnahme und weiteren Funktionsräumen. Alle diese Bereiche sind, falls ein EDV-System eingesetzt wird, mit Computern ausgestattet und vernetzt. Bei der Ausstattung dieser IT wird für den Einsatz des TPS von Windows-Systemen oder vergleichbaren Systemen ausgegangen (Mac OS, Unix mit graphischer Oberfläche), deren

Leistungsreserven ausreichen, um kryptographische Operationen durchzuführen, eine Darstellungskomponente zu betreiben und mit den anderen Rechnern zu kommunizieren.

Für den TPS bedeutet dies, dass abhängig vom Kommunikationskanal (z.B. Bluetooth oder Infrarot) in einer solchen Praxis ggfs. kein zentraler Access-Point eingerichtet werden kann. Dies bedingt, dass zusammenhängende Informationen für einen spezifischen TPS zwischen den vernetzten Rechnern ausgetauscht werden und über dezentrale Access-Points zugänglich gemacht werden muss. Dieses in der Praxis niedergelassener Ärzte auftretende Problem ist im stationären Bereich noch wesentlich ausgeprägter. Durch die größere Ausdehnung der Stationen, die sich teilweise über mehrere Stockwerke verteilen, ist die Problematik dieser Situation noch wesentlich ausgeprägter.

Wesentlich für die technische Realisierung ist die Anbindung des TPS an die PC-Komponenten. Dabei ist die Geschwindigkeit des Kontaktaufbaus bis zur Identifizierung des TPS durch die PC-Komponente ein entscheidender Parameter. Dauert diese Identifizierung zu lange, so wird die Nutzerakzeptanz nicht erreicht. Jeder Entschlüsselungs- oder Signaturvorgang dauert dann zu lange. Dauert das Ausstellen eines Rezepts und das Aufbringen der Signatur z.B. länger als 5-10 Sekunden, dann ist ein solcher Zeitverzug in einer gut laufenden Allgemeinarztpraxis im allgemeinen nicht akzeptabel.

Für den Einsatz im medizinischen Bereich ist es sinnvoll, den TPS bzw. die zugehörigen SW-Komponenten über das Kommunikationsmodul PaDok[®] anzubinden. Dieses wird von den gängigen Praxis Computer Systemen (PCS) und Klinik Informations Systemen (KIS) unterstützt. Damit ist die Schnittstelle aus Sicht des TPS – unabhängig vom konkret anzutreffenden Praxisverwaltungs- oder Krankenhaus-System – immer die PaDok-Schnittstelle. Diese wird im Rahmen des TruPoSign-Projektes so modifiziert, dass sie den Anforderungen einer mobilen Signaturkomponente im Sinne des TPS gerecht wird. Die für die Nutzung des TPS benötigten Informationen sind bereits jetzt Bestandteil der PaDok-Schnittstelle zu den Praxis- bzw. Krankenhaus-Systemen.

Der TPS erbringt mindestens zwei Funktionen in seiner speziellen Eigenschaft als sicheres Signaturwerkzeug im medizinischen Bereich :

- Signatur von elektronischen Dokumenten wie z.B. Arztbriefen oder Rezepten
- Entschlüsselung von (verschlüsselten) Daten, z.B. von einem Kollegen elektronisch (per E-Mail) übermittelte Patienteninformationen

Beide Funktionen werden im direktem Zusammenwirken mit der Softwarekomponente auf einer PC-Arbeitsstation durchgeführt.

Die Software-Komponente auf der aktuellen PC-Arbeitsstation hat zusätzlich die Aufgabe der Synchronisation der aktuellen Arbeitsstation mit den Access-Points bzw. Übernahme der Aufgaben zur Verwaltung und Bedienung der aktuell verfügbaren TPS zu erfüllen.

Ein TPS wird auf jedem Rechner, der innerhalb des Netzwerks als Arbeitsstation in Frage kommt, genutzt werden können. Einerseits ist eine Stelle erforderlich, die alle Arbeitsstationen über die Verfügbarkeit von mobilen TPS informiert, andererseits können so Anforderungen an die einzelnen TPS über eine Art „Dispatcher“ weitergeleitet werden. Insbesondere im Fall von personen-bezogenen Anforderungen („Dr. X zur Unterschrift!“) wird damit ein komfortabler Ablauf möglich:

Auf dem TPS eines Arztes wird ein Signaturwunsch signalisiert. Außerdem wird auf dem TPS eine Information angezeigt, welche PC-Stationen aktuell zur Verfügung stehen und zeigt diese als Icon an. Parallel dazu werden auf allen Arbeitsstationen in der Windows-Fußzeile Icons für alle verfügbaren „Signer“ bzw. deren eingeschaltete TPS angezeigt. Dr. X hat jetzt zwei Möglichkeiten: Entweder er begibt sich zur nächstbesten Arbeitsstation und klickt dort auf das Icon „seines“ TPS oder er tippt auf seinem TPS auf das Icon des Arbeitsplatzes, den er aufsuchen möchte. In beiden Fällen wird die Synchronisation zwischen dem Arbeitsplatz und dem TPS hergestellt und die Dokumente für den TPS bereitstellt. Damit kann das Dokument visualisiert, der Hashwert und der Abstract View erzeugt, der Hash zum TPS übertragen werden usw.. Nach erfolgter Signatur wird die Information an die ursprünglich anfordernde Arbeitsstation gesendet.

Damit kann beispielsweise ein Rezept, das vom Patienten an der Rezeption bestellt wird, von der Sprechstundenhilfe ausgefüllt und dem Arzt „zur Unterschrift vorgelegt“ werden.

Die im Rahmen der ersten Erprobungen zu visualisierenden Dokumente werden auf XML beruhen. Es ist davon auszugehen, dass die Entwicklung neuer Daten- und Datenaustausch-Standards künftig generell auf XML basieren wird bzw. die Systeme zumindest XML-basierte Dokumente unterstützen werden. Für die Visualisierung der zu signierenden Dokumente wird daher ein XML-Viewer implementiert.

3.2 Nutzung des TPS mit der ZKA-Chipkarte

Mit **ZKA-Chipkarte** wird eine Chipkarte bezeichnet, deren Struktur der Daten, Sicherheitsarchitektur sowie Standard- und Administrationskommandos den Spezifikationen in [3] entsprechen. Eine solche ZKA-Chipkarte verfügt über das in [3] spezifizierte Betriebssystem **Secure Chip Card Operating System (SECCOS)** und unterstützt asymmetrische kryptographische Verfahren, zur Zeit auf Basis des RSA-Algorithmus.

In [5] werden die Dateien und Daten einer Applikation beschrieben, die dem Erzeugen digitaler Signaturen dient. Diese Applikation wird als **Signatur-Anwendung der ZKA-Chipkarte** bezeichnet. Für die Signatur-Anwendung der ZKA-Chipkarte sind keine Ergänzungskommandos zu definieren, da die Erzeugung digitaler Signaturen mittels der Standardkommandos des Betriebssystems erfolgt.

3.2.1 Funktionalität

Für die Signatur-Anwendung der ZKA-Chipkarte werden Sicherheitsdienste, Schlüssel und Verfahren sowie PIN und Passwörter bereitgestellt. An Sicherheitsdiensten bietet die ZKA-Chipkarte

- die Erzeugung (gesetzeskonformer) digitaler Signaturen,
- die (Client/Server-) Authentikation und
- das Entschlüsseln

an. Weiterhin werden Mechanismen zur Benutzerauthentikation sowie eine Komponenten-Authentikation zwischen ZKA-Chipkarte und einem Geschäftsterminal-Sicherheitsmodul unterstützt.

3.2.2 Sicherheitsdienste

Zur **Erzeugung digitaler Signaturen (DS)** für den **Karteninhaber (CH)** sind in der Signatur-Anwendung der ZKA-Chipkarte ein karteninhaber-spezifischer privater Signaturschlüssel $S_{K.CH.DS}$ und zugehörige Zertifikate gespeichert. Die Signatur-Anwendung der ZKA-Chipkarte unterstützt außerdem die Sicherheitsdienste

- **(Client-Server-)Authentikation ($AUT_{C/S}$)** für den Karteninhaber CH mit einem karteninhaber-spezifischen privaten RSA-Schlüssel (**CSA-Schlüssel**) $S_{K.CH.AUT_{C/S}}$,
- **Entschlüsselung von Daten (KE)**, i.d.R. symmetrische Sessionkeys, für den Karteninhaber CH mit einem karteninhaber-spezifischen privaten RSA-Schlüssel (**KE-Schlüssel**) $S_{K.CH.KE}$.

CSA-Schlüssel und KE-Schlüssel können identisch oder verschieden sein. Die Bezeichnung für einen kombinierten **CSA-KE-Schlüssel** ist $S_{K.CH.AUT_{C/S}\&KE}$. In diesem Fall muss der öffentliche Exponent des kombinierten Schlüssels F_4 sein. Ggf. kann das Zusammenspiel mit Standardanwendungen den Einsatz von verschiedenen Schlüsselpaaren beeinflussen (bspw. Netscape Communicator Unterstützung von nur einem einzelner Schlüsselpaar).

Bei der Entschlüsselung mit dem privaten Schlüssel $S_{K.CH.KE}$ gilt es zu beachten, dass bei einer Public-Key-Verschlüsselung die Daten üblicherweise zuerst mit einem symmetrischen Sessionkey verschlüsselt werden. Erst danach wird dieser Schlüssel mit dem öffentlichen Schlüssel des Empfängers gesichert. Zu diesem Zweck kann ein eigenes Schlüsselpaar verwendet werden, das sich von dem Signaturschlüsselpaar unterscheidet. Zur Entschlüsselung wird dann wieder der Schlüssel $S_{K.CH.KE}$ benötigt. Die symmetrische Ver- und Entschlüsselung von Nutzdaten wird durch die ZKA-Chipkarte nicht unterstützt.

Der oder die benötigte(n) Schlüssel ist (sind) in der Signatur-Anwendung der ZKA-Chipkarte gespeichert. Die zugehörigen Zertifikate (**CSA-Zertifikate, KE-Zertifikate oder CSA-KE-Zertifikate**) sind, wenn vorhanden, ebenfalls in der Signatur-Anwendung der ZKA-Chipkarte gespeichert.

Im MF einer ZKA-Chipkarte mit Signatur-Anwendung sind weitere asymmetrische Schlüssel und zugehörige Zertifikate gespeichert. Diese sind nicht karteninhaber- sondern komponenten-spezifisch. Sie dienen der **Komponenten-Authentikation** zwischen der Chipkarte (**ICC**) und den Sicherheitsmodulen von Terminals (**IFD**) mit Aushandlung von Sessionkeys. Eine Komponenten-Authentikation ist für den TPS jedoch nicht sinnvoll anwendbar. Die Rolle des Geschäftsterminals käme dem Einsatz eines TPS im Unknown User Mode gleich, der jedoch über kein Sicherheitsmodul verfügt.

Die **Benutzerauthentikation** zwischen Karteninhaber und ZKA-Chipkarte basiert auf der Verwendung von Geheimnissen wie Signatur-PIN bzw. CSA-Passwort. Der Schlüssel $S_{K.CH.DS}$ wird durch die numerische Signatur-PIN abgesichert. Dabei lässt sich aber die Anzahl der möglichen Signaturerstellung nach einer erfolgreichen Benutzerauthentikation

konfigurieren¹. Erst nach korrekter Eingabe einer 6-stelligen Signatur-PIN ist das Erzeugen von Signaturen mit $S_{K.CH.DS}$ möglich.

Der Schlüssel $S_{K.CH.AUT}$ wird mit dem alphanumerischen CSA-Passwort (Client/Server-Authentikation) geschützt. Dieses kann - je nach Entscheidung des Kartenherausgebers - auch für die Absicherung des Schlüssels $S_{K.CH.KE}$ verwendet werden. Andernfalls kann das Kartenprofil auch vorsehen, dass der Schlüssel $S_{K.CH.KE}$ ohne Zugriffsschutz genutzt werden kann. In beiden Fällen gilt für das CSA-Passwort, dass es nur einmal für eine Session verifiziert werden muss. Sowohl Signatur-PIN als auch CSA-Passwort können jeweils durch den Benutzer geändert werden.

Für die Sicherheitsdienste der ZKA-Chipkarte werden drei Schlüsseltypen zur Verfügung gestellt. Diese werden wie folgt unterschieden:

Sicherheitsdienst	Schlüssel	PIN/Passwort
Digitale Signatur	$S_{K.CH.DS}$	Signatur-PIN
Client/Server-Authentikation	$S_{K.CH.AUT}$	CSA-Passwort
Entschlüsselung	$S_{K.CH.KE}$	CSA-Passwort (optional)

Wenn die Signatur-PIN durch zu viele falsche Eingaben blockiert ist, kann die Signatur-PIN neu gesetzt werden. Hierzu ist ein „Resetting Code“ (PUK) anzugeben. Die ZKA-Chipkarte kann bis zu 6 Resetting Codes verwalten, die als Einmal Resetting Code verwendet werden können. D. h. jeder Resetting Code ist genau einmal verwendbar. Bei der Bedienung ist zur Identifizierung des Resetting Codes eine Nummer einzugeben, die dann den Resetting Code identifiziert.

Die Verwendung von Resetting Codes in der ZKA-Chipkarte ist optional – der Kartenherausgeber legt fest, ob diese personalisiert werden oder nicht.

3.3 Der elektronische Scheck – eine kreditwirtschaftliche Formularanwendung

In diesem Kapitel wird zuerst der allgemeine Aufbau und Ablauf von kreditwirtschaftlichen Formularanwendungen dargestellt. Nach diesem kurzen Einblick werden an dem Beispiel des eSchecks die Grundzüge erläutert. Abgeschlossen wird das Kapitel von einer Auflistung der Anforderungen an den TPS.

Allgemeiner Aufbau und Ablauf

Das Ziel einer kreditwirtschaftlichen Formularanwendung ist es, dem Kunden die Funktionalität eines (ihm meistens bekannten) Formulars auch online zur Verfügung zu stellen. Die dazu erforderliche Anwendung kann dem Kunden über eine Webschnittstelle in

¹ Diese Fähigkeit kommt dem TPS-Grundprinzip sehr entgegen.

einem Browser präsentiert werden. Das Formular wird somit im Browser dargestellt und der Kunde kann die entsprechenden Eintragungen vornehmen. Für die Authentisierung und Zuweisbarkeit des Vorgangs wird das Dokument elektronisch signiert. Für das Kreditinstitut steht hierbei insbesondere die Nichtabstreitbarkeit der Aktion durch den Kunden im Vordergrund.

Für den Einsatz des TPS bei einer schon bestehenden Anwendung ist es notwendig, dass für den Signiervorgang auf dem TPS ein Eingriff in den Anwendungsablauf erfolgt, da die Nutzdaten für die Berechnung des Hashwertes und der Signatur transferiert werden müssen. Der erforderliche Eingriff bezieht sich dabei aber nur auf den Ablauf der Signaturerstellung. Alle weiteren Ablaufschritte (bspw. Kommunikation mit einem Hintergrundsystem) bleiben davon unberücksichtigt. Die auf dem TPS erstellte elektronische Signatur muss an die Anwendung übergeben und dort entsprechend eingebunden werden. Nach Abschluss der Signaturerstellung kann der „normale“ Ablauf wieder fortgesetzt werden.

Ein mögliches Szenario aus der Kreditwirtschaft für eine Signaturanwendung ist der elektronische Scheck (kurz: eScheck). Dabei handelt es sich um ein sicheres Zahlungsmittel mit Zahlungsgarantie im Internet, welches auf elektronischen Signaturen basierend für Zahlungen von *Privat an Privat* und von *Privat an Händler* genutzt werden kann. Der eScheck stellt eine elektronische Nachbildung des bis Ende 2001 üblichen Eurocheques dar.

Bei dem Einsatz des eSchecks benötigt nur der Aussteller eine entsprechende Kundenumgebung, die neben einem Computer mit Internetzugang und einem Chipkartenterminal auch eine Banken-Signaturkarte mit Zertifikat beinhaltet. Damit nach der Ausstellung des Schecks keine Manipulation mehr möglich ist, wird er sowohl vom Aussteller als auch von der Scheckautorisierungszentrale (**SAZ**) elektronisch signiert. Der Empfänger erhält den Scheck beispielsweise per Email und benötigt für die Autorisierung und Einreichung bei der Scheckeinreichungszentrale (**SEZ**) nur einen Rechner mit Internetzugang.

Zur Zeit sind nur eSchecks möglich, die (im Sinne eines Verrechnungsschecks) auf ein im eScheck anzugebendes Konto eingezahlt werden können. Der Aussteller eines eSchecks benötigt daher die Daten der Kontoverbindung des Empfängers. Die folgende Abbildung gibt einen Überblick über den Informationsfluss bei dem Ausstellen eines eSchecks mit Zahlungsgarantie zwischen Privatpersonen.

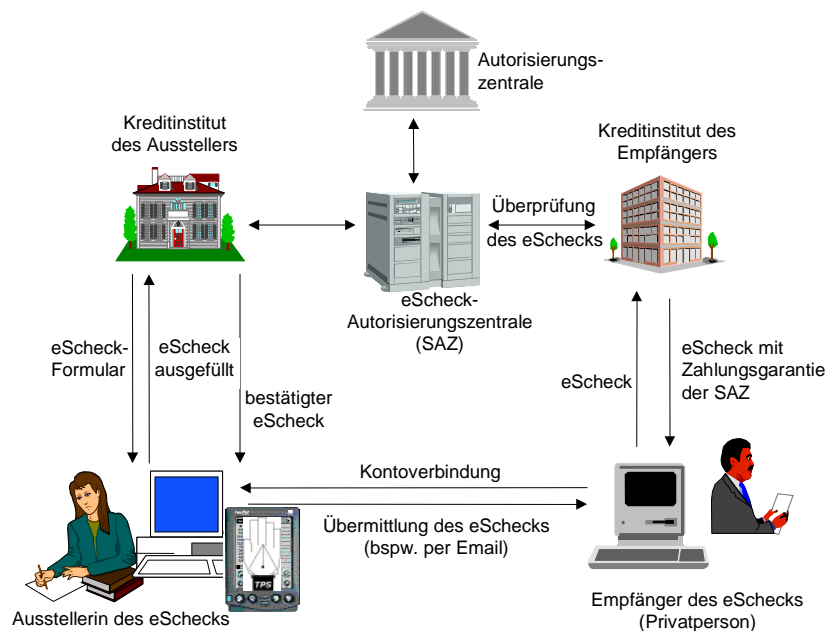


Abbildung 4: Ausstellen eines eSchecks mit Zahlungsgarantie

Die (technische) Abwicklung für das Ausstellen eines eSchecks und für die Überprüfung der Zahlungsgarantie wird durch eine Scheckautorisierungszentrale (SAZ) durchgeführt, die durch den Bank-Verlag betrieben wird. Für die Autorisierung eines eScheck (Ausprechen der Zahlungsgarantie) greift diese auf die bereits vorhandene Autorisierungszentrale der Kreditwirtschaft zurück. Eingereicht wird ein eScheck über eine Scheckeinreichungszentrale (SEZ, in der Abbildung nicht dargestellt), die heute ebenfalls vom Bank-Verlag betrieben wird. Aussteller und Empfänger eines eSchecks greifen im Allgemeinen über Verbindungen zu ihren Kreditinstituten auf die Dienstleistungen von SAZ und SEZ zu, beide sind aber auch direkt über das Internet erreichbar.

Will ein Kunde einen eScheck ausstellen, wendet er sich (über sein Kreditinstitut) an die SAZ. Für die Verbindung zur SAZ wird eine SSL-Verbindung mit Client-Authentikation aufgebaut. Von der SAZ erhält der Aussteller ein eScheck-Formular, in dem bereits sein Name und seine Kontoverbindung (verschlüsselt) sowie eine Nummer des eSchecks enthalten sind. Der Aussteller füllt das Formular mit den Daten des Empfängers, dem Betrag und dem Verwendungszweck aus. Er signiert das Formular dann mit seiner ZKA-Chipkarte, wobei eine (gesetzeskonforme) elektronische Unterschrift erzeugt wird. Ausgefülltes Formular und Signatur werden wieder an die SAZ gesendet, die die Daten des Ausstellers (soweit möglich) überprüft. Über eine Anfrage an die Autorisierungszentrale wird die Möglichkeit zur Zahlung überprüft. Die SAZ unterschreibt den eScheck, wodurch die Gültigkeit des eSchecks bestätigt wird.

Der Aussteller erhält den eScheck von der SAZ zurück. Der eScheck wird dabei als XML-Datei an den Aussteller übermittelt, die durch geeignete Browser angezeigt werden kann. Diese Datei kann der Aussteller z.B. als Anhang für eine E-Mail an den Empfänger übermitteln.

Empfängt eine Person einen eScheck (in Form einer XML-Datei) von einem Aussteller, so kann er diesen zur Überprüfung an die SAZ geben. Die SAZ überprüft die elektronischen Unterschriften des eScheck. Durch eine erneute (elektronische) Unterschrift der SAZ bestätigt sie die Korrektheit des eSchecks und damit insbesondere auch die Zahlungsgarantie. Das Ergebnis (eScheck mit erneuter Unterschrift der SAZ) wird wieder als XML-Datei an den Empfänger übermittelt.

Der Empfänger kann den eScheck zu einem späteren Zeitpunkt (über sein Kreditinstitut) bei der SEZ einreichen. Ein eScheck (und damit auch die Zahlungsgarantie) hat eine festgelegte Gültigkeitsdauer. Der Empfänger kann daher den eScheck auch einfach verfallen lassen, was zum Beispiel dann sinnvoll sein kann, falls der eScheck als Kautions für die Miete eines Objekts ausgestellt hatte.

Im Rahmen der Anwendung eScheck sind weitere Funktionalitäten realisiert. So kann ein Aussteller einen eScheck sperren lassen und ein Empfänger kann einen eScheck "elektronisch zerreißen". Die Abläufe bei der Ausstellung eines eSchecks von "Privat an Händler" sind ähnlich wie der beschriebene Fall. Hierauf soll jedoch nicht weiter eingegangen werden. Im Anhang (Kapitel 5) wird eine technische Beschreibung der Abläufe zwischen Anwenderprogramm und TPS zur Ausstellung eines eSchecks dargestellt.

4 Literatur

- [1] Cremers, Spalka, Langweg, Vermeidung und Abwehr von Angriffen Trojanischer Pferde auf Digitale Signaturen, 7. Deutscher IT Sicherheitskongress, 14.-16.Mai 2001, Bonn
- [2] Gesetz über die Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften, 16.06.2002, Bundesgesetzblatt Jahrgang 2001 Teil I Nr.22, ausgegeben zu Bonn am 21.Mai 2001
- [3] Schnittstellenspezifikation für die ZKA-Chipkarte, Secure Chip Card Operating System (SECCOS), Version 5.0, 05.06.2001 mit Errata vom 13.06.2001
- [4] Schnittstellenspezifikation für die ZKA-Chipkarte, Signatur-Anwendung, Version 1.0, 14.09.2001
- [5] Schnittstellenspezifikation für die ZKA-Chipkarte, Spezifikation des Internet-Kundenterminals für die Unterstützung der Signatur-Anwendung der ZKA-Chipkarte, Version 0.5, 30.04.2002
- [6] DIN-Spezifikation der Schnittstelle zu Chipkarten mit Digitaler Signatur-Anwendung/Funktion nach SigG und SigV, DIN NI-17.4, Version 1.0, 15.12.1998

5 Anhang: Technische Beschreibung der Abläufe zwischen Anwenderprogramm und TPS zur Ausstellung eines eSchecks

In diesem Abschnitt werden die zur Ausstellung eines eSchecks erforderlichen Abläufe in der Kundenumgebung (unter Verwendung eines TPS und Einsatz eines Plugins unter Netscape) beschrieben. Dabei werden nur diejenigen Abläufe betrachtet, die eine Beteiligung der ZKA-

Chipkarte voraussetzen. Diese umfassen den Aufbau einer SSL-Verbindung mit Client-Authentikation sowie die Erzeugung der digitalen Signatur mit dem Signaturschlüssel (DS) der ZKA-Chipkarte.

Betrachtet werden die Anwendungsschnittstelle PKCS #11 und die ZKA-SIG-API bzw. TPS-SIGN-API auf dem PC sowie der TPS. Die erforderlichen Funktionen zur Ermittlung von Zertifikaten und Schlüssel bzw. zur Initialisierung des Tokens sind hier nicht dargestellt.²

Für die Client-Authentikation wird durch Netscape die Funktion C_Sign aufgerufen. In diesem Fall erfolgt das Hashen vorab durch den PKCS #11-Token ohne Einbezug des TPS.

² Es werden verwendet: C - Command, A - Action und R - Response

PKCS #11		ZKA-SIG-API		TPS			
C1	C_SignInit () Methode: RSA_PKCS Key: Handle auf Objekt zu S _K .CH.AUT _{C/S}						
A1	Speichern des Kontextes						
R1	-						
C2	C_Sign () Data: Hashwert Daten: Länge des Hashwerts						
A2	Ermittlung des Kontextes						
		--->	C3	C_cs_authentication Länge des Hashwerts Hashwert	A3	Eingangsprüfungen: Modulus-Länge von S _K .CH.AUT _{C/S} Länge der Daten darf höchstens 40% der Moduluslänge betragen	
					--->	C4	TPS_cs_authentication Länge des Hashwerts, Hashwert
					A4	Aufruf des SECCOS-Kommandos SET (für interne Authentikation). Der Aufbau des Kommandos ist dem SSD-File zu entnehmen.	
					A5	Aufruf des Kommandos INTERNAL AUTHENTICATE mit Länge des Hashwerts und Hashwert	
					A6	ggf. Benutzer-Authentikation mit CSA-Passwort und Wiederholung von A5	
					<---	R4	Signatur, Länge der Signatur
R2	Signatur, Länge der Signatur	<---	R3	Signatur, Länge der Signatur			

Für die Beschreibung der Abläufe zur Erzeugung der digitalen Signatur wird von einem idealtypischen Ablauf ausgegangen, d. h. an der PKCS #11 Schnittstelle wird die Funktion C_Sign mit Übergabe der Nutzdaten und dem Mechanismus „SHA1-RSA-PKCS“ aufgerufen. Die entsprechenden PKCS #11 Aufrufe werden durch einen Browser PlugIn gesteuert.

