(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 602 00 081 T2** 2004.04.22

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 349 031 B1**
(21) Deutsches Aktenzeichen: **602 00 081.5**
(96) Europäisches Aktenzeichen: **02 006 514.0**
(96) Europäischer Anmeldetag: **18.03.2002**
(97) Erstveröffentlichung durch das EPA: **01.10.2003**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **05.11.2003**
(47) Veröffentlichungstag im Patentblatt: **22.04.2004**

(51) Int Cl.[7]: **G06F 1/00**

(73) Patentinhaber:
UBS AG, Zürich, CH

(74) Vertreter:
WUESTHOFF & WUESTHOFF Patent- und Rechtsanwälte, 81541 München

(84) Benannte Vertragsstaaten:
AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LI, LU, MC, NL, PT, SE, TR

(72) Erfinder:
Hiltgen, Dr., Alain P., 8057 Zürich, CH

(54) Bezeichnung: **Sichere Benutzer- und Datenauthenifizierung über ein Kommunikationsnetzwerk**

Beschreibung

Technisches Gebiet

[0001] Die Erfindung betrifft das Gebiet der Netzwerksicherheit. Insbesondere betrifft die Erfindung ein Verfahren, eine Server-Infrastruktur und ein Netzwerksystem, welche die sichere Benutzer- und Datenauthentisierung mittels eines Netzwerk-Client ermöglichen, der über einen Kartenleser Zugriff auf eine intelligente Karte (Smart Card/Chipkarte) hat.

Beschreibung des Standes der Technik

[0002] In den letzten Jahren wurde eine zunehmende Anzahl neuartiger Anwendungen wie sichere Zahlungsdienste und sichere Authentisierungsdienste als kartenbasierte Anwendungen verwirklicht. Heutzutage findet ein Übergang von Karten mit Magnetstreifen zur Technologie mit intelligenten Karten statt, die auch als integrierte Schaltungs- (integrated circuit – IC) oder Chipkarten-Technologie bezeichnet wird. So sind beispielsweise nahezu die Hälfte aller derzeit in Europa sich im Umlauf befindenden Bankkarten bereits chipbasiert und der Prozentsatz der chipbasierten Bankkarten nimmt ständig zu.

[0003] Die Branche nutzt die zusätzliche von intelligenten Karten gebotene Sicherheit, um eine kompatible sichere Infrastruktur für Heimgeräte zur Verfügung zu stellen. Durch Verwendung intelligenter Karten in der häuslichen Umgebung können den Verbrauchern sichere Zahlungs- und Authentisierungsdienste angeboten werden, wodurch Ferndienste wie E-Commerce gefördert werden. Wie das Gebiet des E-Commerce, erfordern auch weitere Bereiche wie das Home-Banking, Sicherheitsdienste sowie E-Behörden die Anwendung einer sicheren und vertrauenswürdigen Infrastruktur intelligenter Karten.

[0004] Eine derartige Chipkarten-Infrastruktur weist notwendigerweise eine Chipkarte auf, auf der ein Signaturschlüssel gespeichert ist, zusammen mit einem sicheren intelligenten Kartenleser wie dem in der Workshop-Vereinbarung CWA 14174 des europäischen Normungsausschusses (CEN) spezifizierten Kartenleser. Ein Hauptziel dieser FINREAD- (FINancial transactional IC card READer) Initiative ist die Spezifikation eines intelligenten Kartenlesers, der Sicherheit für zahlreiche verschiedene Typen von Anwendungen bietet. Folglich unterstützt der FINREAD-Kartenleser nicht nur von Banken ausgegebene intelligente Karten, sondern auch Chipkarten, die für andere als finanzielle Anwendungen ausgegeben werden.

[0005] In Anbetracht der Tatsache, dass ein Personal Computer ein Ziel für Angriffe durch Viren und „Trojanische Pferde" darstellt, stellt der FINREAD-Kartenleser eine zusätzliche Sicherheitsebene bereit, um den Personal Computer oder ein anderes Verbraucherzugriffsgerät zum Bestand einer sicheren und zuverlässigen Umgebung zu machen. Jegliche Verarbeitung innerhalb eines bestimmten Schemas, das eine zuverlässige Abwicklung betrifft, erfolgt nur über den FINREAD-Kartenleser. Dies stellt sicher, dass jede erforderliche Information vom Verbraucher authentisch bestätigt werden kann.

[0006] Die Authentisierung des FINREAD-Kartenlesers ist in Kapitel **10** der CEN-Workshop-Vereinbarung „Financial transactional IC card reader (FINREAD) – Teil **2**: Functional regirements" (Ref. Nr. CWA 14174-2: 2001 E) vom Juli 2001 spezifiziert. Das Hauptziel der Authentisierungsfunktion des FINREAD-Kartenlesers ist es, einem Dienstanbieter wie einem Finanzinstitut oder einem Zahlungssystem zu ermöglichen, den Ursprung der von einem FINREAD-Kartenleser gesendeten Daten zu authentisieren. Diese Funktion bietet Schutz gegen einen gefälschten Kartenleser, der Daten als FINREAD-Einheit schickt, und außerdem gegen Verleugnung, dass eine authentisierte Meldung mit einem FINREAD-Kartenleser geschickt wurde. Die Authentisierungsfunktion des FINREAD-Kartenlesers basiert auf einer eindeutigen Identifikationsnummer, die jeder FINREAD-Kartenleser zusätzlich zu seiner Fähigkeit, mit einem eindeutigen privaten Schlüssel zu signieren, besitzt. Der private Schlüssel ist in einem manipulationsgeschützten Sicherheitsmodul des FINREAD-Kartenlesers gespeichert, das sämtliche vertraulichen Informationen in einer sicheren Umgebung hält.

[0007] Ein Hauptmerkmal des FINREAD-Kartenlesers ist eine Benutzerschnittstelle, die in Kapitel 7 der obigen CEN-Workshop-Vereinbarung beschrieben wird. Die Benutzerschnittstelle weist eine Anzeige auf, die zur Information des Kartenhalters dient. Da der Kartenhalter auf Basis der angezeigten Information entscheidet, ob angezeigte Daten zur Signaturerzeugung an die Chipkarte geschickt werden oder nicht, muss der FINREAD-Kartenleser sicherstellen, dass alle Informationen auf der Anzeige zuverlässig sind. Um die Anzeige des FINREAD-Kartenlesers gegen externe Angriffe zu schützen, verhindert der FINREAD-Kartenleser, dass Benutzergeräte wie ein Personal Computer Informationen direkt an die Anzeige schicken.

[0008] Außer der Anzeige weist die Benutzerschnittstelle einen Tastenblock auf. Der Tastenblock ermöglicht dem Benutzer die Kommunikation mit dem FINREAD-Kartenleser und insbesondere die Eingabe von Informationen, die von Anwendungen, die beispielsweise auf dem Kartenleser laufen, angefordert werden. Der Tastenblock ist faktisch die einzige Möglichkeit, Daten auf zuverlässige Weise einzugeben.

[0009] Gemäß „Financial transactional IC card reader (FINREAD) – Teil **3**: Security requirements" (Ref. Nr. CWA 14174-3: 2001 E) vom Juli 2001, Kap. 6.3, ist die Authentisierung des FINREAD-Kartenlesers kryptographisch mit einer bestimmten Transaktion verknüpft und wird während der Transaktion aktiviert, wenn die Authentisierungsfunktionalität benötigt

wird. Während der Authentisierung des FINREAD-Kartenlesers wird eine digitale Signatur mit dem privaten Schlüssel des Kartenlesers berechnet. Genauer gesagt werden zu signierende Daten dem Sicherheitsmodul des FINREAD-Kartenlesers zur Signaturberechnung mit dem privaten Schlüssel bereitgestellt. Um eine einheitliche Authentisierungsfunktion zu ermöglichen, ist die eindeutige Identifikationsnummer ebenfalls in den signierten Daten enthalten.

[0010] Ausgehend von Anwendungen wie E-Commerce, E-Banking oder E-Behörden, die die Verwendung eines sicheren und vertrauenswürdigen Chipkartenlesers wie des FINREAD-Kartenlesers oder eines anderen Kartenlesers erforderlich machen, besteht ein Bedarf an einem sicheren Verfahren zur Benutzer- und Datenauthentisierung. Genauer gesagt besteht ein Bedarf an einem Verfahren, einem Computerprogrammprodukt, einer Serverinfrastruktur und einem Netzwerksystem, welche es gestatten, die Benutzer- und Datenauthentisierung auf einer höheren Sicherheitsebene unter Verwendung eines Kartenlesers zusammen mit einer entsprechenden Chipkarte durchzuführen.

ZUSAMMENFASSUNG DER ERFINDUNG

[0011] Bezüglich eines Verfahrens wird diesem Bedarf mittels der Durchführung der Benutzer- und Datenauthentisierung über einen Client in Kommunikation über ein erstes Netzwerk mit einer Server-Infrastruktur, vorzugsweise mit einem Anwendungsserver, nachgekommen, wobei der Client über einen seitens des Benutzers steuerbaren Kartenleser Zugriff auf eine Chipkarte hat, auf der ein oder mehrere Schlüssel gespeichert sind. Das Verfahren weist einen Schritt der Benutzerauthentisierung auf, der die Anzeige eines Challenge in einem Authentisierungskontext durch den Kartenleser enthält, das Steuern des Kartenlesers, um den Benutzer zur Genehmigung einer Signatur aufzufordern, und, im Falle der Genehmigung der Signatur, das Übergeben des Challenge, sofern zutreffend zusammen mit den Kontextdaten, oder davon abgeleiteter Daten an die Chipkarte zum Signieren. Das Verfahren weist des Weiteren einen Schritt der Datenauthentisierung auf, der beinhaltet, dass die zu authentisierenden Daten vom Kartenleser angezeigt werden, der Kartenleser gesteuert wird, um vom Benutzer die Genehmigung einer Signatur anzufordern, und, im Falle der Genehmigung der Signatur, die zu authentisierenden Daten oder Daten wie ein davon abgeleiteter Hash-Wert an die Chipkarte zum Signieren übergeben werden. Mit anderen Worten werden in einem Netzwerksystem mit einer Client-Infrastruktur und einer Server-Infrastruktur getrennte Benutzer- und Datenauthentisierungsschritte durchgeführt.

[0012] Für die Schritte der Benutzer- und Datenauthentisierung kann derselbe Signaturschlüssel der Chipkarte oder es können verschiedene Signaturschlüssel der Chipkarte verwendet werden. Wird die Chipkarte mit nur einem darauf abgespeicherten Signaturschlüssel ausgeliefert, gibt es keine andere Wahl als beide Authentisierungsschritte mit diesem einzigen Signaturschlüssel durchzuführen.

[0013] Ein und derselbe Kartenleser kann so konfiguriert sein, dass er sowohl mit Chipkarten arbeiten kann, die die Durchführung beider Authentisierungsschritte mit verschiedenen Signaturschlüsseln ermöglichen, als auch mit Chipkarten, die die Verwendung ein und desselben Signaturschlüssels für beide Authentisierungsschritte erfordern. In einem solchen Fall kann der Kartenleser mit einer proprietären Anwendung ausgestattet sein, die das Verhalten des Kartenlesers entsprechend konfiguriert. Die Anwendung kann also den Zugriff des Kartenlesers auf die Chipkarte in dem Fall steuern, in dem eine Signatur der Chipkarte mit einem bestimmten von mehreren Signaturschlüsseln der Chipkarte erforderlich ist. Im Rahmen dieser Erfindung kann der Begriff „signieren" einen Schritt beinhalten, der aus den zu signierenden Daten Hash-Daten macht. Die Hash-Funktion kann von derselben Komponente, die den Schlüssel anwendet, ausgeführt werden oder von einer anderen Komponente.

[0014] Um den Typ der gerade verwendeten Chipkarte zu bestimmen, kann die Server-Infrastruktur bei Öffnen einer Server-Anwendung eine entsprechende Anfrage an den Chipkartenleser senden. Die empfangene Antwort bestimmt das weitere Verhalten der Anwendung, wenn eine Chipkartensignatur erforderlich ist. Falls zwei oder mehr Signaturschlüssel auf der Chipkarte gespeichert sind, kann eine Signaturanforderung der Anwendung also zusätzlich einen bestimmten Signaturschlüssel der Chipkarte angeben. Vorzugsweise interpretiert der Client solche Signaturanforderungen nicht, sondern gibt sie einfach an den Kartenleser weiter.

[0015] Um eine höhere Sicherheitsebene zu implementieren, vor allem wenn beide Authentisierungsschritte mit demselben Signaturschlüssel durchgeführt werden, kann der Kartenleser so konfiguriert werden, dass er alles anzeigt, was mit dem Signaturschlüssel zu signieren ist. In einem solchen Fall kann nichts signiert werden, das nicht vom Kartenleser angezeigt wird, und alles, was angezeigt wird, sollte signiert werden.

[0016] Enthält ein zu signierender Challenge während des Schrittes der Benutzerauthentisierung keinen lesbaren Text, sondern z. B. einen Zufallswert, kann es den Benutzer verwirren, wenn nur dieser Zufallswert vom Kartenleser angezeigt wird. Es ist deshalb sinnvoll, den zu signierenden Challenge während des Benutzerauthentisierungsschrittes in einem Authentisierungskontext anzuzeigen. Der Authentisierungskontext kann von der Server-Infrastruktur bereitgestellt oder ausgelöst werden, vorzugsweise gesteuert von der Anwendung.

[0017] Der Authentisierungskontext ist vorzugsweise eine Textmeldung, die auf die Tatsache hinweist, dass die Signatur zu Authentisierungszwecken erfor-

derlich ist. Um die Sicherheit noch weiter zu verbessern, kann während des Benutzerauthentisierungsschrittes nicht nur der Challenge signiert werden, sondern entsprechende Kontextdaten, die mit der Textmeldung identisch oder von dieser abgeleitet sein können, werden ebenfalls signiert. Die signierten Challenge- und Kontextdaten werden dann über das erste Netzwerk zurück übertragen.

[0018] Vorzugsweise wird zumindest der Schritt der Benutzerauthentisierung (und wahlweise der Schritt der Datenauthentisierung) über einen verschlüsselten Kanal durchgeführt, der vor dem oder in Zusammenhang mit dem Benutzerauthentisierungsschritt eingerichtet wird. Das Einrichten des verschlüsselten Kanals kann von der Server-Infrastruktur gesteuert werden.

[0019] Eine Abhängigkeit des Benutzerauthentisierungsschrittes, der einen zur Chipkarte gehörigen Signaturschlüssel mit einschließt, von einem Schlüssel der Server-Infrastruktur für Verschlüsselungszwecke kann eingeführt werden. Diese Abhängigkeit kann z. B. aus der Ableitung eines Authentisierungs-Challenge vom Schlüssel für Verschlüsselungszwecke und aus der Signierung dieses Challenge mit dem Signaturschlüssel der Chipkarte resultieren. Auf diese Weise kann der Authentisierungsschritt unmittelbar mit dem auf Basis des Schlüssels für Verschlüsselungszwecke errichteten verschlüsselten Kanal gekoppelt werden.

[0020] Der Schlüssel für Verschlüsselungszwecke der Server-Infrastruktur ist entweder ein symmetrischer oder ein asymmetrischer Schlüssel. Vorzugsweise wird ein asymmetrischer öffentlicher Schlüssel verwendet, um einen symmetrischen Schlüssel für den verschlüsselten Kanal zu vereinbaren.

[0021] Vorzugsweise gehört ein erster Authentisierungsschlüssel zum Kartenleser, und zwar zusätzlich zu dem mindestens einen auf der Chipkarte gespeicherten und zu Datenauthentisierungszwecken verwendeten Signaturschlüssel. Im Folgenden bezieht sich der Begriff „Authentisierungsschlüssel" auf einen Signaturschlüssel, der ausschließlich oder hauptsächlich zur Benutzerauthentisierung (einschließlich der Authentisierung der Chipkarte) oder Geräteauthentisierung (einschließlich der Authentisierung des Kartenlesers) verwendet wird.

[0022] Der zum Kartenleser gehörige erste Authentisierungsschlüssel kann verwendet werden, um die (während des Schrittes der Benutzer- und/oder Datenauthentisierung) mit dem mindestens einem Signaturschlüssel der Chipkarte erzeugte Signatur zu signieren. Die resultierende doppelte Signatur kann dann erforderlichenfalls zusammen mit dem Challenge, den Kontextdaten und/oder den zu authentisierenden Daten an die Server-Infrastruktur übertragen werden. Die doppelte Signatur kann somit sicherstellen, dass die Chipkarte in einem echten Kartenleser betrieben wird, und verknüpft die Leserauthentisierung kryptographisch mit einer auf Chipkarten basierenden Anwendung.

[0023] Im Falle von Signaturen, die eine Wiederherstellung der Meldung (message recovery) ermöglichen, wird nur die doppelte Signatur anschließend an die Server-Infrastruktur geschickt. Ist beispielsweise ein Hash-Wert normaler (lesbarer) Daten signiert worden, wird die doppelte Signatur zusammen mit den entsprechenden lesbaren Daten an die Server-Infrastruktur geschickt.

[0024] Vorzugsweise ist außer dem Signaturschlüssel ein zweiter Authentisierungsschlüssel auf der Chipkarte gespeichert. Während des Schrittes der Benutzerauthentisierung wird dieser zweite Authentisierungsschlüssel zum Signieren des Challenge und erforderlichenfalls der Kontextdaten verwendet. Andererseits wird der auf der Chipkarte gespeicherte Signaturschlüssel während des Schrittes der Datenauthentisierung verwendet, um die zu authentisierenden Daten oder Daten wie einen davon abgeleiteten Hash-Wert zu signieren.

[0025] Der Schritt der Benutzerauthentisierung kann einen oder mehrere Unterschritte enthalten. Der Benutzerauthentisierungsschritt kann beispielsweise einen ersten Benutzerauthentisierungsschritt enthalten, an dem der zweite Authentisierungsschlüssel und der Schlüssel der Server-Infrastruktur für Verschlüsselungszwecke beteiligt sind, und einen zweiten Benutzerauthentisierungsschritt, an dem mindestens der erste und der zweite Authentisierungsschlüssel beteiligt sind.

[0026] Vorzugsweise wird der erste Benutzerauthentisierungsschritt von einer Eingangsstelle der Server-Infrastruktur und der zweite Benutzerauthentisierungsschritt von einer Anwendung der Server-Infrastruktur gesteuert. Der erste Benutzerauthentisierungsschritt kann jedoch alternativ von anderen Komponenten des zweiten Netzwerks gesteuert werden, wie z. B. von einer auf einem Anwendungsserver laufenden Anwendung. Gemäß einer bevorzugten Ausführungsform steuert die Anwendung mindestens den zweiten Benutzerauthentisierungsschritt, während dessen eine authentisierte Verbindung zwischen dem Client und dem Anwendungsserver hergestellt wird. Während des ersten Benutzerauthentisierungsschrittes kann eine end-to-end Verbindung hergestellt werden.

[0027] Der erste Benutzerauthentisierungsschritt und der zweite Benutzerauthentisierungsschritt können auf derselben oder auf verschiedenen Schichten des 7-Schichtenmodells gemäß ISO-OSI durchgeführt werden. Bei einer Ausführungsform wird der zweite Benutzerauthentisierungsschritt auf der Anwendungsschicht (Schicht **7**) und der erste Benutzerauthentisierungsschritt auf einer Schicht unterhalb der Anwendungsschicht durchgeführt. Der erste Authentisierungsschritt kann z. B. auf der Transportschicht (Schicht **4**) oder auf einer Schicht zwischen der Transportschicht und der Anwendungsschicht durchgeführt werden. Der Datenauthentisierungsschritt wird vorteilhafter Weise auf der Anwendungsschicht durchgeführt.

[0028] Doppelte Signaturen, die beispielsweise in Zusammenhang mit dem Schritt der Daten- oder Benutzerauthentisierung erzeugt werden können, werden vorzugsweise nur auf der Anwendungsschicht verwendet. Im Gegensatz dazu können einzelne Signaturen für Prozesse wie die Secure Socket Layer (SSL)-Benutzerauthentisierung verwendet werden, die auf einer Schicht unterhalb der Anwendungsschicht laufen.

[0029] Vorzugsweise wird der erste Authentisierungsschritt gemäß dem SSL-Protokoll, dem Transport Layer Security (TLS)-Protokoll, dem Wireless Transport Layer Security (WTLS)-Protokoll oder jedem anderen Protokoll durchgeführt, das die Einrichtung eines verschlüsselten Kanals mit einer Authentisierungsroutine verknüpft. SSL- und ähnliche Protokolle gelten normalerweise als Teil der Transportschicht und verwenden eine Verschlüsselung mit asymmetrischem Schlüssel, um Informationen über einen symmetrischen Schlüssel auszutauschen, auf deren Basis ein verschlüsselter Kommunikationskanal eingerichtet wird. Es ist jedoch zu beachten, dass andere Authentisierungsprotokolle wie IPSec, das auf der Netzwerkschicht (Schicht **3**) läuft und einen gemeinsamen Geheimschlüssel erfordert, statt dessen verwendet werden können.

[0030] Der erste Benutzerauthentisierungsschritt und der zweite Benutzerauthentisierungsschritt können jeweils einen oder mehrere Unterschritte enthalten. Der erste Benutzerauthentisierungsschritt kann beispielsweise einen ersten Unterschritt umfassen, während dem der verschlüsselte Kanal eingerichtet wird. Der erste Unterschritt kann des Weiteren die Authentisierung der Server-Infrastruktur umfassen. Im zweiten Unterschritt kann der verschlüsselte Kanal zur Übertragung von Informationen, die zur Authentisierung der Client-Seite erforderlich sind, herangezogen werden. Vorzugsweise umfassen die übertragenen Informationen ein Zertifikat, das der Client-Infrastruktur zur Verfügung steht.

[0031] Umfasst wie oben beschrieben ein erster Unterschritt die Authentisierung der Server-Infrastruktur (wahlweise nach oder in Zusammenhang mit dem Einrichten eines verschlüsselten Kanals), und umfasst der zweite Unterschritt die tatsächliche Benutzerauthentisierung unter Verwendung des zweiten Authentisierungsschlüssels, kann der so eingerichtete Kommunikationskanal als gegenseitig authentisierter (sicherer) Kommunikationskanal betrachtet werden. Dieser sichere Kommunikationskanal kann sich über einen verschlüsselten oder über einen nicht verschlüsselten Kanal erstrecken.

[0032] Da der zweite Authentisierungsschlüssel sowohl am ersten als auch am zweiten Benutzerauthentisierungsschritt beteiligt ist, kann die Authentisierungssicherheit erhöht werden, indem eine Gleichheitsprüfung durchgeführt wird, um zu verifizieren, dass der zweite Authentisierungsschüssel tatsächlich sowohl für den ersten als auch den zweiten Benutzerauthentisierungsschritt verwendet worden ist. Eine solche Verifizierung ist besonders nützlich, wenn die Authentisierungsschritte von verschiedenen Komponenten gesteuert werden, z. B. wenn der erste Benutzerauthentisierungsschritt von der Eingangsstelle und der zweite Benutzerauthentisierungsschritt vom Anwendungsserver der Server-Infrastruktur gesteuert wird. Vorzugsweise wird die Verifizierung vom Anwendungsserver durchgeführt.

[0033] Gemäß einer bevorzugten Ausführungsform umfasst der zweite Benutzerauthentisierungsschritt eine doppelte Signatur. Im Einzelnen wird in einem ersten Unterschritt der Challenge mit dem zweiten Authentisierungsschlüssel signiert, um eine erste Signatur zu erzeugen, in einem zweiten Unterschritt wird die erste Signatur mit dem ersten Authentisierungsschlüssel signiert, um eine zweite Signatur zu erzeugen, und in einem dritten Unterschritt wird die doppelte Signatur über den sicheren Kommunikationskanal übertragen. Vorzugsweise wird der Challenge vom Anwendungsserver oder einer anderen Komponente des zweiten Netzwerks erzeugt, und die doppelte Signatur wird zum Anwendungsserver oder zur anderen Netzwerkkomponente zurück übertragen.

[0034] Aufforderungen zur Benuthergenehmigung sind ein wichtiges Mittel zur Sicherstellung einer sicheren und vertrauenswürdigen Umgebung. Obwohl Aufforderungen zur Benutzergenehmigung im Prinzip von jeder Netzwerkkomponente wie dem Client, der Eingangsstelle oder dem Anwendungsserver erzeugt werden können, sind vom Kartenleser erzeugte Aufforderungen zur Benutzergenehmigung im Hinblick auf Sicherheit und Vertraulichkeit besonders vorteilhaft. Der Grund hierfür ist die Tatsache, dass der Kartenleser dann als Schutz zwischen der Chipkarte und externen auf die Chipkarte zugreifenden Komponenten fungiert.

[0035] Der Schritt der Benutzerauthentisierung z. B. weist eine solche Aufforderung auf. Der Benutzer kann über das Anzeigegerät des Kartenlesers aufgefordert werden, zu genehmigen, dass die Authentisierungsprozedur tatsächlich gestartet werden soll. Der Kartenleser kann entsprechende Eingabegeräte wie Tasten haben, die einen benutzergesteuerten Betrieb des Kartenlesers im Sinne einer Genehmigung der Authentisierungsaufforderung ermöglichen.

[0036] Im Falle der Genehmigung durch den Benutzer kann der Kartenleser automatisch ein Zeitintervall überwachen, während dem eine bestimmte Anzahl von Signaturen zugelassen ist. Beispielsweise kann der Kartenleser ein Zeitintervall überwachen, während dem zwei Signaturen mit dem zur Chipkarte gehörigen zweiten Authentisierungsschlüssel und eine einzige Signatur mit dem zum Kartenleser gehörigen ersten Authentisierungsschlüssel zulässig sind. Eine derartige Kontrolle des Kartenlesers verhindert den Missbrauch dieser Authentisierungsschlüssel.

[0037] Gemäß einer bevorzugten Ausführungsform werden zusätzliche Funktionen bezüglich eines Fernmanagements der Chipkarte implementiert. Zu die-

sem Zweck kann ein end-to-end Managementkanal zwischen dem zweiten Netzwerk und der Chipkarte, wahlweise über mindestens entweder den verschlüsselten oder über den sicheren Kommunikationskanal, eingerichtet werden. Vorzugsweise ist die Fernmanagementumgebung der Chipkarte so konfiguriert, dass der Kartenleser Chipkarten-Managementbefehle erkennt und sie ohne sie zu interpretieren transparent an die Chipkarte schickt. Die Managementbefehle können auf die Änderung von Dateien oder die Erzeugung neuer Dateien auf der Chipkarte gerichtet sein.

[0038] Zur Erzeugung des Managementkanals kann eine spezielle Chipkarten-Managementkomponente vorgesehen werden. Eine solche Managementkomponente kann beispielsweise ein Server sein, der im zweiten Netzwerk angeordnet ist und als Zertifizierungsstelle fungiert. In diesem Fall kommuniziert die Zertifizierungsstelle über den end-to-end Managementkanal mit der Chipkarte. Die Zertifizierungsstelle ist zuständig für die Erstellung und/oder Aktualisierung eines oder mehrerer zur Chipkarte gehöriger Zertifikate, die vorzugsweise auf ihr gespeichert sind. Außerdem kann die Zertifizierungsstelle Schlüssel ersetzen oder neue Schlüssel zur Chipkarte hinzufügen. Die über den Managementkanal implementierte Managementfunktionalität könnte auch das Hochladen und/oder Personalisieren von Anwendungen (nach Ausgabe der Chipkarte) und die Prüfung (Auditing) der Chipkarte aufweisen.

[0039] Das erfindungsgemäße Verfahren kann unter Verwendung geeigneter Hardware- und Softwarekomponenten implementiert werden. Soweit es die Software betrifft, sieht die Erfindung ein Computerprogrammprodukt mit Programmcodeeinrichtungen zur Durchführung der Schritte des Verfahrens vor, wenn das Computerprogrammprodukt auf einem Computersystem läuft. Das Computerprogrammprodukt kann auf einem computerlesbaren Auszeichnungsmedium gespeichert sein.

[0040] Unter dem Gesichtspunkt der Hardware kann die Erfindung als ein Netzwerksystem konfiguriert sein, das die oben dargelegte Client- und Server-Infrastruktur aufweist. Zwischen der Client-Infrastruktur und der Server-Infrastruktur kann ein verschlüsselter oder nicht verschlüsselter Kommunikationskanal eingerichtet werden, über den eine (vorzugsweise gegenseitig) authentisierte Verbindung, eine signierte Datenübertragung und/oder ein sicherer Managementkanal eingerichtet werden können/kann. Um die Kompatibilität zwischen einer bestimmten auf dem Client laufenden Anwendung und verschiedenen Chipkarten sicherzustellen, kann ein Wrapper vorgesehen werden, der den Client für die Client-Chipkarten-Kommunikation entsprechend konfiguriert. Der Kartenleser der Client-Infrastruktur kann ein Leser Klasse **4** (oder höher) oder ein FINREAD-kompatibler Kartenleser sein.

[0041] Außer einem Server, auf dem die Anwendung läuft, kann die Server-Infrastruktur eine oder mehrere weitere Komponenten enthalten. Beispielsweise kann die Server-Infrastruktur ein zweites Netzwerk, vorzugsweise ein sicheres Inteanet, umfassen, in dem der Anwendungsserver angeordnet ist. Des Weiteren kann die Server-Infrastruktur eine Eingangsstelle in das zweite Netzwerk besitzen. Das zweite Netzwerk kann also über diese Eingangsstelle mit dem ersten Netzwerk gekoppelt werden.

[0042] Eine erfindungsgemäße Server-Infrastruktur, die über das erste Netzwerk mit der Client-Infrastruktur gekoppelt ist, kann einen verschlüsselten oder nicht verschlüsselten Kommunikationskanal über das erste Netzwerk sowie eine (vorzugsweise gegenseitig) authentisierte Verbindung und eine signierte Datenübertragung aufweisen. Das erste Netzwerk, das die Client- und die Server-Infrastrukturen verbindet, kann das Internet oder jedes andere nicht sichere externe Netzwerk sein. Das zweite, den Anwendungsserver enthaltende Netzwerk ist vorzugsweise ein sicheres Inteanet.

[0043] Die Eingangsstelle vom ersten in das zweite Netzwerk kann eine spezielle Hardware- oder Softwarekomponente sein. Beispielsweise kann ein separater Proxy-Server, wenn möglich in einer DeMilitarized Zone (DMZ) und/oder ein drittes Netzwerk in Form eines öffentlichen internen Netzwerks wie ein DMZ verwendet werden. Die Eingangsstelle kann sich auch in Form einer zusätzlichen Hardware- oder Softwarekomponente auf dem Anwendungsserver befinden.

[0044] Ein DMZ-Server für den öffentlichen Zugriff stellt für das zweite Netzwerk eine zusätzliche Sicherheitsmaßnahme dar. Des Weiteren wird der Netzwerkdurchsatz erhöht, da der externe Verkehr nicht mehr im zweiten Netzwerk erscheint. Die DMZ kann als Software-Implementierung verwirklicht werden, die auf dem internen Host wie etwa auf einer Workstation läuft oder auf einem Server, der als DMZ-Server zu verwenden ist.

[0045] Anstelle von oder zusätzlich zu einer DMZ kann eine Proxy-Server-Komponente bereitgestellt werden. Der Proxy-Server dient zum Zugriff auf Informationen wie Seiten im weltweiten Netz (world wide web – WWW), die auf einem weiteren (Anwendungs)-Server gespeichert sind. Fordert ein Client solche Informationen an, werden sie vom Proxy-Server abgerufen und dann an den anfordernden Client geschickt. Der Endeffekt dieser Aktion ist, dass der die Informationen bereitstellende entfernte Server niemals in direkten Kontakt mit irgendwelchen anderen Systemen als dem Proxy-Server kommt.

[0046] Ist ein Managementkanal zwischen dem zweiten Netzwerk und der Chipkarte einzurichten, weist das zweite Netzwerk vorzugsweise einen als Zertifizierungsstelle fungierenden Server auf, der Managementbefehle bezüglich des Managements eines zur Chipkarte gehörigen Zertifikats erzeugt. Die Zertifizierungsstellen-Funktionalität des zweiten Netzwerks umfasst vorzugsweise nicht nur das Management eines oder mehrerer Zertifikate der Chip-

karte nach Ausgabe der Chipkarte, sondern auch das Erzeugen eines oder mehrerer Zertifikate der Chipkarte vor deren Ausgabe. Folglich kann die komplette Zertifizierungsstellen-Funktionalität zu demselben Netzwerk gehören, das die Anwendung unterstützt, die sich des zur Chipkarte gehörigen Zertifikats für Zwecke der Authentisierung, Erzeugung digitaler Signaturen und dgl. bedient.

BESCHREIBUNG DER ZEICHNUNGEN

[0047] Weitere Vorteile der Erfindung ergeben sich aus der folgenden Beschreibung einer bevorzugten Ausführungsform der Erfindung in Zusammenhang mit den beiliegenden Zeichnungen; es zeigen:
[0048] **Fig.** 1 ein schematisches Diagramm des Netzwerksystems gemäß der Erfindung mit Einzelheiten der Client-Infrastruktur;
[0049] **Fig.** 2 den Softwarestapel des Client;
[0050] **Fig.** 3 das Netzwerksystem gemäß der Erfindung mit Einzelheiten der Server-Infrastruktur;
[0051] **Fig.** 4 ein Flussdiagramm, das die Einstellung des Betriebsmodus des Kartenlesers darstellt;
[0052] **Fig.** 5 einen Überblick über das Öffnen einer Anwendung auf der Chipkarte und der Verifizierung des Benutzers;
[0053] **Fig.** 6 ein Flussdiagramm, das die Schritte der Benutzerverifizierung darstellt;
[0054] **Fig.** 7 ein Flussdiagramm, das die Schritte des Schlüsselmanagement darstellt;
[0055] **Fig.** 8 ein Flussdiagramm, das die Schritte der Genehmigung durch den Benutzer darstellt;
[0056] **Fig.** 9 einen Überblick über den Benutzer-Authentisierungsprozess;
[0057] **Fig.** 10 einen Überblick über die Erzeugung einer doppelten Signatur;
[0058] **Fig.** 11 einen Überblick über das Fernmanagement der Chipkarte;
[0059] **Fig.** 12 ein Flussdiagramm der Schritte, die einer sicheren E-Mall-Übertragung vorausgehen; und
[0060] **Fig.** 13 einen Überblick über den Benutzer-Authentisierungsprozess für den Fall, dass derselbe Signaturschlüssel sowohl zur Benutzer- als auch Datenauthentisierung verwendet wird.

BESCHREIBUNG EINER BEVORZUGTEN AUSFÜHRUNGSFORM

[0061] Obwohl die vorliegende Erfindung in jedem Netzwerksystem verwirklicht werden kann, das eine Authentisierung des Benutzers über einen Kartenleser mit Zugriff auf eine Chipkarte erfordert, wird die folgende Beschreibung einer bevorzugten Ausführungsform beispielhaft bezüglich einer Internet-basierten Anwendung in Form einer sicheren E-Banking-Lösung gegeben.
[0062] Gemäß einer ersten Variante der Erfindung wird derselbe Signaturschlüssel der Chipkarte sowohl zur Benutzer- als auch Datenauthentisierung verwendet. Gemäß einer zweiten Variante der Erfindung werden zwei spezielle Signaturschlüssel verwendet. Der zuletzt genannten Fall bedeutet, dass auf der Chipkarte mindestens zwei spezielle Signaturschlüssel gespeichert werden müssen, nämlich ein erster Schlüssel (als Authentisierungsschlüssel bezeichnet) zur Authentisierung des Benutzers und ein zweiter Schlüssel (als Signaturschlüssel bezeichnet) zur Authentisierung von Daten. Im Folgenden werden beide Varianten der Erfindung erörtert, wobei zunächst auf die zweite eingegangen wird.

I. Benutzer- und Datenauthentisierung mit speziellen Signaturschlüsseln

1. Netzwerksystem

[0063] In **Fig.** 1 ist ein Netzwerksystem **10** gemäß der Erfindung schematisch dargestellt. Das Netzwerksystem **10** weist eine Client-Infrastruktur **12** auf, die über ein erstes Netzwerk in Form des Internet **14** mit einer entfernten Server-Infrastruktur **16** kommuniziert. Obwohl in **Fig.** 1 nur eine einzige Client-Infrastruktur **12** dargestellt ist, kann die Server-Infrastruktur **16** mit einer Mehrzahl von Client-Infrastrukturen gleichzeitig kommunizieren.
[0064] Die Kommunikationsverbindung **18** zwischen der Client-Infrastruktur **12** und dem Internet **14** kann eine drahtlose oder drahtgebundene Verbindung sein. Das Gleiche gilt für die Kommunikationsverbindung **20** zwischen dem Internet **14** und der Server-Infrastruktur **16**.

1.1 Client-Infrastruktur

[0065] Wie aus **Fig.** 1 ersichtlicht ist, weist die Client-Infrastruktur **12** einen Client **22**, einen Kartenleser **24** und eine Chipkarte **26** auf.

1.1.1 Chipkarte

[0066] Die Chipkarte **26** ist eine so genannte Java-Karte mit einem IC **28**, der eine PKSC#15-Anwendung **30** unterstützt. PKCS ist die Abkürzung für Public Key Cryptography Standards. Diese Normen gestatten es Anwendungen, beginnend bei WAP-Browsern bis zu sicheren E-Mall-Clients, miteinander zu arbeiten. Die PKSC#15 beschreibt eine Norm für das Format der Verschlüsselungs-Credentials (Zertifikat und privater Schlüssel), die auf Verschlüsselungs-Token wie der Chipkarte **26** gespeichert sind.
[0067] Der Austausch von Meldungen zwischen der Chipkarte **26** und einer entsprechenden Schnittstelle erfolgt mittels spezifischer Befehls-/Antwort-Application Protocol Data Units (APDU's) der PKSC#15-Anwendung **30**, die auf der Chipkarte **26** läuft. Einige PKSC#15-Befehls-APDU's werden später beispielhaft detalllierter beschrieben.
[0068] Das IC **28** der Chipkarte **26** stellt Speicherplätze für eine Mehrzahl von auf der Chipkarte **26** zu

speichernder Berechtigungsnachweise bereit. Ein erster auf der Chipkarte **26** gespeicherter Satz Berechtigungsnachweise wird für die Benutzerauthentisierung verwendet. Diese Berechtigungsnachweise weisen einen privaten **1024** RSA-Schlüssel $K_{PRIV\_AUT\_CLIENT}$ auf, der während der Personalisierung der Karte auf der Chipkarte **26** erzeugt wird, und ein X.509 Client-Zertifikat C_{AUT_CLIENT}, das von einer speziellen Zertifizierungsstelle (Certificate Authority – CA) des Anwendungsanbieters für die Client-Authentisierung ausgestellt und während der Kartenpersonalisierung gespeichert wird.

[0069] Ein zweiter auf der Chipkarte **26** gespeicherter Satz Berechtigungsnachweise wird zum Signieren von Daten verwendet, die beispielsweise Bank-Transaktionen betreffen. Der zweite Satz Berechtigungsnachweise weist einen privaten **1024** Bit RSA-Schlüssel $K_{PRIV\_SIG\_CLIENT}$ auf, der während der Kartenpersonalisierung auf der Chipkarte **26** erzeugt wird. Der zweite Satz Berechtigungsnachweise weist außerdem ein X.509 Client-Zertifikat C_{AUT_CLIENT} auf, das von einer speziellen CA für Bank-Transaktionen des Anwendungsanbieters ausgestellt und während der Kartenpersonalisierung gespeichert wird. Es ist zu beachten, dass bei der vorliegenden Ausführungsform die Zertifikate für Authentisierung und Signierung von verschiedenen CA's ausgegeben werden. Dies hat den Vorteil, dass auf dem Client **22** laufende Browser automatisch das richtige Zertifikat zur Client-Authentisierung wählen können. Es ist weiter zu beachten, dass alle Zertifikate anonym sein können.

[0070] Ein dritter auf der Chipkarte **26** gespeicherter Satz Berechtigungsnachweise dient zum Management des PKCS#15-Dateisystems. Der dritte Satz Berechtigungsnachweise weist einen kartenspezifischen (abgeleiteten) Triple Data Encryption Standard DES-Schlüssel K_{EA}, auf, der zur Authentisierung und Verschlüsselung dient, wenn ein sicherer Kanal zu Zwecken des Dateisystemmanagement wie das Aktualisieren eines Zertifikats auf der Chipkarte **26** zwischen der Server-Infrastruktur **16** und der Chipkarte **26** eingerichtet wird. Der dritte Satz Berechtigungsnachweise weist des Weiteren einen kartenspezifischen (abgeleiteten) Triple DES-Schlüssel K_{MAC} auf, der zur Erzeugung eines Message Authentication Code (MAC) dient, wenn ein sicherer Kanal zu Zwecken des Dateisystemmanagements zwischen der Server-Infrastruktur **16** und der Chipkarte **26** eingerichtet wird. MAC bezeichnet eine Verschlüsselungs-Hash-Funktion, die für den Hash-Wert eines geheimen Schlüssels gebraucht wird. Die DES-Schlüssel können während der Kartenpersonalisierung mittels der eindeutigen 4-Byte-Chip-ID des IC **28** abgeleitet werden.

[0071] Um die PKCS#15-Anwendung sowie die Chipkarte **26** auf sichere Weise von der entfernten Server-Infrastruktur **16** zu managen, wird eine Open Platform (OP)-Schnittstelle bereitgestellt. Ein OP-Kartenmanager ist eine Kartenmanagementinstanz, die stets auf einer OP-gemäßen Java-Karte wie die Chipkarte **26** vorhanden ist. Er stellt Mittel zum Ausführen von Kartenfunktionen wie Laden der Anwendung, Löschen der Anwendung, Prüfung der Karte (Auditing) etc. bereit. Der OP-Kartenmanager auf der Chipkarte **26** hat drei kartenspezifische Triple DES-Schlüssel, die während der Kartenpersonalisierung abgeleitet werden. Ein erster Schlüssel K_{EA_CM} dient zur OP-Authentisierung und Verschlüsselung, ein zweiter Schlüssel K_{MAC_CM} dient zur OP MAC-Erzeugung und ein dritter Schlüssel K_{KEK_CM} dient zur Verschlüsselung des OP-Schlüssels.

[0072] Der Zugriff auf die Chipkarte **26** geschieht von zwei Quellen aus, nämlich von einem auf dem Client laufenden Browser und einem speziellen (Java Script) Programm, das in der von der Server-Infrastruktur **16** bereitgestellten Web-Seite enthalten ist. Während der Browser auf die Chipkarte zugreift, um die Client-Authentisierung unterhalb der Anwendungsschicht auszuführen, verwendet das in der Web-Seite enthaltene spezielle Programm die Chipkarte **26**, um Sicherheitsfunktionen auf der Anwendungsschicht durchzuführen wie Signieren von Transaktionen und zusätzliche Authentisierung des Client. Um Zugriff auf die Chipkarte **26** zu erhalten, benützt das spezielle Programm ein signiertes Java-Applet, das innerhalb des Browsers läuft.

1.1.2 Kartenleser

[0073] Der Kartenleser **24** stellt den Zugriff auf die Chipkarte **26** bereit und kommuniziert über eine Kommunikationsverbindung **34** mit dem Client **22**. Diese Kommunikationsverbindung **34** kann eine drahtgebundene Verbindung sein wie eine Universal Serial Bus (USB)-Verbindung oder eine drahtlose Verbindung, z. B. gemäß der Bluetooth-Norm. Der Kartenleser **24** ermöglicht die Personalisierung sowie Softwareaktualisierungen und genügt vorzugsweise den FINREAD-Anforderungen und/oder den Anforderungen der Klasse **4**.

[0074] Wie aus **Fig.** 1 ersichtlich ist, hat der Kartenleser **24** einen Tastenblock **36** zum sicheren Management der persönlichen Identifikationsnummer (Personal Identification Number – PIN). Der Kartenleser **24** stellt sicher, dass der Tastenblock **36** nur von der internen Lesersoftware und nicht von Software aus, die auf dem Client **22** läuft, verwendet werden kann. Außer dem Tastenblock **36** hat der Kartenleser **24** eine Anzeige **38**, die zur Anzeige beispielsweise von Daten vor ihrer Übertragung an die Chipkarte **26** dient. Wie im Falle des Tastenblocks stellt der Kartenleser **24** sicher, dass die Anzeige **38** nur von der internen Lesersoftware und nicht von Software aus, die auf dem Client **22** läuft, betrieben werden kann. Dies garantiert, dass die angezeigten Daten dem zur Erzeugung der Signatur verwendeten Schlüssel entsprechen. Wie nachstehend detalllierter beschrieben wird bedeutet dies, dass eine Standardaufforderung in dem Fall angezeigt wird, dass $K_{PRIV\_AUT\_CLIENT}$ zu ver-

wenden ist, und dass die gesamten zu signierenden Transaktionsdaten in dem Fall angezeigt werden, dass $K_{PRIV\_SIG\_CLIENT}$ zu verwenden ist.

[0075] Der Kartenleser **24** kann zumindest bei speziellen (kompatiblen) Chipkarten **26** in einem „sicheren Modus" betrieben werden. Im sicheren Modus des Kartenlesers **24** werden mindestens einige Befehle nicht zur Chipkarte **26** weitergeleitet, ohne auf der Anzeige **38** angezeigt zu werden. Dies ist erforderlich, um Sicherheitsmerkmale wie das sichere PIN-Management durchzusetzen. Außerdem weist der in Zusammenhang mit der in **Fig.** 1 dargestellten Ausführungsform beispielhaft beschriebene Kartenleser **24** einen „transparenten Modus" für Chipkarten auf, die mit dem sicheren Modus nicht kompatibel sind. Im transparenten Modus zeigt der Kartenleser **24** keine Befehle an und leitet die Befehle ohne irgendwelche Maßnahmen zu ergreifen an die Chipkarte weiter.

[0076] Der Kartenleser **24** ist so konfiguriert, dass er **1024** Bit-RSA-Signaturen mit seinem privaten Schlüssel erzeugen kann. Dies beinhaltet das Durchführen einer Hash-Funktion gemäß dem SHA-1-Algorithmus über Eingangsdaten mit variabler Länge. Das Ergebnis, ein **160** Bit-Wert, ist gemäß PKCS#1 zu verarbeiten. Um einer Komponente der Server-Infrastruktur **16** die Möglichkeit zur Prüfung zu geben, ob ein bestimmter Kartenleser **24** für eine bestimmte Operation verwendet worden ist oder nicht, ist der Kartenleser **24** so konfiguriert, dass er von der Chipkarte **26** empfangene Daten signiert, wenn der Kartenleser **24** im sicheren Modus arbeitet, was später detalllierter beschrieben wird.

[0077] Der Kartenleser **24** wird während der Leserherstellung mit eindeutigen privaten Schlüsseln und entsprechenden Zertifikaten sicher personalisiert. Die Zertifikate können von einer CA des Anwendungsanbieters vorerzeugt und dann dem Hersteller des Kartenlesers zur Personalisierung geliefert werden. Dies bietet den Vorteil, dass der private Schlüssel der CA zur Erzeugung der Zertifikate nicht im Betrieb des Leserherstellers verfügbar sein muss.

[0078] Der Kartenleser **24** wird mit einem individuellen privaten **1024** Bit RSA-Schlüssel K_{PRIV_READER} zusammen mit einem X.509 Leser-Zertifikat C_{READER} initialisiert. Des Weiteren enthält der Kartenleser **24** Berechtigungsnachweise, die zur Aktualisierung der Lesersoftware auf Basis der Verschlüsselung mittels privater Schlüssel dienen. Softwareaktualisierungen werden mit Schlüsseln signiert, die vom Anwendungsanbieter, der die Server-Infrastruktur **16** betreibt, kontrolliert werden. Dieses Signieren eingehender Daten stellt sicher, dass nur authentisierte Softwareaktualisierungen akzeptiert werden. Somit sind Integrität und Authentizität gewährleistet. Jede Aktualisierung der Lesersoftware muss auf eine solche Weise durchgeführt werden, dass die während der ursprünglichen Leserinitialisierung eingestellten Daten nicht zerstört werden, d. h. dass Schlüssel und Zertifikate bestehen bleiben.

[0079] Der Kartenleser **24** stellt Einrichtungen wie ein Sicherheitsmodul für die sichere manipulationsgeschützte Speicherung der Schlüssel für Verschlüsselungszwecke bereit. Insbesondere müssen K_{PRIV_READER} sowie die zum Schutz der Softwareaktualisierungsprozedur verwendeten Schlüssel in einem derartigen sicheren Speicherplatz gespeichert werden. Der sichere Speicherplatz kann nicht aus dem Kartenleser **24** entfernt werden und ein Zugriff auf ihn von auf dem Client **22** laufender Software ist nicht möglich.

1.1.3 Client

[0080] Der Client **22** kann ein Personal Computer (PC) oder jede andere Komponente sein, die in der Lage ist, einerseits eine Verbindung über das Internet **14** zur Server-Infrastruktur **16** und andererseits zum Kartenleser **24** herzustellen. Der Client **22** kann beispielsweise auch ein mobiles Endgerät wie ein Mobiltelefon sein, das mit der Server-Infrastruktur **16** über das Internet **14** gemäß dem Wireless Application Protocol (WAP) und über eine Bluetooth-Verbindung mit dem Kartenleser **24** kommuniziert. Im Folgenden wird angenommen, dass der Client **22** ein PC ist.

[0081] In **Fig.** 2 ist der Softwarestapel des Client **22** schematisch dargestellt. Die unterste Schicht besteht aus einem Betriebssystem **40** und einem Lesertreiber **42**. Der Schicht **40** des Betriebssystems folgt eine PKCS#11-Schicht **44**, die eine Schnittstelle für die Chipkarte bildet. PKCS#11 definiert eine technologieunabhängige Programmierschnittstelle für Kryptotoken wie die Chipkarte **26**. Die PKCS#11-Schicht 44 ist für Standardfunktionen wie die Unterstützung der SSL-Benutzerauthentisierung oder die Secure Multipurpose Internet Mall Extension (S/MIME oder einfach sichere E-Mail) erforderlich.

[0082] Gemäß der Erfindung wird die Standard-PKCS#11-Architektur mit einer Mehrzahl von Erweiterungen ergänzt. Die Standard-PKCS#11-Funktionalität wird beispielsweise hinsichtlich der Abfrage des Leserzertifikats ergänzt. Dies bedeutet, dass zusätzlich zum Lesen sämtlicher Informationen von der Chipkarte **26** auch das Zertifikat CREADER des Kartenlesers **24** abgefragt werden kann. CREADER wird mittels einer PKCS#15 Befehls-APDU mit der Bezeichnung GET READER CERTIFICATE angefordert. Dies ist offensichtlicherweise kein Chipkartenbefehl. Bei Empfang der Befehls-APDU GET READER CERTIFICATE schickt der Kartenleser **24** keinerlei Befehle oder leitet Befehle weiter an die Chipkarte **26**. Statt dessen schickt der Kartenleser **24** den angeforderten Teil von C_{READER} zurück zum Client **22**. Typischerweise schickt der Kartenleser **24** Antwort-APDU's in voller Länge zurück mit Ausnahme des letzten Teils von C_{READER} Ist der angeforderte Zertifikatteil nicht verfügbar, schickt der Kartenleser **24** eine Fehlermeldung zurück. Dadurch kann die Client-Software einen bestimmten Zähler hochzählen, bis ein Fehler auftritt, der angibt,

dass C_{READER} vollständig gelesen worden ist.

[0083] Eine weitere ergänzende Funktionalität der PKCS#11-Schicht **44** betrifft das Fern-Management der Chipkarte. Bei der bevorzugten Ausführungsform basiert das Management der Chipkarte auf einer sicheren end-to-end Verbindung zwischen dem Anwendungsserver oder einer anderen Komponente der Server-Infrastruktur **16** und der Chipkarte **26**. Um einen solchen sicheren Managementkanal aufzubauen, müssen managementbezogene APDU's unter Umgehung der PKCS#11-Schicht **44** direkt an den Kartenleser **24** geleitet werden. Dies erfordert eine spezielle Erweiterung in der PKCS#11-Bibliothek, die es gestattet, managementbezogene APDU's der Chipkarte transparent durch die PKCS#11-Schicht **44** zu führen.

[0084] Eine dritte PKCS#11-Erweiterung betrifft die Erzeugung doppelter Signaturen. Zu diesem Zweck wird die PKCS#11-Funktionalität so erweitert, dass Klartextmeldungen beliebiger Länge signiert und doppelte Signaturen zurückgeschickt werden können. Diese Funktionalität basiert auf einer PKCS#15-Befehls-APDU mit der Bezeichnung COMPUTE DIGITAL SIGNATURE.

[0085] Nunmehr sei wieder auf **Fig.** 2 verwiesen, gemäß der oberhalb der PKCS#11-Schicht **44** eine Wrapper-Schicht **46** angeordnet ist. In dem Fall, in dem mehrere PKCS#11-Bibliotheken für verschiedene Typen von Chipkarten erforderlich sind, wählt der PKCS#11-Wrapper automatisch die richtige PKCS#11-Bibliothek, ohne den Benutzer einzubeziehen. Der PKCS#11-Wrapper schickt alle Application Program Interface (API)-Aufrufe an die entsprechende kartenspezifische PKCS#11-Bibliothek. Der Wrapper kann entfallen, wenn nur ein einziger Typ Chipkarte **26** unterstützt zu werden braucht.

[0086] Wie in **Fig.** 2 dargestellt, ist oberhalb der PKCS#11-Wrapper-Schicht 46 ein Browser **48** angeordnet. Je nach Typ des Browsers **48** können weitere Softwarekomponenten zwischen dem Browser **48** und der PKCS#11-Wrapper-Schicht 46 angeordnet werden. Für den Fall, in dem z. B. der Microsoft Internet Explorer verwendet wird, ist eine Cryptographic Service Provider (CSP)-Schicht erforderlich, um zur SSL-Benutzerauthentisierung und S/MIME auf die Chipkarte zugreifen zu können. Um Zugriff auf die Chipkarte **26** zu erlangen, bedient sich der Browser **48** eines Java-Applet **50**, das im Browser **48** läuft. Das Java-Applet **50** wird von einer Komponente der Server-Infrastruktur **16** signiert. Anstelle des Java-Applet **50** könnte ein Component Object Model (COM) oder ein Browser-Plug-In verwendet werden.

1.2 Server-Infrastruktur

[0087] In **Fig.** 3 ist das Netzwerksystem **10** bezüglich der Server-Infrastruktur **16** detalllierter dargestellt. Wie aus **Fig.** 3 ersichtlich ist, weist die Server-Infrastruktur **16** ein zweites Netzwerk in Form eines sicheren Inteanet **52** und ein drittes Netzwerk in Form einer DMZ auf. Die Client-Infrastruktur **12** hat Zugriff auf das Intranet **52** über das Internet **14** und die DMZ **54**.

[0088] Das in **Fig.** 3 dargestellte Inteanet **52** weist zwei Server auf. Ein erster im Inteanet **52** angeordneter Server dient als Anwendungsserver **58** und ein zweiter Server **60** stellt CA-Funktionalitäten bereit. Der CA-Server **60** ist zuständig für das Fernmanagement der Chipkarte. Zu diesem Zweck richtet der CA-Server **60** einen sicheren Managementkanal zur Chipkarte **26** ein, wie nachstehend in Abschnitt 2.6 beschrieben wird. Der Anwendungsserver **58** enthält eine Eingangs-Web-Seite für das Homebanking sowie eine sichere Banking-Web-Seite. Einzelheiten bezüglich des Zugriffs auf diese Web-Seiten werden nachfolgend erörtert.

[0089] Es ist zu beachten, dass die DMZ **54**, der Proxy-Server **56** sowie weitere Software- und/oder Hardwarekomponenten, die als Eingangsstelle bezüglich des Inteanet **52** fungieren, nicht bei dem Inteanet **52** angeordnet werden müssen, sofern geeignete Kommunikationsverbindungen **61**, **63** zum Anwendungsserver **58** und zum CA-Server **60** eingerichtet werden können. Alternativ können eine oder mehrere dieser Einsprungstellen-Funktionalitäten durch entsprechende, auf dem Anwendungsserver **56** angeordnete Software- oder Hardwaremodule verwirklicht werden.

[0090] Die meisten Aspekte der Erfindung können auch dann in die Praxis umgesetzt werden, wenn der CA-Server **60** nicht Bestandteil des Inteanet **52** ist, sondern von einem externen Diensteanbieter betrieben wird. Ist nur die erfindungsgemäße Fernmanagement-Funktionalität für die Chipkarte erforderlich, kann außerdem der Benutzer-Authentisierungsprozess entfallen.

2. Anwendungsfluss

2.1 Einrichten einer Verbindung zwischen der Client- und der Server-Infrastruktur

[0091] Um eine Sitzung zu starten, richtet der Client **22** eine Kommunikationsverbindung **18** zum Internet **14** ein. Der Client **22** verbindet sich dann über das Internet **14** mit der Server-Infrastruktur **16**, um eine auf dem Anwendungsserver **58** bereitgehaltene Eingangs-Web-Seite abzurufen. Wie in **Fig.** 3 dargestellt beinhaltet diese Verbindung die Eingangsstelle des zweiten Netzwerks **52**, welche in der in **Fig.** 3 dargestellten Ausführungsform die DMZ **54** und den Proxy-Server **56** aufweist.

[0092] Die vom Client **22** geladene Eingangs-Web-Seite enthält das signierte, in **Fig.** 2 dargestellte Java-Applet **50**. Das Java-Applet **50** wird von einem Java-Script-Programm verwendet, das ebenfalls in der vom Client **22** geladenen Eingangs-Web-Seite enthalten ist, um über die PKCS#1-Schicht **44** auf die Chipkarte **26** zuzugreifen. Das Java-Applet **50** wird z. B. verwendet, um das

PKCS#11-Token zu öffnen, um Zugriff auf Schlüssel und Zertifikate zu erhalten, die auf der Chipkarte **26** gespeichert sind.

[0093] Die Eingangs-Web-Seite verwendet SSL, um einen verschlüsselten Kanal einzurichten. Dieser verschlüsselte Kanal ist von Vorteil, da zu einem späteren Zeitpunkt ein auf der Chipkarte **26** gespeichertes und Informationen über den Benutzer enthaltendes Zertifikat an die Server-Infrastruktur **16** geschickt werden muss. Das Einrichten des verschlüsselten Kanals kann jedoch entfallen, wenn das Zertifikat z. B. nur anonymisierte Informationen über den Benutzer enthält.

[0094] Zu Beginn einer Sitzung könnte die Chipkarte **26** noch nicht verfügbar sein. Folglich wird während SSL nur die Server-Infrastruktur **16** authentisiert. Die SSL-Benutzer(Client)-Authentisierung erfolgt bei der vorliegenden Ausführungsform zu einem späteren Zeitpunkt.

[0095] Der verschlüsselte SSL-Kanal mit lediglich Server-Authen-tisierung wird zwischen dem Client **22** und der Eingangsstelle des Intranet **52**, d. h. der DMZ **54** mit dem Proxy-Server **56**, eingerichtet. Der verschlüsselte SSL-Kanal wird wie folgt eingerichtet. Wenn der Client **22** eine Verbindung mit der Server-Infrastruktur **16** für das Laden der Eingangs-Web-Seite anfordert („Hello" des Client), schickt der Proxy-Server **54** oder eine andere als Eingangsstelle fungierende Komponente ein zur Server-Infrastruktur **16** gehöriges Zertifikat C_{EP}. C_{EP} ist vom CA-Server **60** oder einer externen CA signiert worden. Der Client **22** prüft dann, um festzustellen, ob es sich bei der CA um eine von ihm akzeptierte handelt, und verifiziert die Signatur auf C_{EP} unter Verwendung des öffentlichen Schlüssels der CA.

[0096] Im nächsten Schritt vergleicht der Client **22** den Namen in C_{EP} mit dem Domaln Name Server (DNS)-Namen des Servers, von dem er annimmt, dass er mit ihm in Verbindung zu treten versucht. Nach diesem Vergleich verwendet der Client **22** Verschlüsselung mittels eines privaten Schlüssels, um ein Geheimnis mit dem von C_{EP} extrahierten öffentlichen Schlüssel K_{EP} der Server-Infrastruktur **16** zu verschlüsseln. Das verschlüsselte Geheimnis wird an die Server-Infrastruktur **16** geschickt und der Client **22** versucht verschlüsselt mit der Server-Infrastruktur **16** zu kommunizieren, wobei die (symmetrischen) Schlüssel von dem durch den Client **22** mit dem öffentlichen Schlüssel K_{EP} der Server-Infrastruktur **16** verschlüsselten Geheimnis abgeleitet werden.

[0097] Kann der Client **22** erfolgreich mit der Server-Infrastruktur **16** kommunizieren, dann müssen beide das gleiche Geheimnis besitzen, um die korrekten Schlüssel ableiten zu können. Dies zeigt, dass die Server-Infrastruktur **16** den korrekten privaten Schlüssel besitzt und authentisiert so die Server-Infrastruktur **16**. Die weitere Kommunikation zwischen dem Client **22** und der Server-Infrastruktur **16** kann nun über einen verschlüsselten Kanal erfolgen.

2.2 Einstellung der Betriebsart des Lesers und Aktivierung der Chipkarte

[0098] Der in **Fig.** 1 dargestellte Kartenleser **24** kann je nach Typ der mit dem Kartenleser **24** verwendeten Chipkarte **26** in zwei verschiedenen Modi verwendet werden. Wird eine kompatible Chipkarte im Kartenleser **24** verwendet, wird der Kartenleser **24** in den „sicheren Modus" geschaltet. In diesem Modus leitet der Kartenleser **24** sicherheitsunkritische Befehle transparent an die Chipkarte **26** weiter. Bei sicherheitskritischen Befehlen jedoch wie dem PIN-Management oder der Signaturerzeugung sperrt der Kartenleser **24** die Transparenz und ergreift zusätzliche Maßnahmen. Der sichere Modus erfordert also, dass der Kartenleser **24** alle erhaltenen Befehle prüft und entscheidet, ob sie transparent an die Chipkarte **26** weitergeleitet werden oder ob zusätzliche Maßnahmen zu ergreifen sind. Einige PKCS#15- Befehle, die vom Kartenleser **24** im sicheren Modus erkannt werden müssen, wurden bereits erörtert. Die Antwort des Kartenlesers **24** auf diese und einige weitere PKCS#15-Befehle wird nachstehend beschrieben.

[0099] Wird eine nicht kompatible Chipkarte **26** im Kartenleser **24** verwendet, bietet der Kartenleser **24** dem Benutzer an, in den „transparenten Modus" zu schalten. Im transparenten Modus fungiert der Kartenleser **24** als Leser der Klasse **1**, was bedeutet, dass jegliche über die Kommunikationsverbindung **34** erhaltenen Befehle überhaupt nicht geprüft werden.

[0100] Die oben beschriebene Einstellung der Betriebsart wird nunmehr detalllierter anhand des Flussdiagramms von **Fig.** 4 erläutert.

[0101] In einem ersten Schritt **400** verlangt die Banking-Eingangs-Web-Seite das Einführen der Chipkarte **26** in den Kartenleser **24**, das Drücken einer Anmeldetaste auf der Web-Seite und das Befolgen der auf der Anzeige **38** des Kartenlesers **24** angezeigten Anweisungen. Beim Einführen oder anderweitigen Verbinden der Chipkarte **26** in den bzw. mit dem Kartenleser **24** erkennt der Kartenleser **24**, ob es sich um eine kompatible Chipkarte handelt oder nicht. Zu diesem Zweck setzt der Kartenleser **24** die Chipkarte **26** in Schritt **402** zurück, um ihren Answer To Reset (ATR)-String zu erhalten. Erkennt der Kartenleser **24** in Schritt **404**, dass ein spezifischer vordefinierter Substring im ATR enthalten ist, fährt der Kartenleser **24** mit Schritt **406** fort. In Schritt **406** schaltet der Kartenleser **24** in den sicheren Modus, wählt den PKCS#15-Signaturschlüssel $K_{PRIV\_SIG\_CLIENT}$ und setzt einen internen Zeitgeber zurück. Im nächsten Schritt **408** zeigt der Kartenleser **24** die Meldung „Bereit" an.

[0102] Ermittelt der Kartenleser **24** in Schritt **404**, dass die Chipkarte **26** nicht kompatibel ist, fährt er mit Schritt **410** fort. In Schritt **410** wird auf der Anzeige **38** eine Meldung angezeigt, die dem Benutzer das Schalten in den transparenten Modus anbietet. Im

nächsten Schritt **412** wird die Antwort des Benutzers bewertet. Lehnt der Benutzer den transparenten Modus ab, schleift das Verfahren zurück zu Schritt **400**. Akzeptiert dagegen der Benutzer den transparenten Modus, schaltet der Kartenleser **24** in Schritt **414** in den transparenten Modus. Nach Schritt **414** fährt das Verfahren fort, indem in Schritt **408** die Meldung „Bereit" angezeigt wird.

[0103] Nachdem die zutreffende Betriebsart eingestellt worden ist, überwacht der Kartenleser **24** in Schritt **416** ständig, ob eine Befehls-APDU vom Client **22** empfangen wird. Wird eine APDU empfangen, prüft der Kartenleser in Schritt **418**, ob er im sicheren Modus arbeitet. Für den Fall, dass der Kartenleser **24** nicht im sicheren Modus arbeitet, geht er zu Schritt **420** über. In Schritt **420** wird die vom Client **22** empfangene Befehls-APDU an die Chipkarte **26** geschickt und eine von der Chipkarte **26** empfangene resultierende Antwort-APDU an den Client **22** zurückgeschickt.

[0104] Wird andererseits in Schritt **418** ermittelt, dass der Kartenleser **24** im sicheren Modus arbeitet, prüft der Kartenleser **24** in Schritt **422**, ob die empfangene Befehls-APDU der Software des Kartenlesers bekannt ist. Ist die Befehls-APDU nicht bekannt, löscht der Kartenleser **24** seinen Vorzeichenpuffer in Schritt **426** und fährt mit Schritt **420** fort wie oben erläutert. Andernfalls, d. h. wenn in Schritt **422** ermittelt wird, dass die Befehls-APDU der Software des Kartenlesers bekannt ist, beginnt der Kartenleser **24** in Schritt **424** mit der Befehlsverarbeitung für diese Befehls-APDU. Flussdiagramme für verschiedene beispielhafte vom Kartenleser **24** verarbeitete Befehls-APDU's werden nachstehend detalllierter erläutert.

[0105] Es ist zu beachten, dass nach der Wahl eines bestimmten Modus des Kartenlesers **24** dieser bis zum nächsten Karteneinführereignis in diesem Modus bleibt. Des Weiteren wird die Web-Seite, die den Benutzer auffordert, die Chipkarte **26** einzuführen, mit grundlegenden Fehlerbehandlungsmechanismen für den Fall ausgestattet, dass keine Chipkarte **26** verfügbar ist oder der Kartenleser **24** nicht gefunden werden kann.

[0106] Im Folgenden wird angenommen, dass eine kompatible Chipkarte **26** in den Kartenleser **24** eingeführt und der Kartenleser **24** im sicheren Modus betrieben wird.

2.3 Öffnen einer Anwendung und Verifizierung des Benutzers

[0107] Sobald eine Chipkarte **26** aktiviert und ein Modus des Kartenlesers **24** eingestellt worden ist, wird eine bestimmte PKCS#15-Anwendung auf der Chipkarte **26** gewählt und geöffnet. Des Weiteren findet die Verifizierung des Benutzers statt und Informationen über auf der Chipkarte **26** gespeicherte Schlüssel und Zertifikate sowie das Leserzertifikat C_{READER} werden abgerufen. Diese Schritte werden nunmehr detalllierter unter Bezugnahme auf die in **Fig.** 5 dargestellte allgemeine Übersicht erläutert. Wie aus **Fig.** 5 ersichtlich ist, beinhalten die Schritte bezüglich des Öffnens der PKCS#15-Anwendung, der Verifizierung des Benutzers sowie des Abrufens des Schlüssels und des Zertifikats den Browser **48** einschließlich des Java-Applets **50** und des Java-Script-Programms, die PKCS#11-Schicht **44**, den Kartenleser **24** sowie die auf der Chipkarte **26** gespeicherte PKCS#15-Anwendung **30**.

[0108] Der Prozess beginnt (Schritt **500**) mit einem Befehl OPEN TOKEN. Wenn das im Browser **48** enthaltene Java-Applet **50** den Befehl OPEN TOKEN an die PKCS#11-Schicht **44** schickt, wird eine Befehls-APDU SELECT PKCS#15 transparent von der PKCS#11-Schicht **44** über den Kartenleser **24** an die PKCS#15-Anwendung **30** auf der Chipkarte **26** geschickt (Schritte **502** und **504**). Die PKCS#15-Anwendung **30** wird direkt über ihr Anwendungskennzeichen (Application Identifier – AID) gewählt.

[0109] Die PKCS#15-Anwendung schickt eine entsprechende Antwort-APDU transparent an die PKCS#11-Schicht **44** zurück (Schritte **506** und **508**).

[0110] Nach der Wahl der Anwendung wird eine Befehls-APDU VERIFY von der PKCS#11-Schicht **44** an den Kartenleser **24** geschickt, um diesen aufzufordern, die Verifizierung des Benutzers vorzunehmen (Schritt **510**). Der Befehl VERIFY authentisiert den Benutzer gegenüber der PKCS#15-Anwendung **30** auf der Chipkarte **26** mittels einer 6- bis 11-stelligen PIN in OP Global PIN-Codierung. Die OP Global PIN dient der PIN-Verifizierung und dem PIN-Management. Der von der PKCS#11-Schicht zum Kartenleser **24** geschickte Befehl VERIFIY wird nicht von PIN-Daten begleitet, da der Kartenleser **24** im sicheren Modus keine PIN-Daten akzeptiert.

[0111] Im sicheren Modus wird der Befehl VERIFY vom Kartenleser **24** erkannt und der Kartenleser **24** veranlasst, zusätzliche Maßnahmen zu ergreifen. Im Einzelnen zeigt der Kartenleser **24** eine Meldung an, die den Benutzer auffordert, seine PIN über den Tastenblock **36** einzugeben. Zur Verifizierung der PIN beendet der Kartenleser **24** den Befehl VERIFY, indem er die vom Benutzer im OP-Format eingegebene PIN codiert. Schließlich schickt der Kartenleser **24** den abgeschlossenen Befehl in Schritt **514** an die Chipkarte **26**. In Schritt **515** verifiziert die PKCS#15-Anwendung **30** die vom Benutzer eingegebene PIN. Ist die korrekte PIN eingegeben worden, wird eine entsprechende Antwort-APDU an den Kartenleser **24** zurückgeschickt und von diesem angezeigt (Schritte **516** und **518**). Wurde eine falsche PIN eingegeben, zählt die PKCS#15-Anwendung **30** einen OP Global Pin-Wiederholungszähler hoch und die Chipkarte **26** schickt eine Fehlermeldung zurück (Schritt **516**), die in Schritt **518** angezeigt wird.

[0112] Schickt die Chipkarte **26** eine Fehlermeldung zurück und ist sie noch nicht gesperrt, gestattet der Kartenleser **24** einen Wiederholungsversuch oder die Stornierung der Operation und zeigt die Anzahl der

verbleibenden Wiederholungsversuche an. Die PKCS#15-Anwendung **30** wird automatisch gesperrt, d. h. ihr Lebenszyklus wird auf GESPERRT (BLOCKED) gesetzt, wenn die Anzahl der OP Global PIN-Wiederholungsversuche eine vordefinierte Anzahl Wiederholungsversuche überschreitet. Misslingt die Verifizierung des Benutzers aus irgendwelchen Gründen, sendet der Kartenleser **24** ein Statuswort bezüglich eines Fehlerzustands an die PKCS#11-Schicht **44**. Außerdem zeigt der Kartenleser **24** den Text „Karte gesperrt" in dem Fall an, in dem der Fehlerzustand bedeutet, dass die Chipkarte **26** gesperrt ist (keine weiteren Wiederholungsversuche).

[0113] Das Ergebnis der Verifizierung des Benutzers wird vom Kartenleser **24** in Schritt **520** an die PKCS#11-Schicht **44** geschickt. Es kann eine Mehrzahl weiterer transparenter Befehls-APDU's folgen (Schritt **522**). Eine besondere nicht transparente Befehls-APDU, die während des Öffnens des Token verwendet wird, ist der Befehl GET READER CERTIFICATE. Dieser Befehl gestattet der PKCS#11-Schicht **44**, C_{READER} vom Kartenleser **24** anzufordern. Der Befehl schickt den angeforderten Teil von C_{READER} an den Client **22** zurück und macht C_{READER} im PKCS#11-Token sichtbar. Zum Lesen des gesamten C_{READER} sind mehrere Befehle GET READER CERTIFICATE erforderlich. Die Kommunikation zwischen der PKCS#11-Schicht **44** und dem Kartenleser **24** bezüglich des Abrufs von C_{READER} ist durch Schritte **524** und **526** dargestellt.

[0114] Sind die Prozesse des Öffnens der PKCS#15-Anwendung **30**, der Verifizierung des Benutzers und des Abrufs des Leserzertifikats durchgeführt worden, wird in Schritt 528 eine Antwort TOKEN OPEN von der PKCS#11-Schicht **44** an den Browser **48** geschickt.

[0115] Die Befehlsverarbeitung des Kartenlesers **24** bezüglich des Befehls VERIFY wird nunmehr detallierter unter Bezugnahme auf das Flussdiagramm von **Fig.** 6 beschrieben.

[0116] Nach Empfang des Befehls VERIFY in Schritt **600** löscht der Kartenleser **24** den Vorzeichenpuffer in Schritt **602** und fordert in Schritt **604** die PIN des Benutzers an. In Schritt **606** bildet der Kartenleser **24** die Befehls-APDU VERIFY unter Verwendung der vom Benutzer eingegebenen PIN und schickt sie an die Chipkarte **24**. Der Kartenleser **24** empfängt eine Antwort-APDU VERIFY von der Chipkarte **26** und prüft in Schritt **608**, ob die Antwort-APDU VERIFY eine korrekte PIN betrifft. Ist die PIN korrekt, zeigt der Kartenleser **24** eine entsprechende Meldung an und schickt in Schritt **610** ein entsprechendes Statuswort an den Client **22** zurück.

[0117] Andernfalls wertet der Kartenleser **24** die Antwort-APDU VERIFY bezüglich der Frage aus, ob die PIN gesperrt ist (Schritt **612**). Sollte die PIN gesperrt sein, zeigt der Kartenleser **24** eine entsprechende Meldung an und schickt in Schritt **610** ein entsprechendes Statuswort an den Client **22** zurück. Ist die PIN nicht gesperrt, fordert der Kartenleser **24** den Benutzer in Schritt **614** zu einem Wiederholungsversuch oder zum Stornieren auf. Die Antwort des Benutzers wird in Schritt **616** ausgewertet. Sollte der Benutzer einen Wiederholungsversuch wünschen, schleift das Verfahren zu Schritt **604** zurück. Andernfalls geht das Verfahren zu Schritt **618** weiter und schickt ein entsprechendes Statuswort an den Client **22** zurück. Die von der Chipkarte **26** an den Client **22** zurückgeschickten Statuswörter entsprechen den Statuswörtern, wie sie von der Chipkarte **26** in der letzten Antwort-APDU zurückgeschickt werden.

2.4 Authentisierung des Benutzers

[0118] Die Authentisierung des Benutzers umfasst mehrere Aspekte. Ein wesentlicher Aspekt der Authentisierung des Benutzers ist die Frage, ob der Benutzer tatsächlich authentisiert werden möchte. Folglich wird der Authentisierungsprozess nur gestartet, wenn der Benutzer dies genehmigt. Außerdem erfolgt die Authentisierung des Benutzers als zweistufige Prozedur. Mindestens eine der beiden Stufen bezieht die auf dem Anwendungsserver laufende Anwendung mit ein.

2.4.1 Genehmigung des Benutzers

[0119] In Absatz 1.1.1 wurde erwähnt, dass auf der Chipkarte **26** verschiedene Schlüssel gespeichert sind. Insbesondere die Benutzerauthentisierung und das Signieren von Transaktionen (Datenauthentisierung) kann somit mit verschiedenen Schlüsseln erfolgen, um das Durchsetzen eines sicheren Prinzips der Schlüsselverwendung zu gestatten, das deutlich zwischen einer Authentisierung von Zufallsdaten (z. B. SSL) und der Signierung von aussagekräftigem Inhalt unterscheidet. Es ist deshalb möglich, ein Prinzip der Schlüsselverwendung zu implementieren, das zwangsweise immer dann zur Anwendung kommt, wenn verschlüsselte Befehle wie Signieren oder Entschlüsseln zur Chipkarte **26** zu schicken sind. Ein solches Prinzip der Schlüsselverwendung gibt dem Benutzer die Möglichkeit, die Schlüsselverwendung explizit zu kontrollieren.

[0120] Eine bevorzugte Lösung der Kontrolle der Schlüsselverwendung basiert auf Zertifikatserweiterungen. Bei der vorliegenden Ausführungsform werden jedoch Dateikennzeichen für diesen Zweck verwendet. Da die Dateikennzeichen der Schlüsseldateien in der PKCS#15-Anwendung **30** konstant sind, d. h. für alle Chipkarten gleich und bereits während der Kartenpersonalisierung definiert, weiß der Kartenleser **24** eindeutig, welche Datei den Authentisierungsschlüssel $K_{PRIV\_AUT\_CLIENT}$ enthält und welche den Signaturschlüssel $K_{PRIV\_SIG\_CLIENT}$. Auf welche Datei zuzugreifen ist, d. h. welcher Schlüssel zum Signieren zu verwenden ist, kann mittels der Befehls-APDU MANAGE SECURITY ENVIRONMENT gewählt werden. Diese Befehls-APDU ist ein PKCS#15-Befehl,

der vom Kartenleser **24** erkannt wird und den Kartenleser **24** veranlasst, die in **Fig.** 7 dargestellten Schritte auszuführen.

[0121] Nach Empfang der Befehls-APDU MANAGE SECURITY ENVIRONMENT in Schritt **700** löscht der Kartenleser **24** in Schritt **702** seinen Vorzeichenpuffer und prüft in Schritt **704**, ob das Dateikennzeichen (File IDentifier – FID) in der Befehls-APDU MANAGE SECURITY ENVIRONMENT der String „AUTH" ist oder nicht. Ist FID gleich „AUTH", setzt der Kartenleser **24** den aktuell gewählten PKCS#15-Schlüssel in Schritt **706** auf „AUTH". Andernfalls wird der aktuell gewählte PKCS#15-Schlüssel in Schritt **708** auf „SIG" gesetzt. Ungeachtet der Einstellung des aktuell gewählten PKCS#15-Schlüssels geht das Verfahren zu Schritt **710** weiter, wo die Befehls-APDU MANAGE SECURITY ENVIRONMENT zur Chipkarte **26** weitergeleitet und die von der Chipkarte **26** empfangene resultierende Antwort-APDU an den Client **22** zurückgeschickt wird.

[0122] Falls dem Befehl MANAGE SECURITY ENVIRONMENT PKCS#15-Befehle folgen, merkt sich der Kartenleser **24** stets das FID des letzten an die Chipkarte **26** geschickten Befehls MANAGE SECURITY ENVIRONMENT. Dadurch weiß der Leser stets, welcher Schlüssel verwendet wird und stellt sicher, dass der Authentisierungsschlüssel $K_{PRIV\_AUT\_CLIENT}$ niemals verwendet wird, ohne dass der Benutzer vorher um Genehmigung gefragt wurde.

[0123] Für die Verwendung des Schlüssels zur Benutzerauthentisierung $K_{PRIV\_AUT\_CLIENT}$ wird eine besondere Regel aufgestellt. Diese Regel besagt, dass es während einer vorgegebenen Zeitspanne, nachdem der Benutzer die Verwendung des Schlüssels gegenüber dem Kartenleser **24** genehmigt hat, möglich ist, $K_{PRIV\_AUT\_CLIENT}$ eine vordefinierte Anzahl von Malen ohne zusätzliche Genehmigungsanfrage zu verwenden. Der Kartenleser **24** setzt diese Regel mittels eines internen Zeitgebers durch. Die Zeitspanne, die vom internen Zeitgeber überwacht wird, sollte so gewählt werden, dass generell der vollständige Prozess der Client-Authentisierung einschließlich zweier Signaturen mit $K_{PRIV\_AUT\_CLIENT}$ und einer einzigen Signatur mit K_{PRIV_READER} ausgeführt werden kann.

[0124] Jegliches Signieren wird durch eine Befehls-APDU COMPUTE DIGITAL SIGNATURE veranlasst. Im Flussdiagramm von **Fig.** 8 ist das Verhalten des Kartenlesers **24** als Reaktion auf die Befehls-APDU COMPUTE DIGITAL SIGNATURE dargestellt. Im Moment seien nur die Schritte betrachtet, die für die Genehmigung seitens des Benutzers in Zusammenhang mit der Benutzerauthentisierung relevant sind, obwohl diese Befehls-APDU allgemein zum Signieren gegebener Eingangsdaten unter Verwendung jedes auf der Chipkarte **26** gespeicherten privaten Schlüssels verwendet werden kann.

[0125] Nach Empfang der Befehls-APDU COMPUTE DIGITAL SIGNATURE in Schritt **800** prüft der Kartenleser **24** in Schritt **802**, ob der derzeit gewählte PKCS#15-Schlüssel der Authentisierungsschlüssel $K_{PRIV\_AUT\_CLIENT}$ der Chipkarte **26** ist. Ist dies der Fall, geht das Verfahren zu Schritt **804** weiter. In Schritt **804** fragt der Kartenleser **24** seinen internen Zeitgeber hinsichtlich eines Zeitablaufs ab. Da der Zeitgeber zunächst auf 0 gesetzt ist (siehe Schritt **406** in **Fig.** 4), zeigt der Kartenleser **24** normalerweise eine Meldung auf seiner Anzeige **38** an, welche den Benutzer um Authentisierungsgenehmigung bittet (Schritt **806**).

[0126] Der Kartenleser **24** bestimmt dann in Schritt **808**, ob der Benutzer durch entsprechende Steuerung des Kartenlesers **24** die Genehmigung erteilt oder nicht. Diese Steuerung kann beispielsweise durch Drücken einer bestimmten Tasten des Tastenblocks **36** erfolgen. Gibt der Benutzer die Genehmigung nicht, wird in Schritt 810 ein entsprechendes Fehlerstatuswort an den Client **22** geschickt. Andernfalls wird in Schritt **812** noch einmal geprüft, ob der aktuell gewählte PKCS#15-Schlüssel $K_{PRIV\_AUT\_CLIENT}$ ist. Ist dies der Fall, wird der interne Zeitgeber des Kartenlesers **24** auf 30s eingestellt und gestartet (Schritt **814**). Der Kartenleser **24** geht dann zu Schritt **816** weiter und bildet die entsprechende Befehls-APDU, die anschließend an die Chipkarte **26** geschickt wird. Außerdem löscht der Kartenleser **24** seinen Vorzeichenpuffer.

[0127] Wie aus den folgenden Abschnitten deutlich werden wird, erfordert einvollständiger Benutzerauthentisierungszyklus, dass die Schritte **800** bis **816** zweimal durchgeführt werden. Im ersten Benutzerauthentisierungszyklus wird der Zeitgeber auf 30s eingestellt und in Schritt 814 gestartet. Im zweiten Benutzerauthentisierungszyklus bestimmt der Kartenleser **24** in Schritt **804**, ob der in Schritt **814** gestartete Zeitgeber bereits abgelaufen ist. Ist dies der Fall, wird vom Benutzer in Schritt 806 erneut eine Genehmigung angefordert. Andernfalls geht der Kartenleser **24** sofort zu Schritt 816 weiter und bildet die Befehls-APDU, die an die Chipkarte **26** zu schicken ist. Dies bedeutet, dass bei Durchführen zweier Benutzerauthentisierungsschritte innerhalb von 30s der Benutzer nur einmal, nämlich zu Beginn des ersten Authentisierungsschrittes, zur Genehmigung aufgefordert wird. Andernfalls muss der Benutzer jeden der beiden Benutzerauthentisierungsschritte getrennt genehmigen. Dies verbessert die Sicherheit der Authentisierung und verhindert den Missbrauch von $K_{PRIV\_AUT\_CLIENT}$

2.4.2 Erster Benutzerauthentisierungsschritt: SSL-Client-Authentisierung

[0128] Sobald das im Browser **48** laufende Java-Applet **50** (siehe **Fig.** 2) das PKCS#15-Token wie in **Fig.** 5 dargestellt erfolgreich geöffnet hat, leitet es den Client **22** von der Eingangs-Web-Seite zu einer sicheren Banking-Web-Seite. Wie in Absatz 2.1 erwähnt worden ist, ist bisher nur die Authentisierung des Server erfolgt, während der verschlüsselte Kanal

eingerichtet wurde. Die sichere Banking-Web-Seite erfordert nun zusätzlich die Authentisierung des Benutzers. Die Authentisierung des Benutzers wird durchgeführt, um den zuvor eingerichteten verschlüsselten Kanal zu einem gegenseitig authentisierten verschlüsselten Kanal wandeln.

[0129] Es sollte jedoch beachtet werden, dass ein authentisierter Kanal auch dann eingerichtet werden könnte, wenn bisher noch kein verschlüsselter Kanal verfügbar ist. Wie oben erwähnt wurde, wird in der beispielhaften Ausführungsform die SSL-Authentisierung des Client über den verschlüsselten Kanal durchgeführt, weil das Benutzerzertifikat C_{AUT_CLIENT}, das während der SSL-Authentisierung des Client an die Server-Infrastruktur **16** geschickt wird, Benutzerdaten enthält, die dem Bankgeheimnis unterliegen. Falls das Zertifikat anonymisierte Benutzerdaten enthält, könnte die Benutzerauthentisierung zumindest teilweise unverschlüsselt erfolgen.

[0130] Die sichere Banking-Web-Seite erfordert das SSL-Protokoll mit Benutzerauthentisierung. Die SSL-Authentisierung wird von der Server-Infrastruktur **16** („Hello" des Server) angestoßen und weist im Wesentlichen die in Absatz 2.1 in Zusammenhang mit dem Einrichten des verschlüsselten Kanals erörterten SSL-Schritte auf. Es ist zu beachten, dass während der SSL-Authentisierung bestimmte Parameter wie die Sitzungsschlüssel erneut initialisiert werden. Dies bedeutet beispielsweise, dass die für einen neu eingerichteten sicheren Kommunikationskanal verwendeten symmetrischen Schlüssel von den für den verschlüsselten Kanal verwendeten symmetrischen Schlüsseln verschieden sind.

[0131] Während der SSL-Authentisierung schickt der Browser **48** eine Befehls-APDU COMPUTE DIGITAL SIGNATURE an die Chipkarte **26**, um die Chipkarte **26** zu veranlassen, einen Hash-Wert mit dem Authentisierungsschlüssel $K_{PRIV\_AUT\_CLIENT}$ zu signieren. Der Kartenleser **24** erkennt die Befehls-APDU COMPUTE DIGITAL SIGNATURE und überprüft den internen Zeitgeber hinsichtlich der Verwendung des Authentisierungsschlüssels wie unter Bezugnahme auf das Flussdiagramm von **Fig.** 8 (Schritte **800** bis **816**) beschrieben worden ist. In Schritt **816** wird die PKCS#15-Anwendung **30** auf der Chipkarte **26** angewiesen, einen Hash-Wert mit $K_{PRIV\_AUT\_CLIENT}$ zu signieren. Wird die Signatur von der Chipkarte **26** an den Kartenleser **24** zurückgeschickt, prüft der Kartenleser in Schritt **818** einen zum Befehl COMPUTE DIGITAL SIGNATURE gehörigen LE-Code, um festzustellen, ob die von der Chipkarte **26** zurückgeschickte Signatur vom Kartenleser **24** zu signieren ist. Im Falle des ersten Benutzerauthentisierungsschrittes ist dies nicht erforderlich (LE ≠ 00) und der Kartenleser **24** schickt den mit $K_{PRIV\_AUT\_CLIENT}$ signierten Hash-Wert in Schritt **820** ohne weitere Maßnahmen zu ergreifen an den Client **22** zurück.

[0132] Der erste Benutzerauthentisierungsschritt wird nunmehr detalllierter unter Bezugnahme auf **Fig.** 9 beschrieben. **Fig.** 9 gibt einen Überblick über den vollständigen Prozess der Benutzerauthentisierung. Bei der beispielhaften Ausführungsform erfolgt der Prozess der Benutzerauthentisierung vollständig über einen verschlüsselten Kanal.

[0133] Der erste Benutzerauthentisierungsschritt beginnt wie mit einem Doppelpfeil **62** zwischen dem Client **22** und dem Proxy-Server **56** gekennzeichnet. Der erste Benutzerauthentisierungsschritt wird auf der SSL-Ebene unter Verwendung des auf der Chipkarte **26** gespeicherten Authentisierungsschlüssels $K_{PRIV\_AUT\_CLIENT}$ und eines Challenge, der u. a. vom von der Server-Infrastruktur **16** bereitgestellten Schlüssel für Verschlüsselungszwecke K_{EP} abgeleitet wird, durchgeführt. Da bei der in Zusammenhang mit **Fig.** 9 beschriebenen Ausführungsform der erste Benutzerauthentisierungsschritt von der Eingangsstelle gesteuert wird, ist der Schlüssel für Verschlüsselungszwecke K_{EP} der Kryptographie-Schlüssel der Eingangsstelle, d. h. ein Schlüssel für Verschlüsselungszwecke, der z. B. vom Proxy-Server **56** verwaltet wird. Der (öffentliche) Schlüssel für Verschlüsselungszwecke K_{EP} ist Bestandteil des Zertifikats, während der entsprechende private Schlüssel aus Sicherheitsgründen auf einem getrennten Hardwaremodul gespeichert wird, auf das der Proxy-Server **56** zugreifen kann.

[0134] Nach einem erfolgreichen ersten Benutzerauthentisierungsschritt ist ein sicherer Kommunikationskanal **64** eingerichtet. Bei der Ausführungsform gemäß **Fig.** 9 ist der sichere Kommunikationskanal **64** eine gegenseitig authentisierte end-to-end Verbindung zwischen dem Client **22** und einer Eingangsstelle des Intranet **52** in Form der den Proxy-Server **56** beinhaltenden DMZ **54**. Der sichere Kommunikationskanal **64** erstreckt sich über den zuvor eingerichteten verschlüsselten Kanal. Die weitere Kommunikation zwischen beliebigen Komponenten des Intranet **52**, z. B. dem Anwendungsserver **58**, und dem Client **22** erfolgt über diesen sicheren Kommunikationskanal **64**. Die weitere Kommunikation weist insbesondere einen zweiten Benutzerauthentisierungsschritt und die Übertragung signierter Transaktionsdaten auf.

2.4.3 Zweiter Benutzerauthentisierungsschritt: Authentisierung auf der Anwendungsschicht

[0135] Nach der erfolgreichen SSL-Authentisierung (erster Benutzerauthentisierungsschritt) veranlasst der Anwendungsserver **58** einen zusätzlichen zweiten Authentisierungsschritt, um sowohl die Chipkarte **26** als auch den Kartenleser **24** zu authentisieren. Zu diesem Zweck schickt der Anwendungsserver **58** über den sicheren Kommunikationskanal **64** einen zufälligen Challenge an den Client **22** und fordert eine doppelte Signatur an. Dies bedeutet, dass der Challenge mittels des Authentisierungsschlüssels $K_{PRIV\_AUT\_CLIENT}$ der Chipkarte **26** zu signieren ist, und die resultierende Signatur dann vom Kartenleser **24** mittels des Authentisierungsschlüssels K_{READER} des

Kartenlesers **24** zu signieren ist.

[0136] Vom Kartenleser **24** aus gesehen ist der zweite Benutzerauthentisierungsschritt ähnlich dem oben unter Bezugnahme auf **Fig.** 8 (Schritte **88** bis **816**) beschriebenen ersten Benutzerauthentisierungsschritt. Der einzige Unterschied besteht darin, dass eine doppelte Signatur erforderlich ist. Diese Anforderung wird durch den zur Befehls-APDU COMPUTE DIGITAL SIGNATURE gehörigen LE-Code angegeben. Ist der LE-Code auf 00 gesetzt, weiß der Kartenleser **24**, dass er von der Chipkarte **26** empfangene Daten zu signieren hat. Stellt also der Kartenleser **24** in Schritt **818** fest, dass der LE-Code gleich 00 ist, signiert er die von der Chipkarte **26** empfangene Signatur in Schritt **822** mit seinem eigenen Authentisierungsschlüssel K_{READER} und schickt in Schritt **824** die doppelte Signatur zum Client **22** zurück.

[0137] Der zweite Authentisierungsschritt wird nunmehr in Zusammenhang mit der in **Fig.** 9 dargestellten Übersicht beschrieben.

[0138] Nachdem der sichere Kommunikationskanal **64** während des ersten Authentisierungsschrittes eingerichtet worden ist, sendet der Anwendungsserver **58** seinen Challenge über den sicheren Kommunikationskanal **64** an den Client **22**. Der Client **22** veranlasst, dass dieser Challenge sowohl von der Chipkarte **26** als auch vom Kartenleser **24** signiert wird und schickt den signierten Challenge, wie durch Pfeil **66** gekennzeichnet, an den Anwendungsserver **58** zurück.

[0139] Der Anwendungsserver **58** prüft dann auf einer Anwendungsebene unter Verwendung der öffentlichen Schlüssel sowohl der Chipkarte **26** als auch des Kartenlesers **24** die Authentizität der Signaturen. Diese öffentlichen Schlüssel sind Bestandteil des Zertifikats C_{READER} des Kartenlesers **24** und des Zertifikats C_{AUT_CLIENT} der Chipkarte **26**. Eine derartige Prüfung erfordert natürlich, dass die beiden Zertifikate dem Anwendungsserver **58** vor dem zweiten Benutzerauthentisierungsschritt bekannt sind. Die beiden Zertifikate können beispielsweise über den verschlüsselten sicheren Kommunikationskanal **64** übertragen worden sein. Des Weiteren können die Zertifikate von dem Server **60** mit der Funktionalität einer Zertifizierungsstelle erzeugt worden sein, der Bestandteil des Intranet **52** ist (siehe **Fig.** 3).

[0140] Um die Sicherheit der Authentisierung zu erhöhen, weist der zweite Benutzerauthentisierungsschritt außerdem wie durch den Doppelpfeil **68** gekennzeichnet eine Gleichheitsprüfung auf. Während dieser Gleichheitsprüfung wird festgestellt, ob zur Benutzerauthentisierung ein und derselbe Schlüssel $K_{PRIV\_AUT\_CLIENT}$ während des ersten und zweiten Benutzerauthentisierungsschrittes verwendet worden ist. Zu diesem Zweck speichert der Proxy-Server **56** oder eine andere Komponente der Eingangsstelle des Intranet **52** vorübergehend den während des ersten Benutzerauthentisierungsschrittes verwendeten Authentisierungsschlüssel $K_{PRIV\_AUT\_CLIENT}$ der Chipkarte **26**. Die Gleichheitsprüfung wird vom Anwendungsserver **58** veranlasst.

2.5 Signieren von Transaktionen

[0141] Immer wenn eine Bank-Transaktion von der Chipkarte **26** signiert werden muss, schickt das im Browser **48** laufende Java-Applet **50** eine Meldung beinhaltend Informationen über die gewünschte Finanztransaktion an die Chipkarte **26** zum Signieren mit dem Signaturschlüssel $K_{PRIV\_SIG\_CLIENT}$ Diese an die Chipkarte **26** geschickte Meldung geht einher mit der Befehls-APDU COMPUTE DIGITAL SIGNATURE. Ist die letzte Signatur mit dem Authentisierungsschlüssel $K_{PRIV\_AUT\_CLIENT}$ der Chipkarte **26** erzeugt worden, wird eine in Zusammenhang mit **Fig.** 7 beschriebene Befehls-APDU MANAGE SECURITY ENVIRONMENT (FID ≠ AUTH) an die Chipkarte **26** geschickt, um vor der Befehls-APDU COMPUTE DIGITAL SIGNATURE den zutreffenden Schlüssel zu setzen.

[0142] Wenn und nur wenn der Signaturschlüssel $K_{PRIV\_SIG\_CLIENT}$ in Verwendung ist, ermöglicht der Kartenleser **24** eine Befehlsverkettung, um das Signieren von Daten beliebiger Länge zu gestatten. Im Falle einer Befehlsverkettung schickt der Kartenleser **24** keinen Befehl an die Chipkarte **26**, sondern puffert die Daten nur intern. Ein Byte einer spezifischen Klasse von 00_{HEX} gibt den letzten (oder einzigen) Befehl an (Schritte **826** und **828** in **Fig.** 8). Bestimmt der Kartenleser in Schritt **826**, dass die letzte Befehls-APDU empfangen wird, zeigt er auf seiner Anzeige **38** die Meldung (z. B. „Überweisung 100.000 CHF") als Aufforderung zur Benutzergenehmigung an (Schritt **830** und **806**). Die nachfolgenden Schritte entsprechen den in **Fig.** 8 dargestellten Schritten, die in Zusammenhang mit der Benutzerauthentisierung beschrieben wurden. Der einzige Unterschied besteht darin, dass dann, wenn der Benutzer die Transaktion genehmigt, der Kartenleser **24** einen Hash-Wert (SHA-1) über die Meldung berechnet und die einem Hashing unterzogene Meldung an die Chipkarte **26** zum Signieren sendet.

[0143] Falls eine Bank-Transaktion signiert werden muss, ist die Befehls-APDU COMPUTE DIGITAL SIGNATURE mit einem LE-Code **00** verbunden, der die Anforderung einer doppelten Signatur bedeutet (Schritte **818**, **822**, **824**). Um die zweite Signatur zu erzeugen, berechnet der Kartenleser **24** einen Hash-Wert (SHA-1) über die von der Chipkarte **26** empfangenen Signatur, fügt PKCS#11 (z. B. Blocktyp 1)-Blindgruppen (Padding) hinzu und führt schließlich mit K_{READER} das Signieren mittels des privaten RSA-Schlüssels durch. Letzteres stellt explizit sicher, dass die Chipkarte **26** in einem echten Kartenleser **24** betrieben wird und verknüpft kryptographisch die Authentisierung des Kartenlesers mit einer Transaktion auf Chipkartenbasis. Dadurch kann der Anwendungsserver **58** nicht nur identifizieren, welcher Kartenleser **24** in Verwendung ist, sondern der Anwen-

dungsserver **58** kann auch Signaturen von bestimmten Kartenlesern **24** zurückweisen z. B. im Falle einer Annullierung des Leserzertifikats.

[0144] Die signierte Datenübertragung in Zusammenhang mit einer Bank-Transaktion ist durch Pfeil **70** in **Fig.** 9 gekennzeichnet. Wie aus **Fig.** 9 deutlich wird, wird die signierte Datenübertragung **70** über den sicheren Kommunikationskanal **64** durchgeführt, der bei der Ausführungsform auf dem verschlüsselten Kanal eingerichtet wurde. Sowohl der Signaturschlüssel $K_{PRIV\_SIG\_CLIENT}$ der Chipkarte **26** als auch der Authentisierungsschlüssel K_{READER} des Kartenlesers **24** werden verwendet, wenn eine signierte Datenübertragung durchgeführt wird. Aus **Fig.** 9 ist des Weiteren ersichtlich, dass doppelte Signaturen (Pfeile **66** und **70**) nur (auf der Anwendungsschicht) vom Anwendungsserver **58** verwendet werden.

[0145] Die Interaktion zwischen der PKCS#11-Schicht **44**, dem Kartenleser **24** und der PKCS#15-Anwendung **30** auf der Chipkarte **26** bei einer zu erzeugenden doppelten Signatur wird nunmehr detalllierter unter Bezugnahme auf **Fig.** 10 beschrieben.

[0146] Erhält die PKCS#11-Schicht **44** eine Aufforderung, die eine bestimmte, mit einer doppelten Signatur zu signierende Meldung (Daten) betrifft (Schritt **1002**), erzeugt sie eine entsprechende Befehls-APDU COMPUTE DIGITAL SIGNATURE bezüglich dieser Meldung und schickt sie in Schritt **1004** an den Kartenleser **24**. Der Kartenleser **24** erkennt diese Befehls-APDU und ergreift in Schritt **1006** zusätzliche Maßnahmen. Diese zusätzlichen Maßnahmen können das Anzeigen der zu signierenden Daten und eine Aufforderung zur Genehmigung seitens des Benutzers aufweisen.

[0147] Der Kartenleser **24** leitet dann die Befehls-APDU COMPUTE DIGITAL SIGNATURE zusammen mit den zu signierenden (einem Hashing unterzogenen) Daten an die PKCS#15-Anwendung **30** auf der Chipkarte **26** weiter. Die PKCS#15-Anwendung **30** signiert die vom Kartenleser **24** erhaltenen Daten und schickt in Schritt **1010** eine Antwort-APDU mit den signierten Daten (CSIG) an den Kartenleser **24** zurück. In Schritt **1012** berechnet der Kartenleser **24** die Lesersignatur (RSIG) über CSIG wie in **Fig.** 10 angegeben. In Schritt **1014** schickt der Kartenleser **24** die doppelte Signatur an die PKCS#11-Schicht **44**, die sie an den Browser **48** weiterleitet, von wo sie zur Verifizierung an den Anwendungsserver **58** geschickt wird.

[0148] Ist die Bank-Transaktion abgeschlossen und wählt der Benutzer die Abmeldung von der Banking-Sitrung, wird die Chipkarte **26** zurückgesetzt, um die PIN ungültig zu machen. Andere Ereignisse wie unerwartete Fehler der Chipkarte können die Banking-ing-Sitzung ebenfalls beenden. Die die Banking-Sitrung beendenden Ereignisse sind auf Anwendungsebene definiert und erfordern keine weitere Funktionalität des Kartenlesers.

2.6 Fernmanagement der Chipkarte

[0149] Das Fernmanagement der Chipkarte, ein Aspekt der Erfindung, der in der obigen Beschreibung mehrmals kurz erwähnt worden ist, ist ein Merkmal, das der entweder in Kombination mit dem Verfahren der Benutzerauthentisierung oder davon getrennt verwirklicht werden kann. Zu Zwecken des Fernmanagements der Chipkarte ist ein (sicherer) end-to-end Managementkanal zwischen der Chipkarte **26** auf der einen Seite und der Server-Infrastruktur **16** auf der anderen Seite einzurichten. Der Managementkanal kann auf Basis des verschlüsselten Kanals, der zu Beginn einer Verbindung zur Server-Infrastruktur **16** eingerichtet wird (siehe Absatz 2.4.1), oder auf Basis des sicheren Kommunikationskanals **64**, der während des ersten Benutzerauthentisierungsschrittes eingerichtet wird, oder auf Basis beider Kanäle eingerichtet werden.

[0150] In **Fig.** 11 ist eine Übersicht der am Fernmanagement der Chipkante beteiligten Komponenten dargestellt. Bei der in **Fig.** 11 dargestellten Ausführungsform wird der durch den Doppelpfeil **72** gekennzeichnete Managementkanal auf Basis des sicheren Kommunikationskanals **64** eingerichtet, der während des ersten Benutzerauthentisierungsschrittes eingerichtet worden ist. Wie aus **Fig.** 11 deutlich wird, erkennt der Kartenleser **24** Managementbefehle für die Chipkarte und leitet sie transparent an die Chipkarte **26** weiter. Dieser Managementkanal **72** umgeht den Kartenleser **24**, da die Managementbefehle dem Kartenleser **24** unbekannt sind und deshalb transparent an die Chipkarte **26** weitergeleitet werden. Dies wird durch eine spezielle Erweiterung in der PKCS#11-Bibliothek möglich.

[0151] Die Hauptaspekte des Fernmanagements der Chipkarte sind das PKCS#15-Sicherheitsmanagement und das OP-Chipkartenmanagement. Im Folgenden wird nur das PKCS#15-Sicherheitsmanagement näher betrachtet. Das OP-Chipkartenmanagement enthält im Wesentlichen das PKC5#15-Sicherheitsmanagement und charakteristische Funktionalitäten, die Aspekte wie das Laden von Anwendungen nach der Kartenausgabe oder die Kartenprüfung (Auditing) betreffen.

[0152] Das PKCS#15-Sicherheitsmanagement wird nunmehr detaillierter unter Bezugnahme auf **Fig.** 11 erläutert. Wie aus **Fig.** 11 ersichtlich ist, wird der Managementkanal **72** zwischen der Chipkarte **26** und dem CA-Server **60** des Intranet **52** durch den Proxy-Server **56** in der DMZ **54** hindurch unter Umgehung des Anwendungsservers **58** eingerichtet. Der Managementkanal **72** wird über den auf PKCS#15 basierenden sicheren Managementkanal eingerichtet. Die Verschlüsselung des Managementkanals erfolgt unter Verwendung des chipkartenspezifischen Triple DES-Schlüssels K_{EA}. Die PKCS#15-Anwendung auf der Chipkarte **26** stellt eine zweite PIN bereit, die nur über der sicheren (verschlüsselten) Managementkanal **72** verifiziert werden kann. Bei der

zweiten PIN handelt es sich um eine dem Herausgeber der Chipkarte zugewiesene PIN. Alle Schreibzugriffsbedingungen der PKCS#15-Dateien auf der Chipkarte **26** sind an diese PIN gebunden.

[0153] Sobald der sichere Managementkanal eingerichtet und die PIN des Ausgebers verifiziert worden ist, kann der CA-Server **60** Dateien modifizieren oder neue Dateien auf der Chipkarte **26** erzeugen und kann so beispielsweise ein oder mehrere Zertifikate aktualisieren oder einen oder mehrere Schlüssel hinzufügen. Der CA-Server **60** kann deshalb nicht nur die Schlüssel und Zertifikate auf der Chipkarte **26** bei deren Ausgabe erzeugen, sondern auch diese Berechtigungsnachweise (Credentials) nach der Ausgabe der Chipkarte verwalten.

[0154] Wie aus **Fig.** 11 ersichtlich ist, weist die Server-Infrastruktur **16** zusätzlich Certificate Revocation List (CRL)-Funktionalitäten auf. Zu diesem Zweck weist das Intranet **52** einen mit dem CA-Server **60**, dem Anwendungsserver **58** und dem Proxy-Server **56** kommunizierenden CRL-Server **74** auf. Der CRL-Server **74** verwaltet die Listen der Zertifikatannullierungen und sperrt eine Banking-Transaktion oder jede andere vom Client **22** angeforderte Operation, wenn er bestimmt, dass ein von der Client-Infrastruktur **12** bereitgestelltes Zertifikat annulliert worden ist.

2.7 Sichere E-Mall

[0155] Die oben erläuterte Authentisierungslösung kann auch für S/MIME eingesetzt werden, um dem Benutzer einen Mehrwert anzubieten. Das heißt, sobald eine Chipkarten-Infrastruktur aufgebaut worden ist, kann sie auch von Programmen wie Microsoft Outlook oder Netscape Messenger verwendet werden. Dies könnte es erforderlich machen, für die S/MIME-Verwendung ein weiteres Schlüsselpaar auf der Chipkarte **26** zu speichern. Alternativ könnte der Authentisierungsschlüssel $K_{PRIV\_AUT\_CLIENT}$ für S/MIME verwendet werden. Dies jedoch kann den Austausch des Authentisierungszertifikats C_{AUT_CLIENT} gegen ein Zertifikat erforderlich machen, das für S/MIME geeignet ist (nicht anonymes Zertifikat). Zu diesem Zweck könnte das im vorigen Absatz beschriebene Fernmanagement der Chipkarte verwendet werden.

[0156] Ist S/MIME zu unterstütren, muss die Software des Kartenlesers **26** außerdem eine Befehls-APDU DECIPHER unterstütren. Diese Befehls-APDU entschlüsselt Daten und könnte verwendet werden, wenn der Authentisierungsschlüssel $K_{PRIV\_AUT\_CLIENT}$ für S/MIME verwendet wird. Die Chipkarte **26** lässt die Entschlüsselung von Daten mit dem Signaturschlüssel $K_{PRIV\_SIG\_CLIENT}$ nicht zu. Der Befehl DECIPHER sollte vom Kartenleser **24** transparent an die Chipkarte **26** weitergeleitet werden, wobei er nur das Prinzip der Schlüsselverwendung, das wie in **Fig.** 12 dargestellt einen Genehmigungsprozess durch den Benutzer beinhaltet, durchsetzt. Da die einzelnen in **Fig.** 12 dargestellten Schritte im Wesentlichen mit denen von **Fig.** 8 übereinstimmen, wird auf eine detallliertere Beschreibung von **Fig.** 12 verzichtet.

II. Benutzer- und Datenauthentisierung mit demselben Signaturschlüssel

[0157] **Fig.** 13 zeigt eine Übersicht des gesamten Benutzer- und Datenauthentisierungsprozesses für den Fall, dass ein einziger Signaturschlüssel $K_{PRIV\_SIG\_CLIENT}$ der Chipkarte verwendet wird. **Fig.** 13 entspricht im Prinzip **Fig.** 9.

[0158] Wie aus **Fig.** 13 ersichtlich ist, wird nur ein einziger Benutzerauthentisierungsschritt **66** durchgeführt, um den gegenseitig authentisierten Kanal **64** einzurichten. Die signierte Datenübertragung in Zusammenhang mit einer Banking-Transaktion ist mit Pfeil **70** gekennzeichnet. Wie **Fig.** 13 zeigt, erfolgt die signierte Datenübertragung **70** über den Kanal **64**. Sowohl der Signaturschlüssel $K_{PRIV\_SIG\_CLIENT}$ der Chipkarte **26** als auch der Authentisierungsschlüssel K_{READER} des Kartenlesers **24** sind in Verwendung, wenn die signierte Datenübertragung **70** durchgeführt wird.

[0159] Die in **Fig.** 13 dargestellte Ausführungsform entspricht grundsätzlich der oben in Zusammenhang mit der Chipkarte, die zwei spezielle Signaturschlüssel aufweist, beschriebenen Ausführungsform. Im Prinzip wird die Verwendung von $K_{PRIV\_AUT\_CLIENT}$ einfach durch die Verwendung von $K_{PRIV\_SIG\_CLIENT}$ ersetzt. Folglich kann eine detallliertere Erläuterung von **Fig.** 13 entfallen. Allerdings wird ein Hauptunterschied zwischen beiden Ausführungsformen kurz besprochen.

[0160] Dieser Unterschied betrifft die Tatsache, dass im Gegensatz zur ersten Ausführungsform der während des Benutzerauthentisierungsschrittes zu signierende Challenge (z, B. ein geeigneter Zufallswert) in einem Authentisierungskontext vollständig auf der Anzeige **38** des Kartenlesers **24** angezeigt wird. Der Authentisierungskontext ist eine Textmeldung, die den Benutzer informiert, dass der angezeigte Challenge zu Anmeldungs- und Authentisierungszwecken zu verwenden ist. Des Weiteren fordert die Textmeldung zur Genehmigung der Anmeldung (und Authentisierung) auf. Wie oben erwähnt, wird in der ersten Ausführungsform nur diese Aufforderung während der Anmeldung/Benut-zerauthentisierung angezeigt.

[0161] Gibt der Benutzer die Genehmigung, schickt der Kartenleser **24** den Challenge zusammen mit der Textmeldung zum Signieren an die Chipkarte **26**. Die Chipkarte **26** signiert sowohl den Challenge als auch die Textmeldung mit $K_{PRIV\_SIG\_CLIENT}$ und schickt die Signatur zum Kartenleser **24** zurück. Der Kartenleser **24** signiert dann die Signatur und leitet die doppelte Signatur, falls erforderlich mit zusätzlichen Daten, an die Server-Infrastruktur **16** weiter.

[0162] Nach der erfolgreichen Benutzerauthentisierung erfolgt die Signierung der Transaktion unter Verwendung von $K_{PRIV\_SIG\_CLIENT}$ wie in Zusammenhang

mit der ersten Ausführungsform beschrieben. Während der Signierung der Transaktion werden doppelte Signaturen ($K_{PRIV\_SIG\_CLIENT}$, K_{READER}) verwendet.

Patentansprüche

1. Verfahren zum Durchführen einer Benutzer- und Datenauthentisierung über einen Client (**22**), der über ein erstes Netzwerk (**14**) mit einer Server-Infrastruktur (**16**) kommuniziert, wobei der Client (**22**) über einen seitens des Benutzers steuerbaren Kartenleser (**24**) Zugriff auf eine Chipkarte (**26**) hat, auf der mindestens ein Signaturschlüssel, $K_{PRIV\_AUT\_CLIENT}$ und $K_{PRIV\_SIG\_CLIENT}$ gespeichert ist, wobei das Verfahren folgende Schritte umfasst:
– Durchführen eines Benutzerauthentisierungsschrittes, wobei der Benutzerauthentisierungsschritt Folgendes enthält: Anzeigen eines Authentisierungskontextes durch den Kartenleser (**24**), Steuern des Kartenlesers (**24**), um vom Benutzer die Genehmigung der Signatur anzufordern, und, im Falle der Genehmigung der Signatur, Übermitteln eines Challenge, falls angebracht zusammen mit Kontextdaten, oder davon abgeleiteter Daten an die Chipkarte (**26**) zum Signieren;
– Durchführen eines Datenauthentisierungsschrittes, wobei der Datenauthentisierungsschritt Folgendes enthält: Anzeigen der zu authentisierenden Daten durch den Kartenleser (**24**), Steuern des Kartenlesers (**24**), um vom Benutzer die Genehmigung der Signatur anzufordern, und, im Falle der Genehmigung der Signatur, Übermitteln der zu authentisierenden oder davon abgeleiteter Daten an die Chipkarte (**26**) zum Signieren.

2. Verfahren nach Anspruch 1, bei dem während des Benutzer- und Datenauthentisierungsschrittes derselbe Schlüssel $K_{PRIV\_SIG\_CLIENT}$, der auf der Chipkarte (**26**) gespeichert ist, zur Erzeugung der Signatur verwendet wird.

3. Verfahren nach Anspruch 1 oder 2, bei dem der Benutzerauthentisierungsschritt über einen verschlüsselten Kanal (**64**) erfolgt, der vor dem oder in Zusammenhang mit dem Schritt der Benutzerauthentisierung eingerichtet wird.

4. Verfahren nach einem der Ansprüche 1 bis 3, bei dem ein erster Authentisierungsschlüssel K_{READER} zum Kartenleser (**24**) gehört.

5. Verfahren nach Anspruch 4, das des Weiteren das Signieren der Signatur, die mit dem mindestens einen auf der Chipkarte (**26**) gespeicherten Signaturschlüssel, $K_{PRIV\_AUT\_CLIENT}$ und $K_{PRIV\_SIG\_CLIENT}$ erzeugt wurde, mit dem zum Kartenleser (**24**) gehörenden Authentisierungsschlüssel K_{READER} und das Übertragen der beiden Signaturen, erforderlichenfalls zusammen mit dem Challenge, den Kontextdaten und/oder den zu authentisierenden Daten, an die Server-Infrastruktur (**16**) enthält.

6. Verfahren nach einem der Ansprüche 1 bis 5, bei dem ein zweiter Authentisierungsschlüssel $K_{PRIV\_AUT\_CLIENT}$ auf der Chipkarte (**26**) gespeichert ist.

7. Verfahren nach Anspruch 6, bei dem während des Benutzerauthentisierungsschrittes der zweite Authentisierungsschlüssel $K_{PRIV\_AUT\_CLIENT}$ zum Signieren verwendet wird, und bei dem während des Datenauthentisierungsschrittes der Signaturschlüssel $K_{PRIV\_SIG\_CLIENT}$ zum Signieren verwendet wird.

8. Verfahren nach Anspruch 6 oder 7, bei dem der Benutzerauthentisierungsschrittes einen ersten Schritt der Benutzerauthentisierung enthält, an dem der verschlüsselte Kanal und der zweite Authentisierungsschlüssel $K_{PRIV\_AUT\_CLIENT}$ beteiligt sind, und einen zweiten Schritt der Benutzerauthentisierung, an dem mindestens der erste und der zweiten Authentisierungsschlüssel, K_{READER}, $K_{PRIV\_AUT\_CLIENT}$ beteiligt sind.

9. Verfahren nach Anspruch 8, bei dem eine Abhängigkeit des ersten Schrittes der Benutzerauthentisierung von einem Schlüssel für Verschlüsselungszwecke K_{EP} der Server-Infrastruktur (**16**) eingeführt wird.

10. Verfahren nach einem der Ansprüche 1 bis 8, bei dem die Server-Infrastruktur (**16**) des Weiteren Folgendes enthält:
– ein zweites Netzwerk (**52**), vorzugsweise ein sicheres Intranet, in dem der Anwendungsserver (**58**) angeordnet ist, und
– eine Eingangsstelle (**54**, **56**) des zweiten Netzwerks (**52**), wobei die Eingangsstelle (**54**, **56**) mit dem ersten Netzwerk (**14**) gekoppelt ist.

11. Verfahren nach Anspruch 10, bei dem der erste Benutzerauthentisierungsschritt von der Eingangsstelle gesteuert wird.

12. Verfahren nach einem der Ansprüche 8 bis 11, bei dem der erste Benutzerauthentisierungsschritt auf einer Schicht unter der Anwendungsschicht und der zweite Benutzerauthentisierungsschritt auf der Anwendungsschicht durchgeführt wird.

13. Verfahren nach einem der Ansprüche 8 bis 12, bei dem der erste Benutzerauthentisierungsschritt einen ersten Unterschritt enthält, während dessen ein verschlüsselter Kanal eingerichtet wird, und einen zweiten Unterschritt, während dessen der verschlüsselte Kanal zur Übertragung von zur Benutzerauthentisierung erforderlichen Informationen verwendet wird.

14. Computerprogrammprodukt mit Programmcodeeinrichtungen zum Durchführen der Schritte ei-

nes der Ansprüche 1 bis 13, wenn das Computerprogrammprodukt auf einem Computersystem läuft.

15. Computerprogrammprodukt nach Anspruch 14, das auf einem computerlesbaren Aufzeichnungsmedium gespeichert ist.

16. Server-Infrastruktur (**16**) mit:
– einem Kommunikationskanal zwischen der Server-Infrastruktur (**16**) und einer Client-Infrastruktur (**12**) über ein erstes Netzwerk (**14**), wobei die Client-Infrastruktur (**12**) einen Client (**22**) enthält, der Zugriff auf einen seitens des Benutzers steuerbaren Kartenleser (**24**) mit einer Anzeige (**38**) hat;
– einer authentisierten Verbindung zwischen der Server-Infrastruktur (**16**) und der Client-Infrastruktur (**12**), wobei die authentisierte Verbindung eingerichtet wird, indem ein Authentisierungskontext auf der Anzeige (**38**) des Kartenlesers (**24**) angezeigt wird, der Kartenleser (**24**) so gesteuert wird, dass er den Benutzer zur Genehmigung der Signatur auffordert, und, im Falle der Genehmigung der Signatur, indem ein Challenge, erforderlichenfalls zusammen mit Kontextdaten, oder davon abgeleitete Daten an die Chipkarte (**26**) zum Signieren übertragen wird, und
– einer signierten Datenübertragung (**70**) zwischen der Client-Infrastruktur (**12**) und der Server-Infrastruktur (**16**), wobei die signierte Datenübertragung (**70**) eingerichtet wird, indem die zu authentisierenden Daten auf der Anzeige (**38**) angezeigt werden, der Kartenleser (**24**) so gesteuert wird, dass er den Benutzer zur Genehmigung der Signatur auffordert, und, im Falle der Genehmigung der Signatur, indem die zu authentisierenden Daten oder davon abgeleitete Daten an die Chipkarte (**26**) zum Signieren übertragen werden.

17. Server-Infrastruktur nach Anspruch 16, bei der das erste Netzwerk (**14**) das Internet oder ein unsicheres externes Netzwerk ist.

18. Server-Infrastruktur nach Anspruch 16 oder 17, bei der die Server-Infrastruktur (**16**) des Weiteren Folgendes enthält:
– ein zweites Netzwerk (**52**), vorzugsweise ein sicheres Inteanet, in dem der Anwendungsserver (**58**) angeordnet ist, und
– eine Eingangsstelle (**54**, **56**) des zweiten Netzwerks (**52**), wobei die Eingangsstelle (**54**, **56**) mit dem ersten Netzwerk (**14**) gekoppelt ist.

19. Server-Infrastruktur nach Anspruch 18, bei der die Eingangsstelle eine Proxy-Serverkomponente (**56**) umfasst und/oder auf dem Anwendungsserver (**58**) angeordnet ist.

20. Server-Infrastruktur nach Anspruch 18 oder 19, bei der das zweite Netzwerk (**52**) die Funktionalität einer Zertifizierungsstelle (**60**) umfasst.

21. Server-Infrastruktur nach Anspruch 20, die des Weiteren einen sicheren end-to-end Managementkanal (**72**) zwischen dem die Zertifizierungsstelle (**60**) aufweisenden zweiten Netzwerk (**52**) und der Chipkarte (**26**) umfasst.

22. Netzwerksystem mit:
– einer Client-Infrastruktur (**12**) mit einem Client (**22**), der mit einem Kartenleser (**24**) für eine Chipkarte (**26**) assoziiert ist, wobei die Chipkarte (**26**) einen ersten sicheren Speicherplatz zum Speichern mindestens eines ersten Signaturschlüssels, $K_{PRIV\_AUT\_CLIENT}$ und $K_{PRIV\_SIG\_CLIENT}$ besitzt und der Kartenleser (**24**) eine Anzeige (**38**) besitzt;
– einer authentisierten Verbindung zwischen der Server-Infrastruktur (**16**) und der Client-Infrastruktur (**12**), wobei die authentisierte Verbindung eingerichtet wird, indem ein Authentisierungskontext auf der Anzeige (**38**) des Kartenlesers (**24**) angezeigt wird, der Kartenleser (**24**) so gesteuert wird, dass er den Benutzer zur Genehmigung der Signatur auffordert, und, im Falle der Genehmigung der Signatur, indem ein Challenge, falls angebracht zusammen mit Kontextdaten, oder davon abgeleitete Daten an die Chipkarte (**26**) zum Signieren übertragen wird, und
– einer signierten Datenübertragung (**70**) zwischen der Client-Infrastruktur (**12**) und der Server-Infrastruktur (**16**), wobei die signierte Datenübertragung (**70**) eingerichtet wird, indem die zu authentisierenden Daten auf der Anzeige (**38**) angezeigt werden, der Kartenleser (**24**) so gesteuert wird, dass er den Benutzer zur Genehmigung der Signatur auffordert, und, im Falle der Genehmigung der Signatur, indem die zu authentisierenden Daten oder davon abgeleitete Daten an die Chipkarte (**26**) zum Signieren übertragen werden.

23. Netzwerksystem nach Anspruch 22, bei dem der Kartenleser (**24**) einen zweiten sicheren Speicherplatz zum Speichern eines Authentisierungsschlüssels K_{READER} hat.

24. Netzwerksystem nach Anspruch 22 oder 23, bei dem der Client (**12**) einen Wrapper (**46**), z. B. PKCS#11, zum Einrichten einer Kommunikationsverbindung zwischen einer Clientanwendung (**48**) und einer bestimmten aus einer Vielzahl verschiedener Chipkarten (**26**) aufweist.

25. Netzwerksystem nach einem der Ansprüche 22 bis 24, bei dem der Kartenleser (**24**) mindestens ein Leser der Klasse **4** oder ein FINREAD-kompatibler Kartenleser ist.

Es folgen 13 Blatt Zeichnungen

Anhängende Zeichnungen

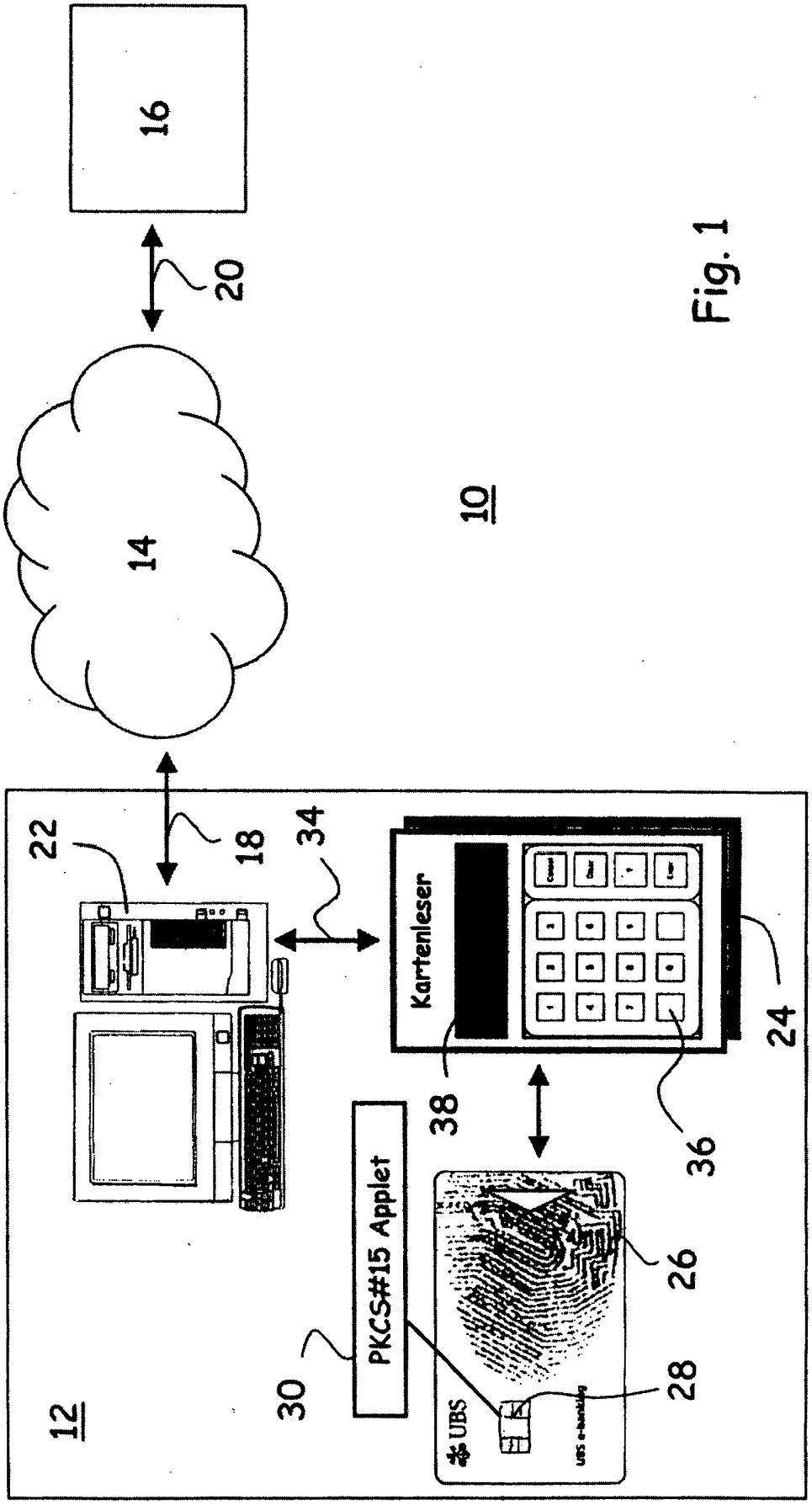

Fig. 1

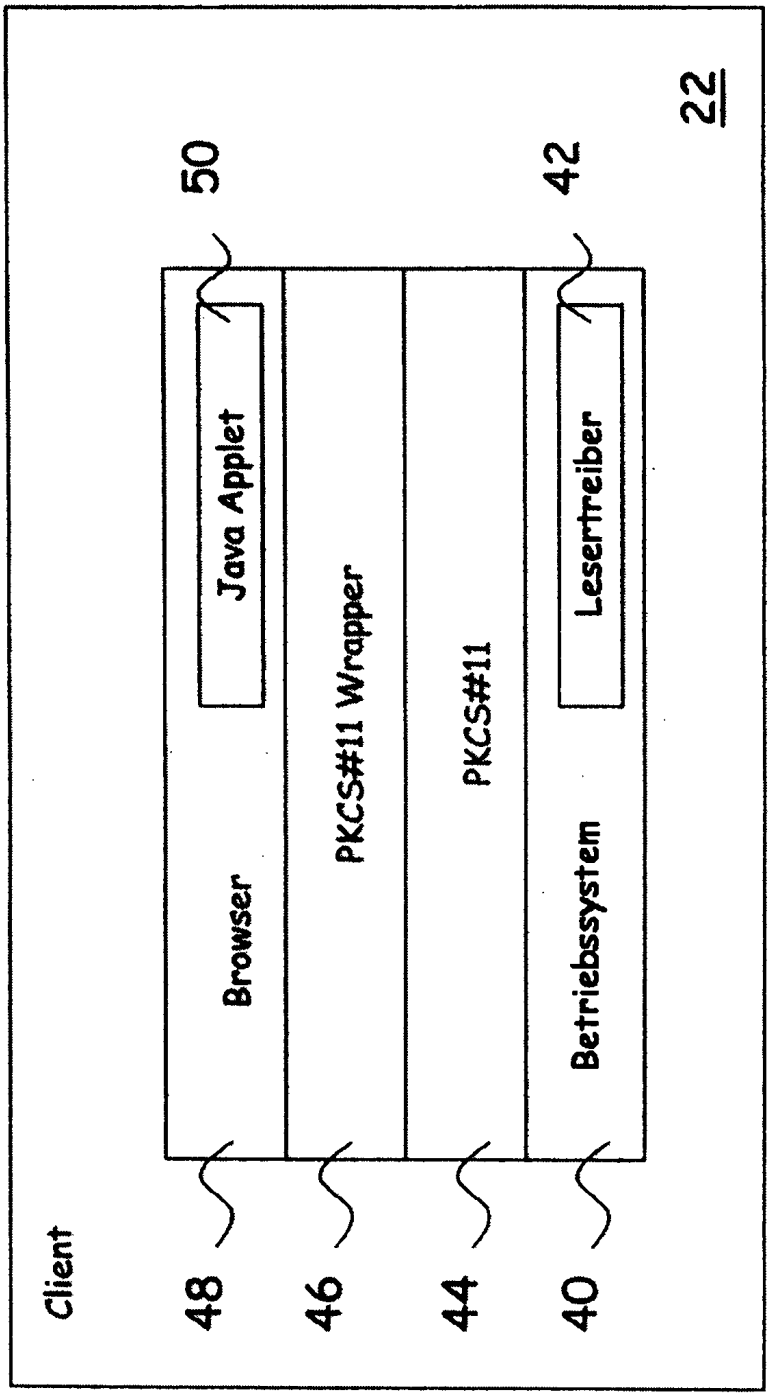
22
50
42
Java Applet
Lesertreiber
Browser
PKCS#11 Wrapper
PKCS#11
Betriebssystem
Client
48
46
44
40

Fig. 2

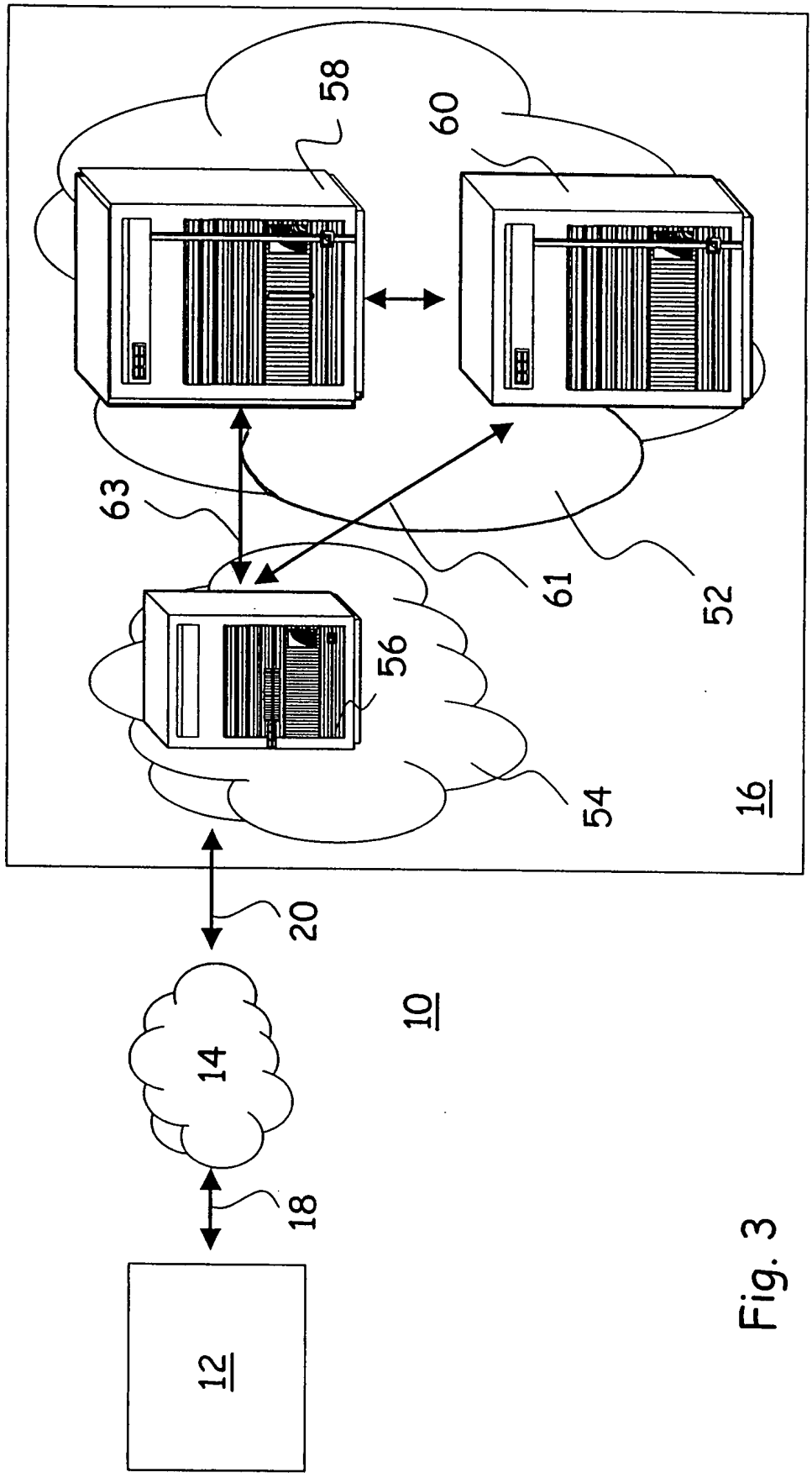

Fig. 3

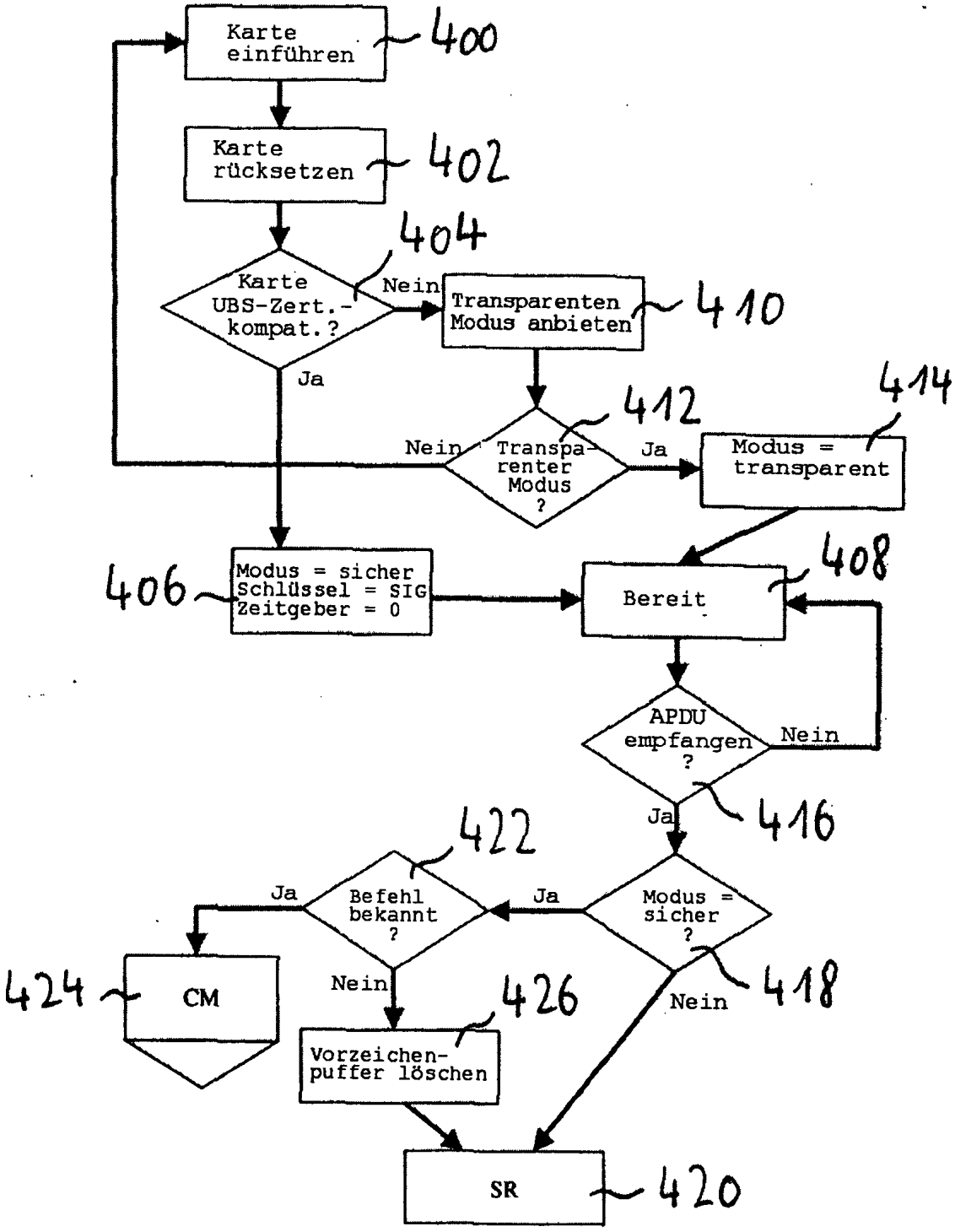
Karte einführen
400
Karte rücksetzen
402
404
Karte UBS-Zert.-kompat.?
Nein
Transparenten Modus anbieten
410
Ja
412
Nein
Transpa-renter Modus ?
Ja
414
Modus = transparent
406
Modus = sicher Schlüssel = SIG Zeitgeber = 0
Bereit
408
APDU empfangen ?
Nein
Ja
416
422
Ja
Befehl bekannt ?
Ja
Modus = sicher ?
424
CM
Nein
426
Nein
418
Vorzeichen-puffer löschen
SR
420

Fig. 4

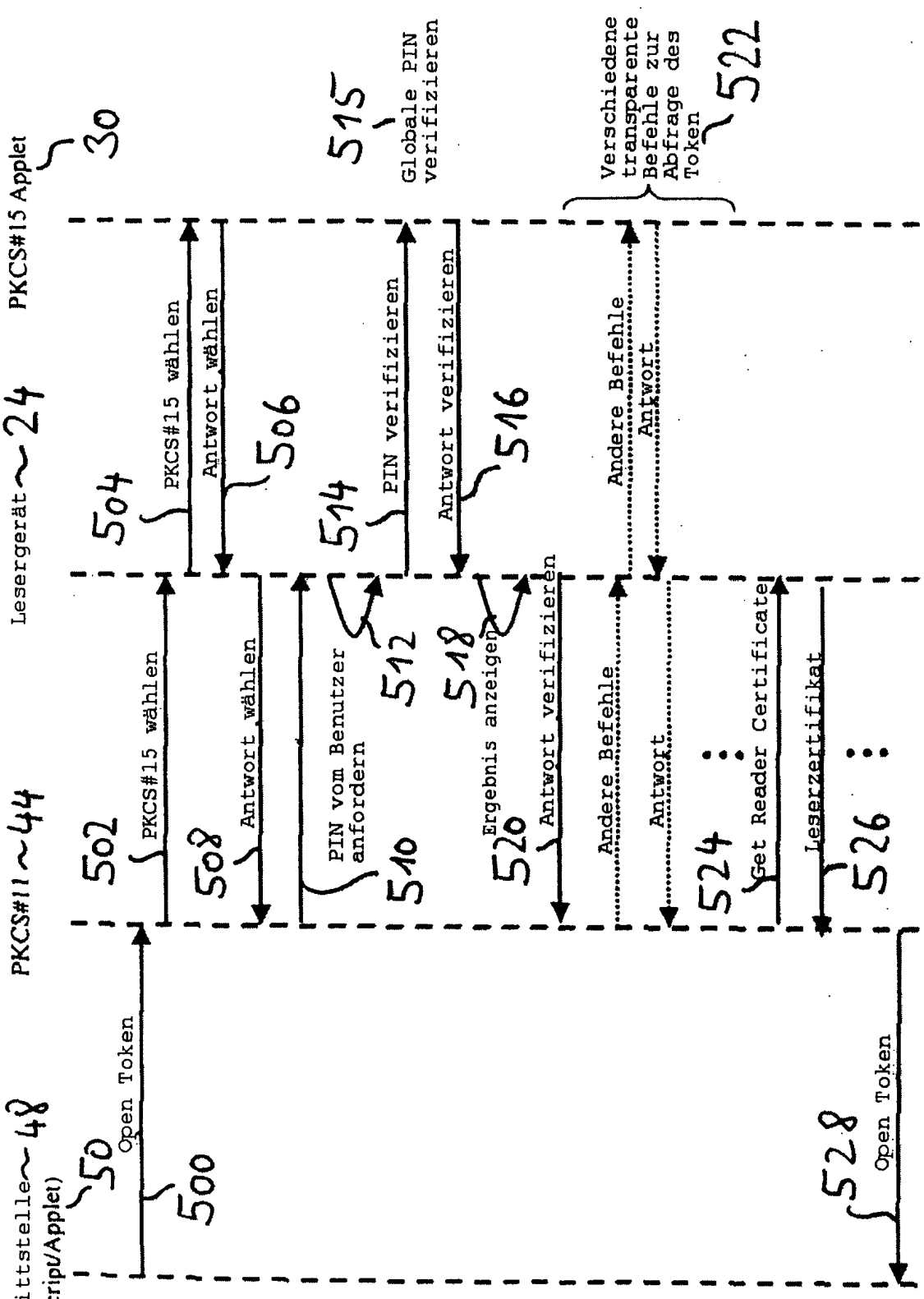
Browser-Schnittstelle (Java Script/Applet) 48
PKCS#11 44
Lesergerät 24
PKCS#15 Applet 30
Benutzer veranlasst Anmeldung
50
Open Token
500
PKCS#15 wählen
502
Antwort wählen
504
PKCS#15 wählen
506
PIN vom Benutzer anfordern
510
PIN verifizieren
512
Ergebnis anzeigen
514
Antwort verifizieren
516
Globale PIN verifizieren
515
Andere Befehle
518
Antwort
520
Verschiedene transparente Befehle zur Abfrage des Token
522
Get Reader Certificate
524
Lesezertifikat
526
Open Token
528

Fig. 5

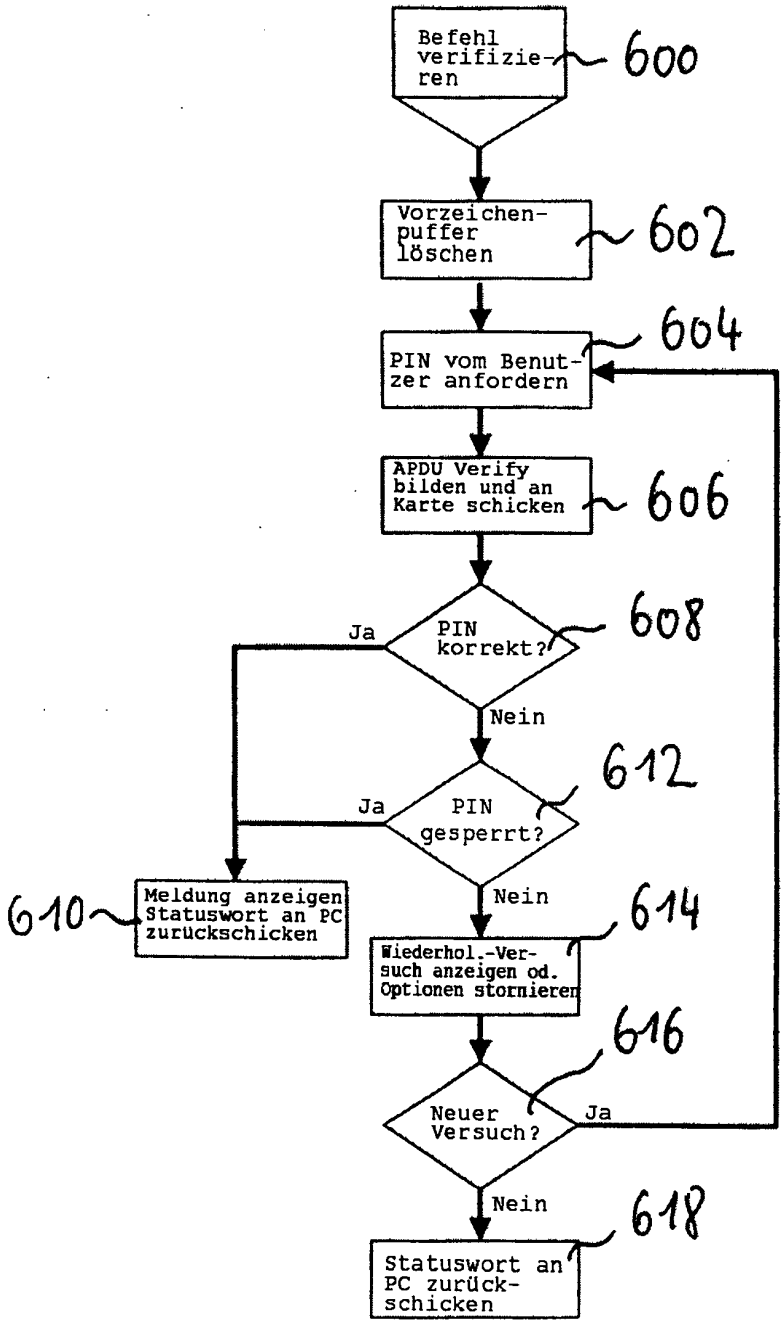
Befehl verifizieren
600
Vorzeichenpuffer löschen
602
PIN vom Benutzer anfordern
604
APDU Verify bilden und an Karte schicken
606
PIN korrekt?
608
Ja
Nein
PIN gesperrt?
612
Ja
Nein
Meldung anzeigen Statuswort an PC zurückschicken
610
Wiederhol.-Versuch anzeigen od. Optionen stornieren
614
Neuer Versuch?
616
Ja
Nein
Statuswort an PC zurückschicken
618

Fig. 6

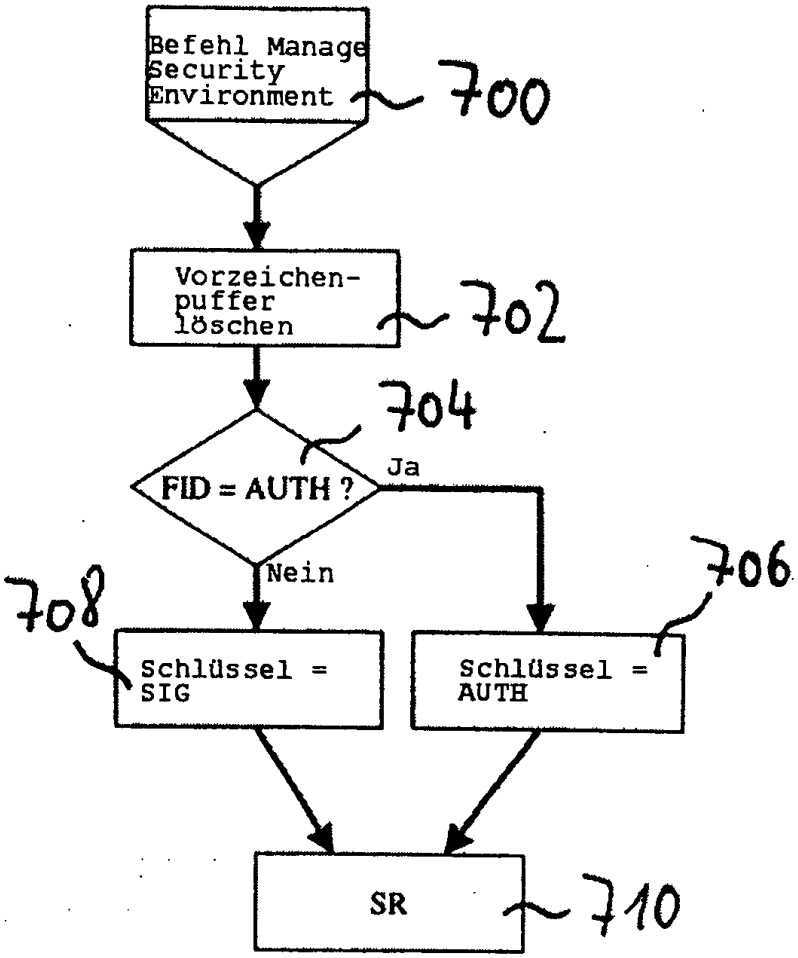
Befehl Manage Security Environment
700
Vorzeichen-puffer löschen
702
704
FID = AUTH ?
Ja
Nein
706
708
Schlüssel = SIG
Schlüssel = AUTH
SR
710

Fig. 7

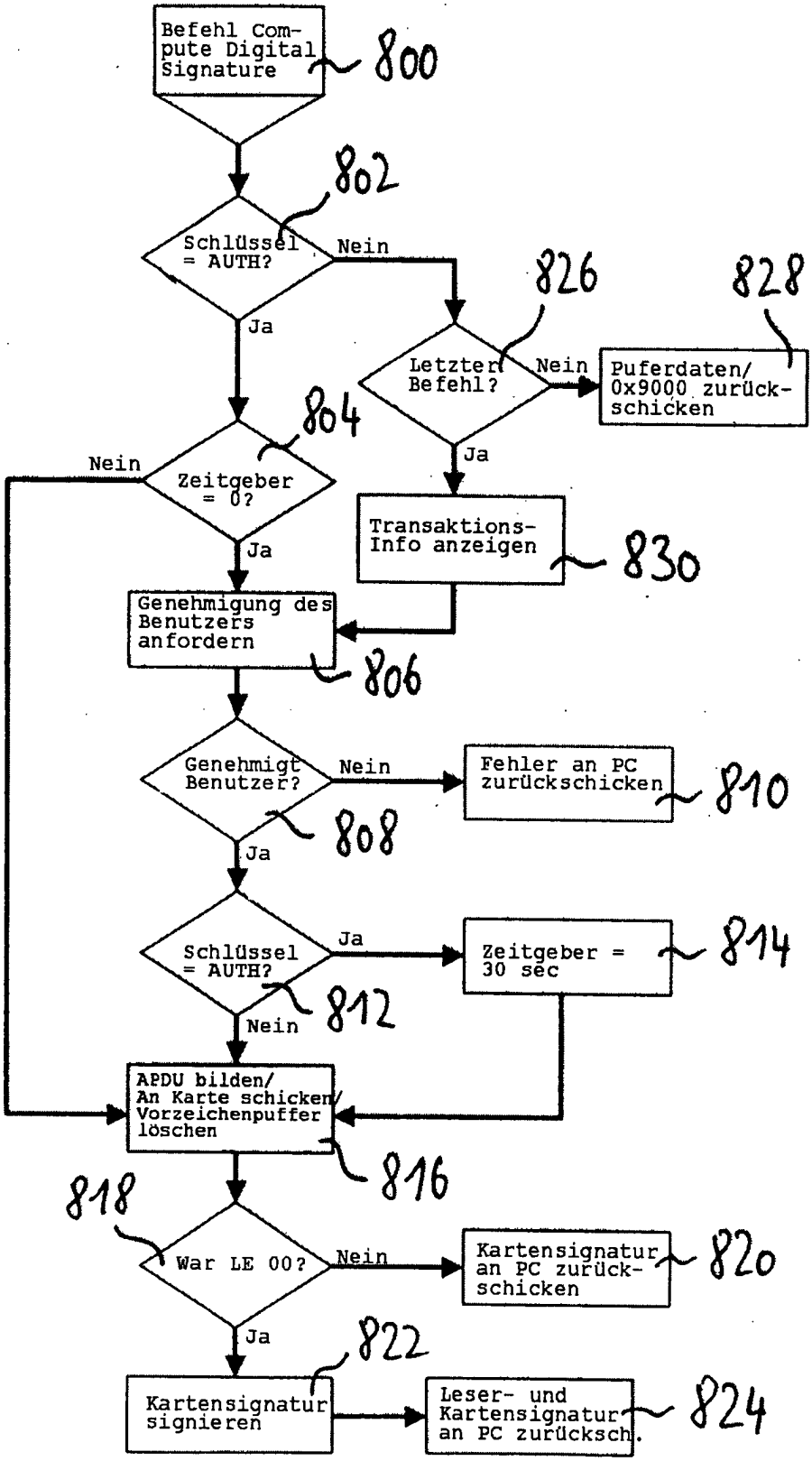
Befehl Compute Digital Signature
800
802
Schlüssel = AUTH?
Nein
Ja
826
Letzter Befehl?
Nein
828
Puferdaten/ 0x9000 zurückschicken
Ja
804
Zeitgeber = 0?
Nein
Ja
Transaktions-Info anzeigen
830
Genehmigung des Benutzers anfordern
806
Genehmigt Benutzer?
Nein
Fehler an PC zurückschicken
810
808
Ja
Schlüssel = AUTH?
Ja
Zeitgeber = 30 sec
814
812
Nein
APDU bilden/ An Karte schicken/ Vorzeichenpuffer löschen
816
818
War LE 00?
Nein
Kartensignatur an PC zurückschicken
820
Ja
822
Kartensignatur signieren
Leser- und Kartensignatur an PC zurücksch.
824

Fig. 8

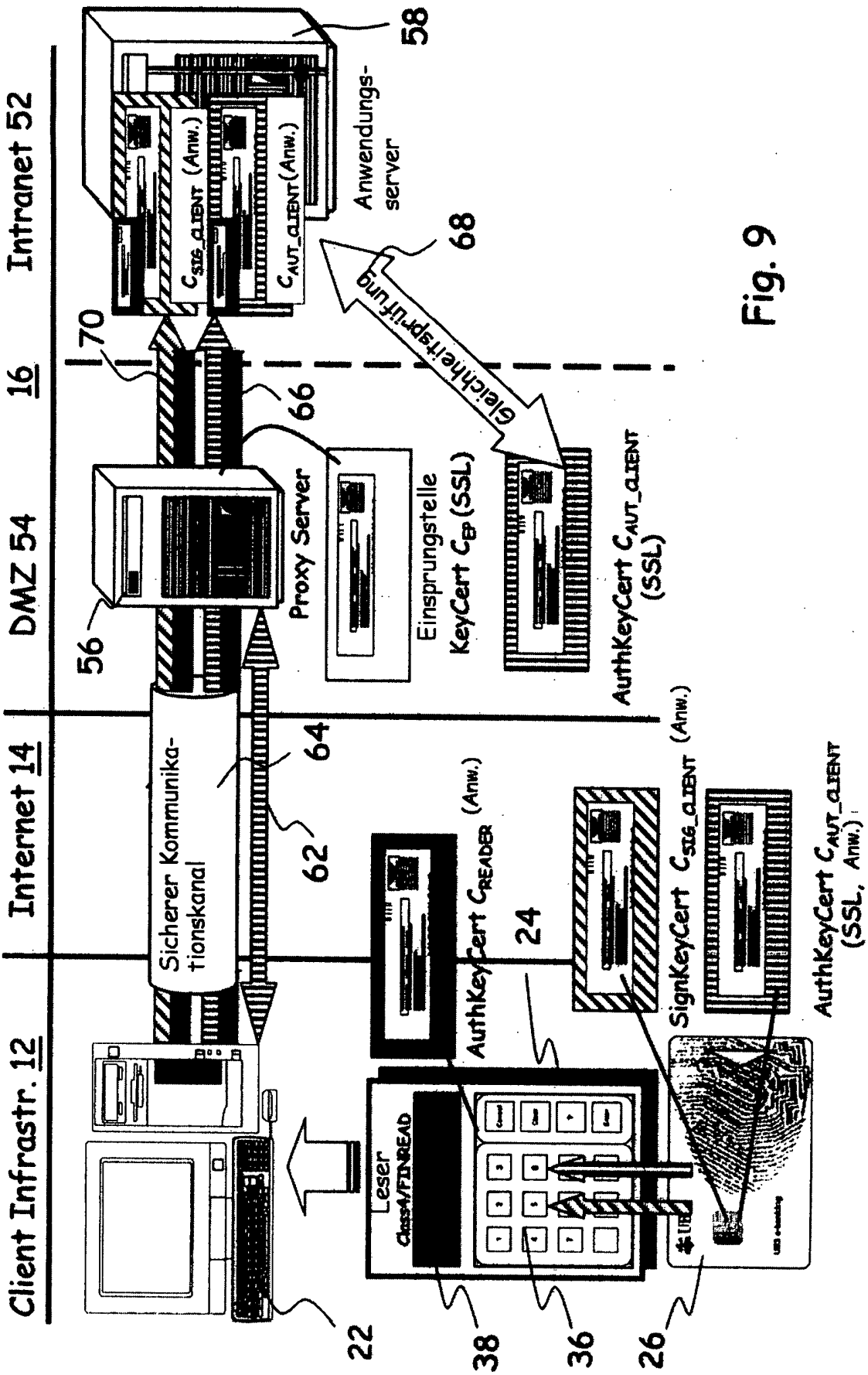
Client Infrastr. 12
Internet 14
DMZ 54
16
Intranet 52
58
Anwendungs-server
C_{SIG_CLIENT} (Anw.)
C_{AUT_CLIENT} (Anw.)
68
Gleichheitsprüfung
70
66
56
Proxy Server
Einsprungstelle
KeyCert C_{EP} (SSL)
AuthKeyCert C_{AUT_CLIENT} (SSL)
64
62
Sicherer Kommunikationskanal
AuthKeyCert C_{READER} (Anw.)
24
SignKeyCert C_{SIG_CLIENT} (Anw.)
AuthKeyCert C_{AUT_CLIENT} (SSL, Anw.)
Leser
22
38
36
26

Fig. 9

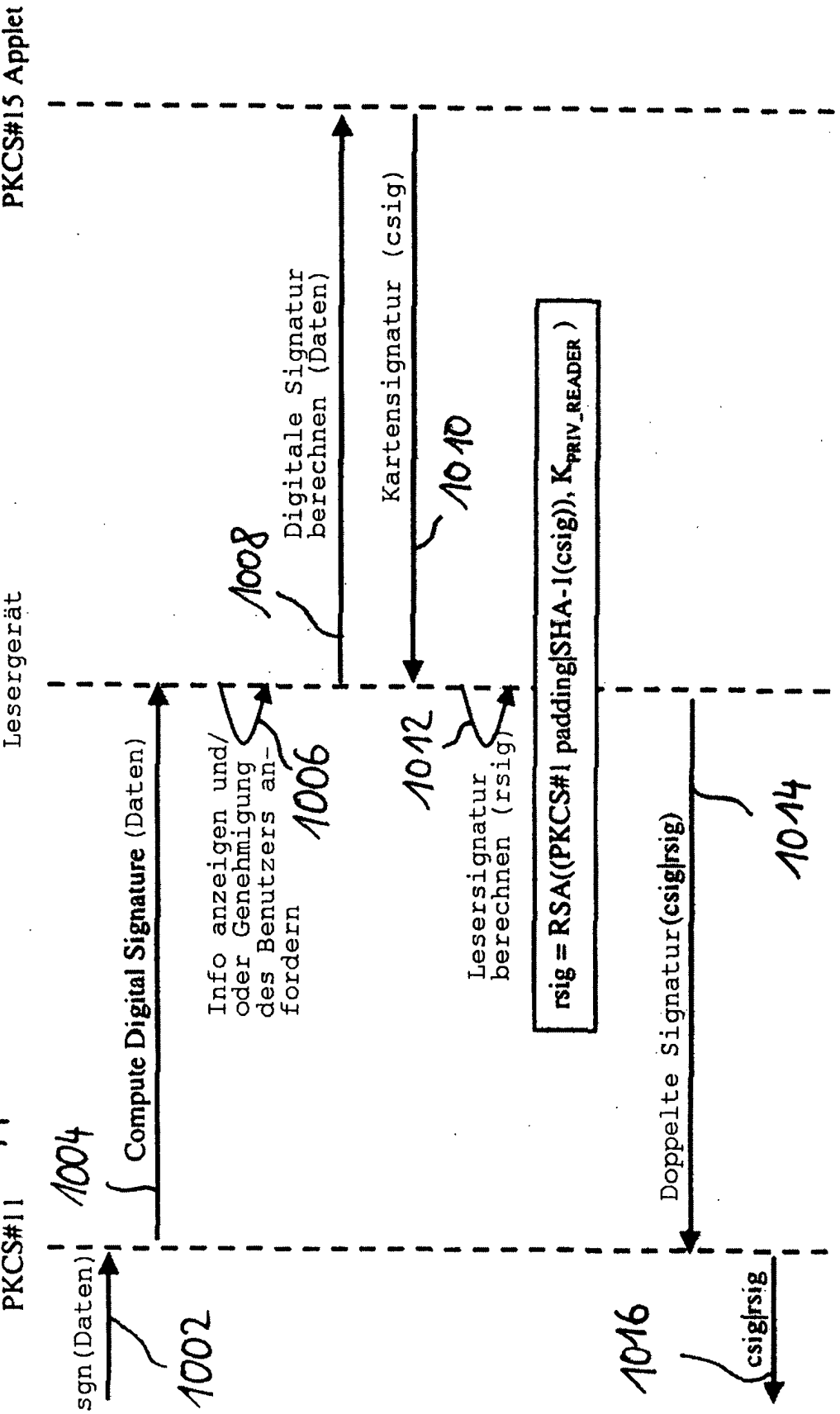
PKCS#11
44
PKCS#15 Applet
30
Lesergerät
24
sgn(Daten)
1002
Compute Digital Signature (Daten)
1004
Info anzeigen und/
oder Genehmigung
des Benutzers an-
fordern
1006
Digitale Signatur
berechnen (Daten)
1008
Kartensignatur (csig)
1010
Lesersignatur
berechnen (rsig)
1012
rsig = RSA((PKCS#1 padding|SHA-1(csig)), K PRIV_READER)
Doppelte Signatur(csig|rsig)
1014
csig|rsig
1016

Fig. 10

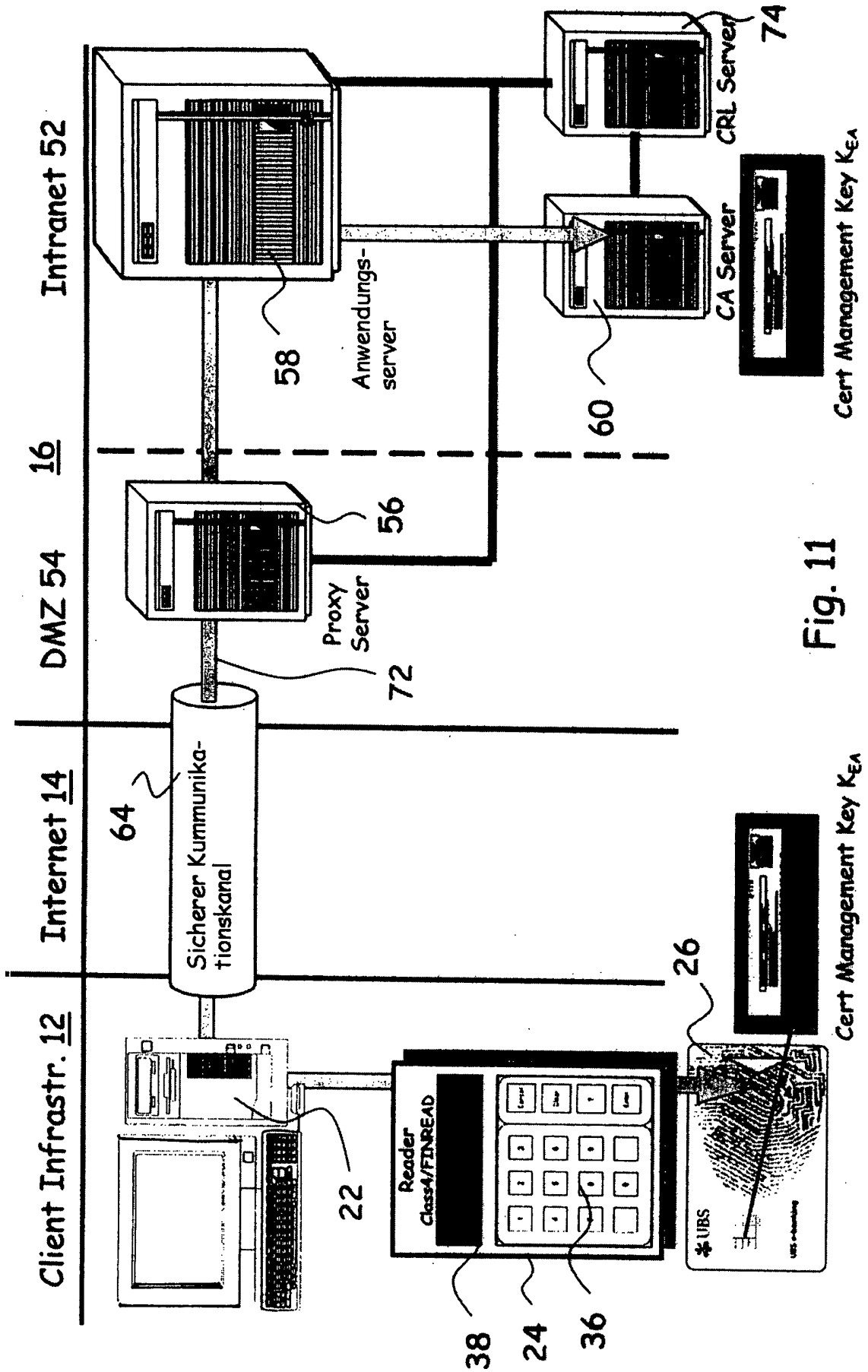
Client Infrastr. 12
Internet 14
DMZ 54
16
Intranet 52
Sicherer Kommunikationskanal
64
Proxy Server
72
56
Anwendungsserver
58
CA Server
60
CRL Server
74
Cert Management Key K_{EA}
Cert Management Key K_{EA}
26
Reader Class4/FINREAD
22
38
24
36
UBS

Fig. 11

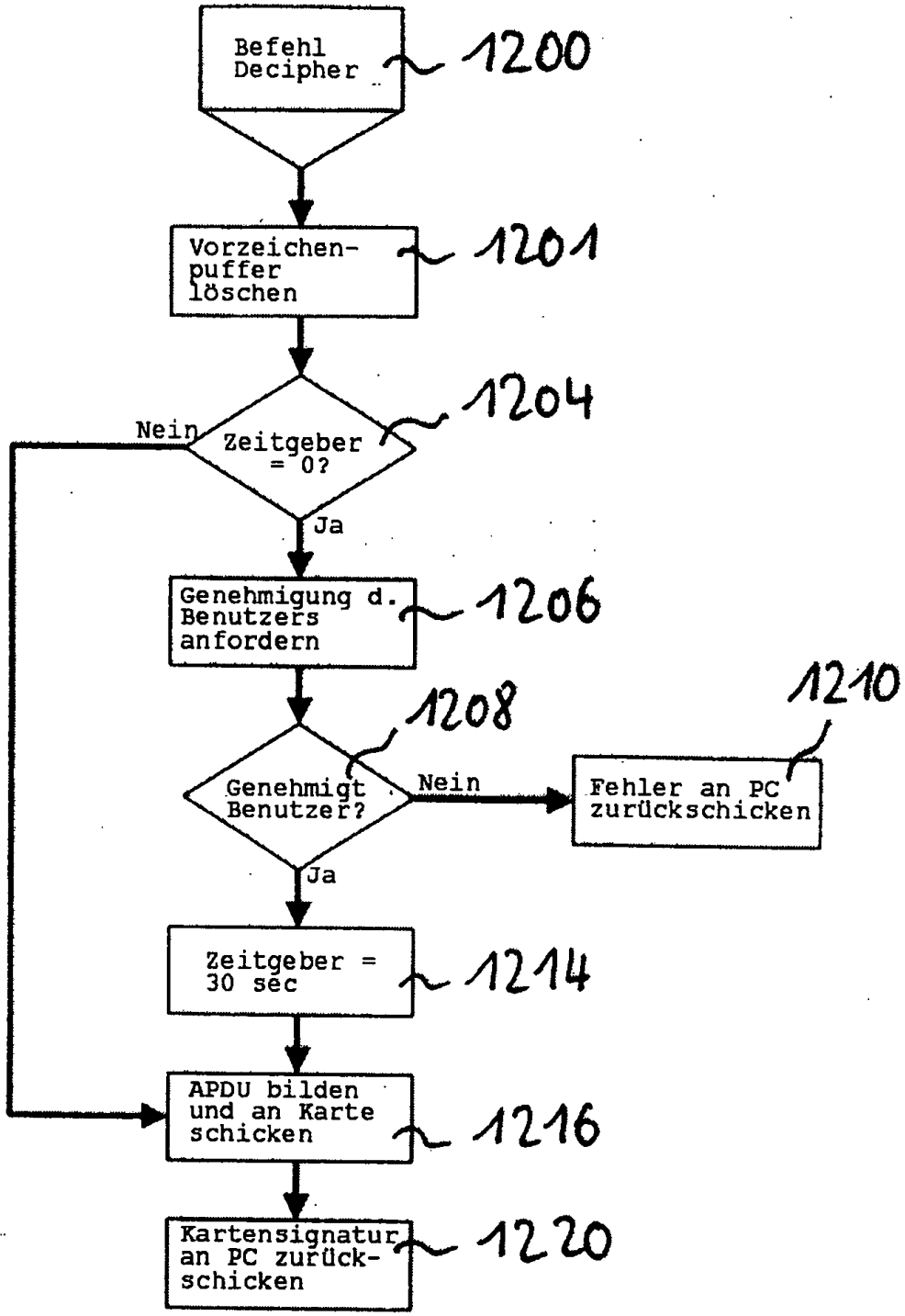
Befehl Decipher
1200
Vorzeichen-puffer löschen
1201
Zeitgeber = 0?
1204
Nein
Ja
Genehmigung d. Benutzers anfordern
1206
Genehmigt Benutzer?
1208
Nein
Fehler an PC zurückschicken
1210
Ja
Zeitgeber = 30 sec
1214
APDU bilden und an Karte schicken
1216
Kartensignatur an PC zurück-schicken
1220

Fig. 12

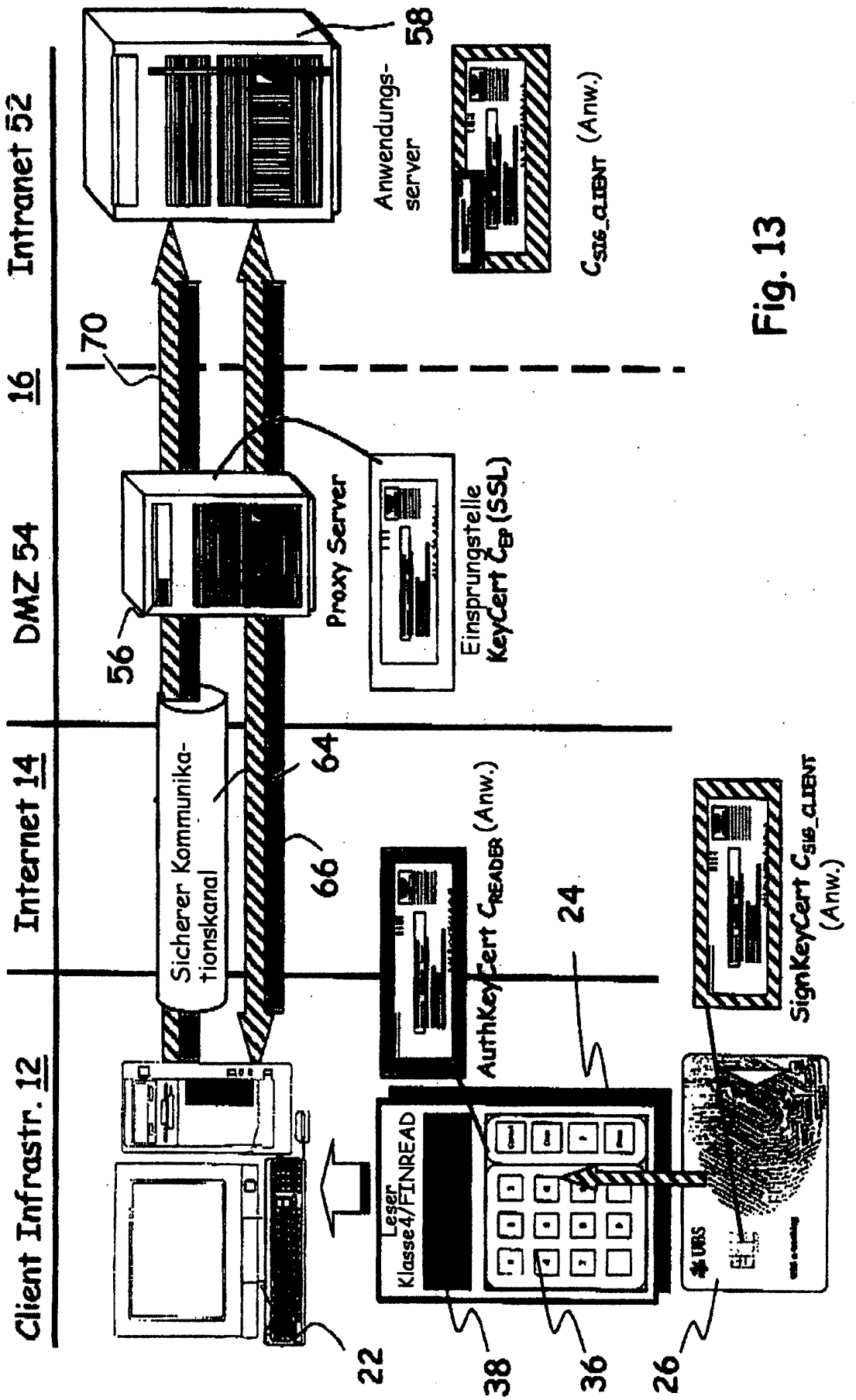
Client Infrastr. 12
Internet 14
DMZ 54
16
Intranet 52
70
58
Anwendungs-server
C_{SIG_CLIENT} (Anw.)
56
Proxy Server
Einsprungstelle
KeyCert C_{EP} (SSL)
Sicherer Kommunikationskanal
64
66
24
AuthKeyCert C_{READER} (Anw.)
SignKeyCert C_{SIG_CLIENT} (Anw.)
Leser Klasse4/FINREAD
UBS
22
38
36
26

Fig. 13