TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

MASTERARBEIT

MODERNE KRYPTOGRAFIE

ausgeführt am Institut für

SOFTWARETECHNIK UND INTERAKTIVE SYSTEME

der Technischen Universität Wien

unter der Anleitung von

Univ.-Doz. Dipl.-Ing. Dr. Ernst Piller

durch

Emanuel Höfenstock

Eugenia 71

3943 Schrems

Datum

Unterschrift

Dank an meine Eltern

Kurzfassung

In dieser Magisterarbeit soll ein Überblick über moderne kryptografische Anwendungen gegeben werden. Dabei war das Hauptziel, dass dem Leser ein Einblick in aktuelle Verfahren gegeben und deren praktische Relevanz gezeigt wird. Die Arbeit selbst gliedert sich in zwei große Hauptkapitel.

Im ersten Kapitel wird auf elektronisches Geld eingegangen. Dieses Thema hat eine große aktuelle Relevanz, da die Anzahl von Bezahlvorgängen über das Internet ständig steigt. Dabei wurde zuerst eine Einführung in die Thematik gegeben, um dem Leser wichtige Informationen, die für die spätere Ausführung der kryptografischen Verfahren notwendig sind, gegeben. Dazu zählen die Eigenschaften und die Grundkonzepte elektronischer Zahlungsmittel. Nach diesem theoretischen Teil wendet sich dieses Kapitel der Praxis zu. Das Hauptaugenmerk in diesem Abschnitt liegt auf dem Verfahren ECash von David Chaum.

Das zweite Hauptkapitel befasst sich mit dem höchstaktuellen Thema der maschinenlesbaren Reisedokumente. Dazu zählt auch der neue Reisepass, der im Moment in Österreich in der 1. Ausbaustufe ausgegeben wird. Für die Zukunft wurde bereits die Einführung der 2. Ausbaustufe beschlossen. Bei dieser werden biometrische Daten auf dem Chip des Reisepasses gespeichert. Da es sich bei diesen Daten um hochsensible und personenbezogene handelt, müssen diese auch besonders gegen Missbrauch geschützt werden. Dazu wurde eine erweiterte Inspektionsprozedur definiert, die sich „Extended Access Control" (EAC) nennt. Mit Hilfe der EAC kann der Aussteller des Reisepasses bestimmen, wer Zugriff auf die biometrischen Daten erhält. Realisiert wird dies durch die Vergabe von entsprechenden Zertifikaten an die einzelnen Inspektionssysteme.

Inhaltsverzeichnis

Kapitel II

Maschinenlesbare Reisedokumente 39

Abbildungsverzeichnis

Kapitel I

Elektronisches Geld

Kapitel II

Maschinenlesbare Reisedokumente

Kapitel I
Elektronisches Geld

1. Einleitung

Verfahren, die eine bargeldlose Übertragung von Geldbeträgen auf elektronischem Wege möglich machen, werden mit dem Überbegriff des elektronischen Geldes (auch Digitales Geld genannt) zusammengefasst. Dabei ist eine Speicherung am Computer, einem Datenträger oder Handy möglich.

Nach der E-Geld-Richtlinie 2000/46/EG lautet die für Europa geltende offizielle Definition für elektronisches Geld wie folgt:

> *"ein monetärer Wert in Form einer Forderung gegen die ausgebende Stelle, der*
>
> - *auf einem Datenträger gespeichert ist,*
> - *gegen Entgegennahme eines Geldbetrags ausgegeben wird, dessen Wert nicht geringer ist als der ausgegebene monetäre Wert,*
> - *von anderen Unternehmen als der ausgebenden Stelle als Zahlungsmittel akzeptiert wird."*

2. Eigenschaften elektronischer Zahlungssysteme

An elektronische Zahlungssysteme wird eine Vielzahl von Anforderungen gestellt, die ein bestimmtes Maß an Sicherheit gewährleisten.
Die größte und auch wichtigste Anforderung stellt die sichere Datenübertragung dar. Transaktionen zwischen Banken werden über das

eigens entwickelte SWIFT-Netz abgewickelt, welches einen geschlossenen und abgesicherten Zahlungsverkehr ermöglicht. Besonders gefährdet sind Transaktionen über das Internet. Um diese Aktivitäten dennoch so sicher wie möglich zu machen, werden zur Absicherung der Daten komplexe kryptografische Verschlüsselungsmechanismen verwendet. Diese Mechanismen haben neben der Hauptaufgabe der Verschlüsselung noch die Aufgabe, die Authentizität und die Unversehrtheit mittels digitaler Signatur zu gewährleisten. Zusätzlich wird die Berechtigung des Nutzers mit Passwörtern, PINs, Smartcards oder ähnlichen Methoden überprüft. In der Zukunft werden diese Systeme durch andere Identifikationsmöglichkeiten wie zum Beispiel Fingerabdrücke, Stimmprofile oder genetische Fingerabdrücke abgelöst. Daraus ergibt sich der Vorteil, dass man sich keine Zeichenfolgen mehr merken muss. Jedoch birgt es auch die Gefahr, dass man sich bei Transaktionen von hohem Wert in große Gefahr begibt. Ab einem gewissen Wert muss man leider davon ausgehen, dass für kriminelle Personen ein menschliches Leben weniger wert ist, als der Wert der Ware.

Eine weitere wichtige Eigenschaft elektronischer Zahlungssysteme ist die Höhe der Transaktionskosten. Diese entstehen bei jeder Transaktion und beinhalten die Kosten für die Verarbeitung der Daten, die Installation der benötigten Hardware, die Instandhaltung der Sicherungsfunktionen und die Vergütung des Zeitaufwandes. Daraus kann man auf drei Gruppen schließen:

- Niedrige Transaktionskosten: Kosten im Centbereich, Minimierung der Ausgaben für Hard- und Software sowie Kommunikation, Beispiel: Telefonkarte
- Mittlere Transaktionskosten: Kosten ca. 50 Cent, hohe Fixkosten durch automatisierte Transaktionen, Beispiel: POS-Systeme (Point Of Sale)
- Hohe Transaktionskosten: Kosten ca. 1-2 Euro, Systeme mit zum Teil manueller Abwicklung, Beispiel: Kreditkarten

Aufgrund der Höhe des Transaktionsvolumens kann man in vier Kategorien unterteilen:

- Nanopayments (ca. 1 – 10 Cent)
- Micropayments (ca. 10 Cent – 10 Euro)
- Medium-Payments (ca. 10 – 10.000 Euro)
- Macropayments (ab ca. 10.000 Euro)

Aufgrund dieser Einteilung ist ersichtlich, dass nicht jedes Zahlungssystem für jede Transaktion sinnvoll einsetzbar ist. Besonders bei Nanopayments und Micropayments muss ein System verwendet werden, welches keine oder nur sehr geringe Transaktionskosten aufweist.

In gewissen Situationen ist die Anonymität des Käufers gefordert, da dieser seine Identität nicht Preis geben möchte. Aus diesem Grund ist die Rückverfolgbarkeit eine wichtige Eigenschaft elektronischen Geldes. Laut Univ. Prof. Dr. Wolfgang Klas kann man vier Stufen der Rückverfolgbarkeit unterscheiden:

- Uneingeschränkte Rückverfolgbarkeit (z.B. Bezahlung mit Kreditkarte)
- Eingeschränkte Rückverfolgbarkeit (z.B. Chipkarten mit Wertspeicherung)
- Nicht rückverfolgbare Rückverfolgbarkeit (z.B. Bezahlung mit Bargeld)
- Benutzergesteuerte Rückverfolgbarkeit (z.B. verschlüsselte Quittung, die nur der Zahlende entschlüsseln und somit die Anonymität aufheben kann)

Weiters unterscheiden sich elektronische Zahlungsmittel in der Überprüfbarkeit. Zum einen gibt es Systeme mit einer Online-Überprüfung, bei der die Liquidität des Käufers während des Bezahlvorganges bei der entsprechenden Bank überprüft wird. Diese Vorgangsweise ist zum Beispiel häufig bei kreditkartenbasierten Systemen zu finden. Durch den komplizierten Ablauf fallen große Transaktionskosten an. Im Gegensatz dazu stehen Systeme, die auf eine Online-Prüfung verzichten. Der große

Vorteil liegt in den wesentlich geringeren Kosten und der schnelleren Abwicklung. Zum Einsatz kommen fälschungssichere Hardware oder kryptografische Techniken, wobei eine geringere Systemsicherheit in Kauf genommen wird. Anzutreffen ist diese Technik zum Beispiel bei Prepaid-Systemen bzw. bei Transaktionen von geringem Wert.

Ob sich ein Onlinezahlungssystem durchsetzen und sich verbreiten kann hängt von der Akzeptanz ab. In „Elektronisches Geld im Internet" erläutert Markus Stolpmann die Akzeptanz in Abhängigkeit von drei Faktoren:

- die Sicherheit des Systems
- die entstehenden Kosten auf beiden Seiten
- dem Verbreitungsgrad

Eine andere Definition der Akzeptanz lautet nach Univ. Prof. Dr. Wolfgang Klas wie folgt: *„Ein Zahlungssystem besitzt Akzeptanzfähigkeit, falls es überall angenommen wird, d.h. elektronische Beträge, die eine bestimmte Bank herausgibt, werden auch von anderen Banken angenommen.*"

Möchte man Bargeld elektronisch nachbilden, so ist die Übertragbarkeit eine wichtige und nicht zu vergessende Eigenschaft. Übertragbarkeit heißt, dass elektronisches Guthaben weitergegeben werden kann, ohne dass in diesem Vorgang die Bank involviert ist. In der Praxis ist das fast nicht realisierbar und es leidet die Systemsicherheit sehr stark daran. Das Hauptproblem liegt darin, dass Kopien von weitergegebenen Guthaben auf mehreren Computern liegen und diese dann von verschiedenen Personen eingelöst werden können. Im Moment ist der Stand der Technik noch nicht soweit, dass ein sicheres System mit unbegrenzter Übertragbarkeit realisiert werden kann.

Ein Vorteil vieler elektronischer Zahlungssysteme gegenüber Bargeld ist die Teilbarkeit. Bei Bargeld kann man zum Beispiel einen 100 Euro Schein nicht in beliebig kleine Teilbeträge aufteilen, ohne dass man den Schein gegen andere tauscht. Bei elektronischen kontobasierten Systemen ist eine Abbuchung jedes beliebigen Teilbetrages kein Problem. Der Begriff

Teilbarkeit lässt sich laut Bernd Rothhaas wie folgt definieren: „*In einem elektronischen Zahlungssystem sollte ein Wert x in eine beliebige Anzahl von Beträgen jedweder Höhe unterteilt werden können. (Der Gesamtwert der Beträge muss natürlich dann wieder x entsprechen.)*"

Aus diesen vielen Eigenschaften und Anforderungen ergibt sich das Problem, dass es nicht das Zahlungssystem gibt. Je nach Anwendungsbereich und geforderten Eigenschaften muss das optimale System gewählt werden. Um einen Eindruck von den vielfältigen Möglichkeiten an verschiedenen Zahlungssystemen zu bekommen werden im nächsten Abschnitt die vier möglichen Grundkonzepte erläutert.

3. Grundkonzepte elektronischer Zahlungssysteme

Im Großen und Ganzen kann man elektronische Zahlungssysteme in drei große Gruppen, nämlich Überweisungssysteme, Schecksysteme und Token Systeme, unterteilen. Bei Token Systemen kann man zusätzlich noch zwischen Einweg und Mehrweg Systemen unterscheiden.

3.1 Überweisungssysteme

Bei dem elektronischen Überweisungssystem handelt es sich um ein kontobasiertes System. Das heißt, dass sowohl der Händler als auch der Kunde ein Konto bei einer Bank besitzt, wobei diese durchaus bei verschiedenen Banken sein können. Vergleicht man das herkömmliche Überweisungssystem mit dem elektronischen System so stellt man fest, dass es lediglich Unterschiede beim Ablauf von Zahlungsaufforderung und Überweisungsauftrag gibt. Diese werden nämlich auf elektronischem Wege durchgeführt. Das folgende Schema soll die Funktionsweise grafisch darstellen:

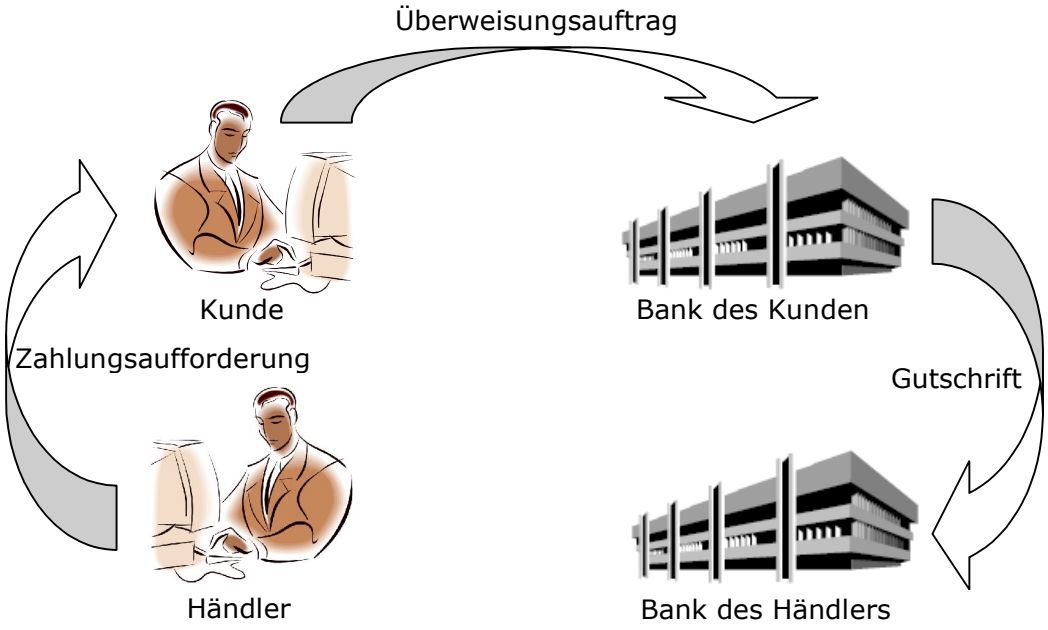


Abbildung 1: Überweisungssystem

Bei diesem System hängt die Sicherheit im Wesentlichen von folgenden Punkten ab:

- Wirksamkeit der Zugangskontrolle und der Identifikationsmechanismen bei der Autorisierung
- Verlässlichkeit des Datentransfers
- Infrastruktur der Übertragungskanäle

Diese Punkte ergeben bei Einhaltung einen wirksamen Schutzmechanismus, der gewährleistet, dass die Übertragung sicher und gegen Fälschung abgesichert ist. Durch den großen technischen Aufwand und die Bearbeitungszeiten und damit verbundenen Kosten ergeben sich relativ hohe Transaktionskosten. Aufgrund der Tatsache, dass die Bank alle Daten einer Überweisung kennt (Quelle, Ziel, Betrag, Datum, Uhrzeit), ergibt sich eine uneingeschränkte Rückverfolgbarkeit.

3.2 Schecksysteme

Wie auch das Überweisungssystem ist das elektronische Schecksystem kontobasiert. Der Schecknehmer (der Empfänger) besitzt die Möglichkeit, dass er den Scheck bei seiner eigenen Bank oder bei der Bank des Ausstellers einlöst. Der Ablauf bei einer Bezahlung mittels Scheck lässt sich wie folgt beschreiben:

Eine Person A (der Kunde) möchte eine Rechnung begleichen, wobei er das Geld nicht in Bar bei sich hat, sondern bei seiner Bank ein entsprechendes Guthaben besitzt. Dazu stellt er ein Schreiben auf, in dem er die Bank auffordert der Person eine bestimmte Geldsumme auszubezahlen und diese Summe von seinem Konto abzuziehen. Dieses Schreiben nennt man einen Scheck. Diesen Scheck übergibt der Kunde nun der begünstigten Person (dem Händler). Diese Person kann nun bei der Bank des Ausstellers oder bei seiner eigenen Bank das Geld ausgehändigt bekommen.

Das folgende Schema soll die Funktionsweise grafisch darstellen:

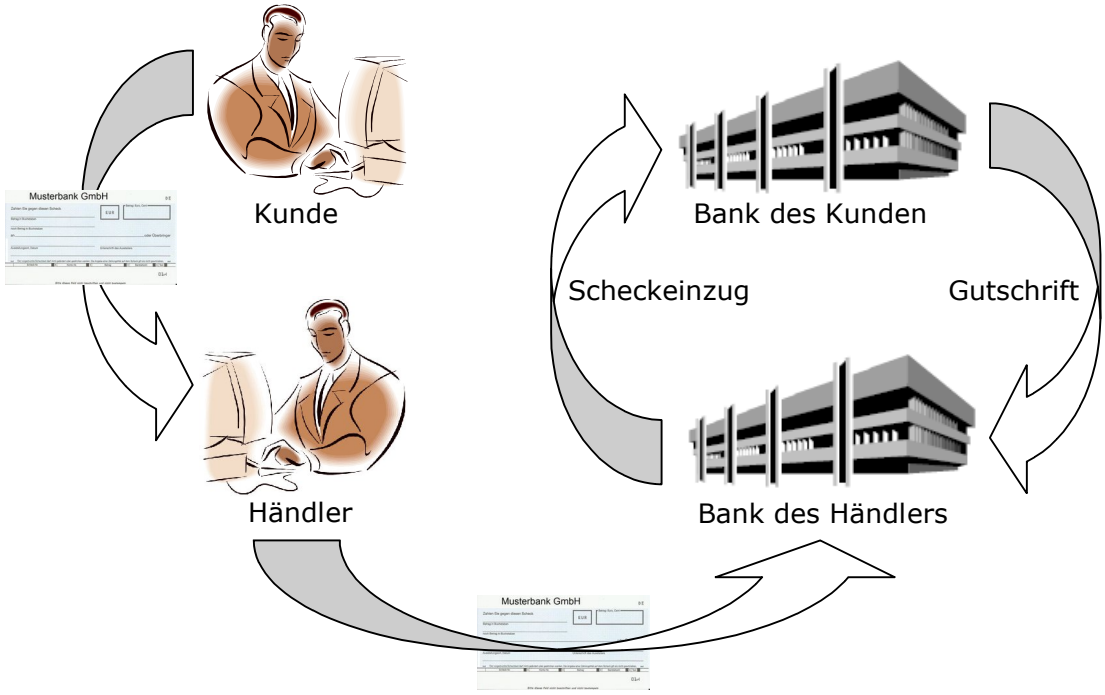


Abbildung 2: Schecksystem

Bei diesem System hängt die Sicherheit im Wesentlichen von den gewählten Verschlüsselungsmechanismen und der verwendeten digitalen Signatur ab.
Durch die vom Scheckbearbeitungsverfahren hervorgerufenen hohen Transaktionskosten ist das Schecksystem nicht gerade für die Verwendung bei kleinen Beträgen geeignet. Ein weiterer Wehrmutstropfen ist der Datenschutz. Durch die Funktionsweise ist es nötig, dass der Aussteller zu jedem Zeitpunkt der Transaktion nachvollziehbar ist.

3.3 Token Systeme

Bei Tokensystemen werden elektronische Datenpakete erzeugt, die bei einer Bank gegen reales Geld getauscht werden können. Diese Datenpakete werden Token genannt und entsprechen einer gewissen Geldeinheit. Transaktionsdetails können nicht übermittelt werden, da der Transfer nicht immer innerhalb einer oder zwischen mehreren Banken stattfindet. Das heißt es können Token theoretisch beliebig oft weitergegeben werden bis sie wieder bei einer Bank eingetauscht werden. Solche Token werden Mehrweg-Token genannt und sind mit dem derzeitigen Wissensstand in der Kryptografie nicht realisierbar.
Es sind nur Token realisierbar, die generiert, weitergegeben und wieder in reales Geld umgetauscht werden. Diese Token nennt man Einweg-Token.
Generell ergeben sich bei Token Systemen drei große Probleme. Token sollen weder fälschbar noch manipulierbar sein, dürfen nicht kopierbar und mehrmals verwendbar sein und sie sollen einen hohen Grad an Anonymität realisieren. Um die Echtheit der Token zu garantieren werden diese mit einer digitalen Signatur der ausgebenden Bank ausgestattet. Das Problem, dass Kopien von Token erstellt werden, die mehrmals verwendet werden, nennt man double spending und wird in 3.3.1 genauer erklärt. Zur Lösung wurde das so genannte two-part-lock Verfahren entwickelt, welches in 3.3.2 beschrieben wird. Nun bleibt noch das letzte große Problem übrig und zwar die Forderung nach einem hohen Grad an Anonymität. Um auch diesen Punkt zu realisieren wurde das Prinzip der blinden Signatur entwickelt, auf das in 3.3.3 genauer eingegangen wird.

3.3.1 Problem des double spending

Das Kopieren und mehrfache Wiederverwenden von elektronischen Münzen (Token) nennt man „double spending". Da dies ein krimineller Vorgang ist, wäre es für die Bank sehr wichtig, dass sie eine Möglichkeit besitzt zu bestimmen, ob der Händler oder der Kunde die Geldeinheit vervielfacht hat. Dies ist mit dem two-part-lock Verfahren und einer Datenbank, in der alle verwendeten Seriennummern abgelegt werden, möglich. Um die Größe der Datenbank nicht explodieren zu lassen ist ein Ablaufdatum der Seriennummern auf der elektronischen Geldeinheit sinnvoll.

3.3.2 Einschub two-part-lock Verfahren

Bei diesem Verfahren werden die Kundendaten symmetrisch und der passende Schlüssel mittels dem two-part-lock Verfahren verschlüsselt. Diese beiden verschlüsselten Daten werden der elektronischen Geldeinheit beigefügt. Wird nun die Geldeinheit eingelöst, so entschlüsselt der Händler die erste Hälfte der Kundendaten und schickt diese mit der Seriennummer an die Bank. Die Bank überprüft die Seriennummer und fügt sie in die Datenbank samt den Kundendaten ein. Ist diese Nummer noch nicht in der Datenbank registriert, so wird sie erstmalig verwendet und die Transaktion ist gültig. Wird die Geldeinheit nun zu einem zweiten Händler geschickt, so entschlüsselt dieser die zweite Hälfte der Kundendaten und schickt diese mit der Seriennummer an die Bank. Bei dieser ist die Seriennummer bereits in der Datenbank registriert und die erste Hälfte der Kundendaten gespeichert. Nun werden die Daten zusammengeführt und die Bank hat offenen Zugang zu den Daten. So kann nicht nur eine mehrfache Verwendung aufgedeckt werden sondern auch der Schuldige gefunden werden.

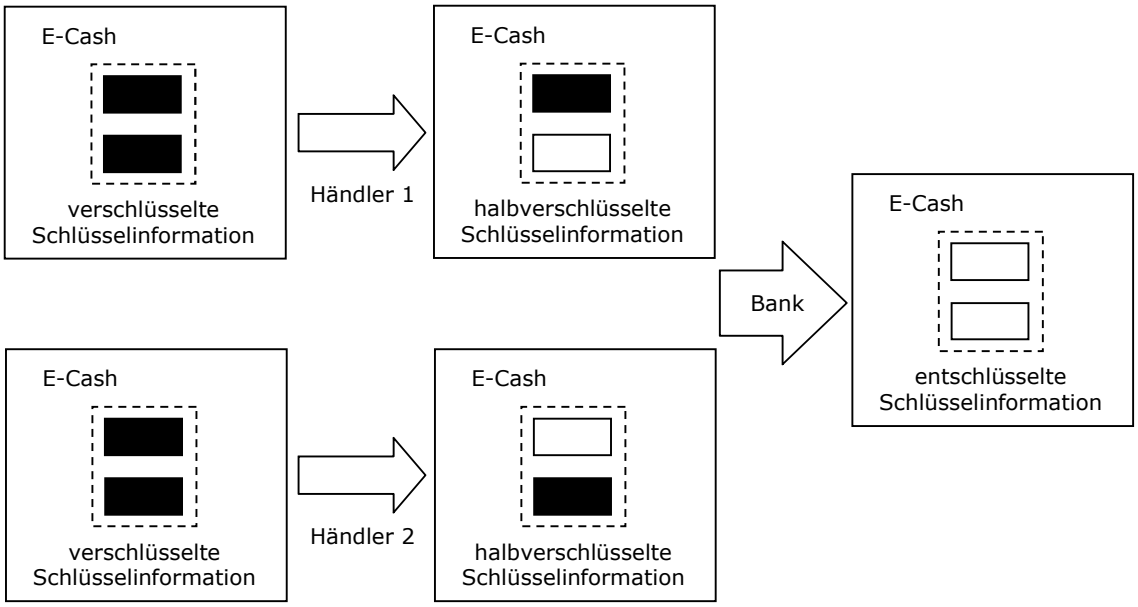


Abbildung 3: Funktionsweise des two-part-lock Verfahrens

Funktionsweise des two-part-lock Verfahrens:
Erster Schritt ist die Zerlegung der Daten (Kundendaten, Schlüsseldaten) in Byte-Strings. Diese Byte-Strings werden N mal kopiert, wobei die Sicherheit größer wird, je größer dieser Faktor gewählt wird. Durch die Zerlegung mittels Secret-Splitting-Protokoll erhält man Stringpaare mit einem linken und einem rechten String. Aus den einzelnen Strings lassen sich die ursprünglichen Daten nicht rekonstruieren, nur durch anwenden der Exklusiv-Oder Funktion auf die beiden Strings ist dies möglich.
Nun werden die Stringpaare einzeln verschlüsselt, das heißt es werden 2N Schlüssel benötigt.
Als nächster Schritt werden die Banknotendaten mit den angehängten verschlüsselten Strings an die begünstigte Person (dem Händler) übermittelt. Dieser fordert vom Kunden pro String einen Schlüssel, damit er entweder die linke oder rechte Seite entschlüsseln kann. Die erhaltenen Daten schickt nun der Händler seiner Bank weiter, die nun diese mit Hilfe der Datenbank überprüft. Liefert die Durchsuchung der Datenbank einen Treffer, so liegt die Situation des double-spendings vor. Nun gibt es zwei Möglichkeiten:

1. idente Schlüsselverteilung: mit großer Wahrscheinlichkeit versucht der Händler die Banknote ein zweites Mal einzureichen. Begründet ist diese Annahme, da es 2N verschiedene Möglichkeiten zur Schlüsselverteilung gibt und daher die Wahrscheinlichkeit derselben Schlüssel äußerst gering ist.
2. nicht idente Schlüsselverteilung: der Schuldige ist mit Sicherheit der Kunde, von dem die Identität ermittelt werden kann. Durch die unterschiedliche Schlüsselverteilung wird mindestens ein String-Paar vollständig entschlüsselt und so werden die Kundendaten frei lesbar.

3.3.3 Prinzip der blinden Signatur

Eine Signatur ist eine digitale Unterschrift. Bei der blinden Signatur unterschreibt eine Person etwas, von dem diese nicht weiß was es ist. Bei elektronischem Geld kann man sich folgendes Szenarium vorstellen. Eine Person schickt eine elektronische Banknote an dessen Bank, ohne dass diese eine Seriennummer kennt. Die Bank unterschreibt diese Banknote und bürgt dadurch für die Echtheit dieser Banknote. Danach wird die signierte Banknote wieder an die Person zurück geschickt. Jetzt kann die Person diese Banknote für eine beliebige Transaktion verwenden und die Zielperson (der Händler) hat durch die Signatur die Gewissheit über die Echtheit der Banknote.

Funktionsablauf:
Der Kunde erstellt eine Banknote BN mit einer dazugehörigen Seriennummer SN. Mit Hilfe eines zufällig gewählten Ausblendfaktors AF verschlüsselt er die Seriennummer des Scheins, protokolliert das Tupel (BN, SN, AF) und schickt die Banknote mit verschlüsselter Seriennummer an die Bank.
Die Bank kann die Seriennummer nicht sehen, kann aber den gewünschten Wert vom Konto des Kunden abbuchen und die Banknote mit einer digitalen Signatur unterschreiben. Somit ist die Echtheit der Banknote garantiert und die Anonymität des Kunden ist gewährleistet, da bei einer Zahlung die

Banknote mit keinen Kundendaten in Verbindung gebracht werden kann. Die Bank protokolliert das Tupel (BN, XX) und schickt die Banknote zurück zum Kunden.

Dieser kann nun mittels Division durch den Ausblendfaktor die Seriennummer wieder einblenden und somit ist die Banknote ein gültiges Zahlungsmittel, das beliebig eingesetzt und nicht auf den Kunden zurückverfolgt werden kann.

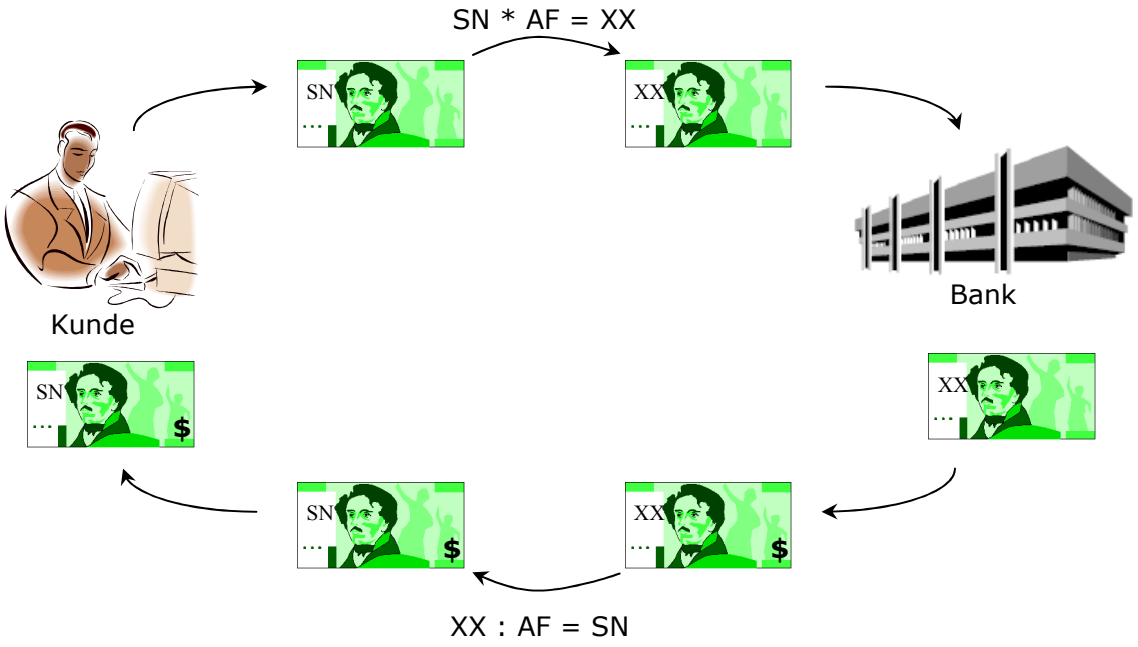


Abbildung 4: Prinzip der Blinden Signatur

3.3.4 Einweg Token Systeme

Bei diesem System ist der Ablauf wie folgt zu beschreiben. Eine Person A (der Kunde) schickt an die Bank einen Abhebungsantrag. Die Bank zieht den gewünschten Betrag vom entsprechenden Konto ab und schickt Token zu diesem Wert an die Person A zurück. Nun kann diese bei einer Person B (der Händler) die gewünschte Rechnung mit diesen Token bezahlen, sofern dieser solche Token akzeptiert. Nun muss der Händler die Token zu seiner Bank bringen, um dafür reales Geld zu bekommen. Die Bank des Händlers schickt die Token weiter an die Bank des Kunden, die wiederum ein entsprechendes Guthaben zurück sendet. Dieses Guthaben wird dann am Konto des Händlers verbucht.
Das folgende Schema soll die Funktionsweise grafisch darstellen:

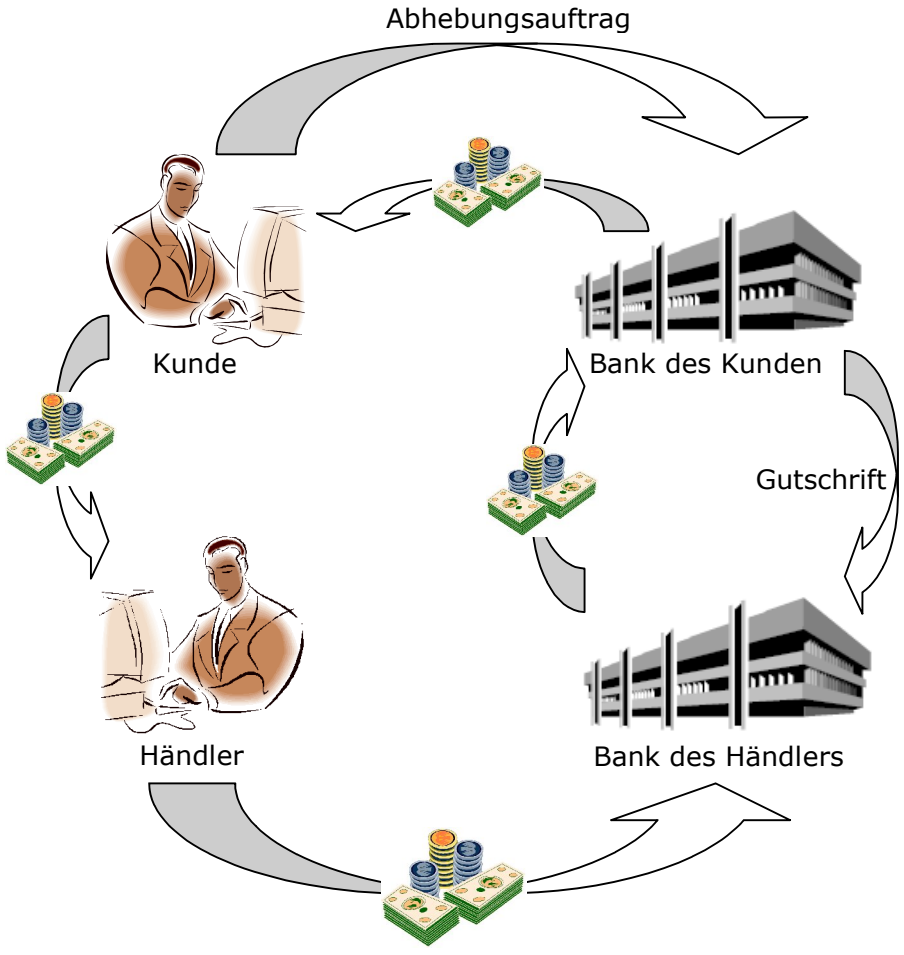


Abbildung 5: Einweg Token System

4. Elektronische Zahlungssysteme

4.1 ECash

ECash wurde von David Chaum, dem Gründer des Unternehmens DigiCash, entwickelt und als geschützter Markenname eingetragen. Der Hauptbestandteil dieses Systems ist die Einbindung der blinden Signatur. David Chaum beschäftigte sich bereits zu seiner Studienzeit mit der Verschlüsselung von Daten insbesondere in Hinsicht auf elektronische Wahlen. Von diesem Standpunkt aus erkannte er, dass auch für elektronische Zahlungsmittel die drei wichtigsten Faktoren, nämlich eine ausgebende Stelle, die Anonymität des Benutzers und die einmalige Verwendung, gewährt werden müssen. Ecash war besonders gut für Kleinstbeträge (Micropayment) geeignet, da die Transaktionskosten sehr klein waren. Leider war Chaum seiner Zeit voraus und konnte damit den Durchbruch aufgrund mangelnden Interesses nicht schaffen. Seine Firma DigiCash wurde Ende der 90er Jahre des vorigen Jahrhunderts aufgelöst. Erst jetzt wird das Interesse von der Seite der Banken größer, da es mittlerweile eine relativ große Nachfrage an solchen Systemen gibt.
Die Herstellung von elektronischem Geld kann der Benutzer über eine einfach zu bedienende Software tätigen. Dieses Geld liegt dann in Form von Dateien auf der Festplatte des Benutzers und kann von diesem für einen Einkauf im Internet verwendet werden. In diesen Dateien werden der Wert, die Seriennummer und die Signatur abgespeichert. Um das System in der Praxis testen zu können, wurde im Jahre 1994 ein Pilotprojekt gestartet. Es wurden sogenannte Cyberbucks an etwa 30000 Internet-Benutzer ausgegeben, die diese in einbezogenen Onlinegeschäften einlösen konnten.

4.1.1 Eigenschaften von Ecash

In diesem Abschnitt werden die wichtigsten Eigenschaften von Ecash angeführt, um das Zahlungssystem besser einordnen zu können. Betrachtet man das Ecash System so fällt einem auf, dass dafür ein eigenes Protokoll entwickelt wurde, das sich über dem TCP/IP Protokoll befindet, um die

Kommunikation zu realisieren. Ein ganz wichtiger Punkt für Chaum war die Anonymität des Benutzers und um dies zu ermöglichen entwickelte er das System der blinden Signatur. Ein anderes Problem, auf das Chaum aufmerksam wurde, war die mehrfache Verwendung der elektronischen Geldmünzen, auch double spending genannt. Durch ein innovatives System wurde ermöglicht, dies zu unterbinden und zusätzliche Mechanismen einzubauen, damit der Bösewicht identifiziert werden konnte. Aufgrund der Tatsache, dass die elektronischen Münzen lokal am Rechner des Benutzers abgelegt werden, war es notwendig, dass Sicherheitsmaßnahmen getroffen werden um im Falle eines Systemabsturzes oder Ähnlichem die Münzen wieder herstellen zu können. Dafür wurden zur Sicherheit zwei Möglichkeiten integriert, nämlich eine Wiederherstellung mittels Coin-ID und andererseits das ständige Sichern von Backups. Um die Kommunikation zu schützen wurde eine Verschlüsselung mittels RSA eingebaut. Dabei wird der öffentliche Schlüssel der Bank mit der Software mitgeliefert und der Benutzer selbst erhält sein Schlüsselpaar bei der Installation durch einen Schlüsselgenerator.

4.1.2 Blinde Signatur im Verfahren von David Chaum

Wie bereits unter Punkt 3.3.3 erklärt, wird bei der blinden Signatur ein Dokument von einer Person unterzeichnet, ohne dass diese weiß, was sie unterschreibt. In der Wirklichkeit könnte man das so realisieren, dass man über das Dokument ein Kohlepapier legt und beides dann in einen Umschlag gibt. Diesen Umschlag unterschreibt dann die signierende Person. Im Verfahren von David Chaum wird dies auf elektronischem Weg gemacht, somit musste er einen kryptografischen Weg finden um zum gewünschten Ergebnis zu gelangen. Chaum realisierte die blinde Signatur mit Hilfe des RSA-Verfahrens, womit es wie folgt funktioniert:
Eine Person A wählt eine Zufallszahl r, die teilerfremd zu n sein muss und berechnet:

$$x = dokument \cdot r^e$$

Diese Zahl wird einer Person B, die das Dokument unterschreiben soll, vorgelegt. Nun signiert diese Person das Dokument, indem sie die Zahl mit ihrem geheimen Schlüssel d potenziert:

$$y = x^d$$

Nun wird das blind unterschriebene Dokument wieder zurück an die Person A geschickt. Diese muss nun noch die Verschlüsselung aufheben, um an das signierte Dokument zu gelangen:

$$y \cdot r^{-1} (= dokument^d)$$

Was für den Benutzer ziemlich einfach aussieht ist in Wahrheit ein relativ komplexes System, welches als nächstes behandelt wird.

4.1.3 Verfahrensbeschreibung

Beim Zahlungssystem Ecash gibt es folgende drei agierende Parteien:

- Kunde: um dieses System nutzen zu können muss der Kunde bei einer Bank ein Konto besitzen, damit er reales Geld in digitales Geld und umgekehrt tauschen kann. Das elektronische Geld wird als Datei auf der Festplatte gespeichert. Zusätzlich benötigt der Kunde die spezielle Ecash Software.
- Händler: muss Ecash als Zahlungsmittel akzeptieren. Damit das erkennbar war, gab es ein eigenes Logo (*We accept ecash*) und zusätzlich wurde von DigiCash eine Liste mit Vertragspartnern veröffentlicht. Der Händler konnte auch eine Privatperson sein. Auch auf Händlerseite wurde die Ecash Software benötigt.
- Banken: tauschen reales in elektronisches Geld. Weiters sind sie für die Verwaltung der Konten zuständig und garantieren den realen Gegenwert des elektronischen Geldes.

4.1.4 Anforderungen

Verifikation: die Banknote muss für jeden Händler erkennbar sein und als Zahlungsmittel akzeptiert werden.

Anonymität: es darf niemand die Möglichkeit besitzen, dass der Käufer nach dem Zahlungsvorgang rückverfolgt werden kann.

4.1.5 Ablauf und digitale Realisierung

Für die folgende Darstellung des Verfahrens von David Chaum wird angenommen, dass eine Person einen elektronischen Geldschein im Wert von 10 Euro erstellen und mit diesem bei einem Händler zahlen möchte. Die Bank der Person ist in diesem Fall befugt, dass sie elektronisches Geld erzeugt.

Schematische Darstellung:

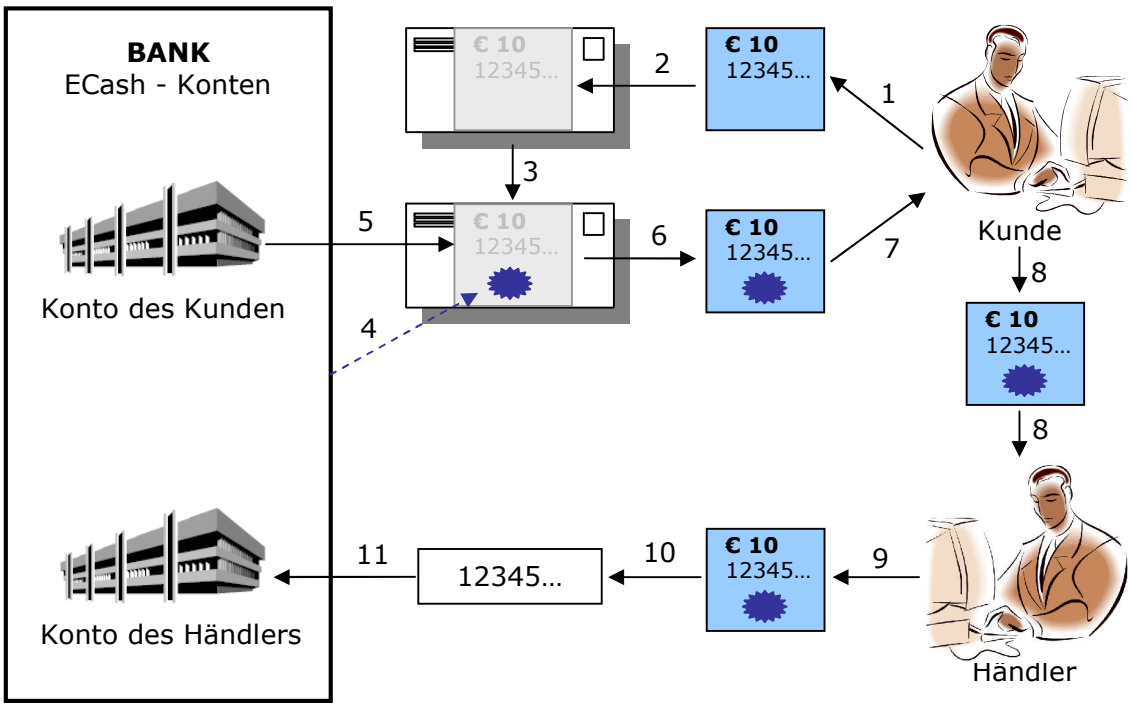


Abbildung 6: ECash Ablauf

Möchte eine Person A (in diesem Fall der Kunde) eine Bezahlung mittels Ecash bezahlen, so muss sie bei der Bank zuerst eine entsprechende digitale Münze erzeugen lassen. Dazu braucht sie nur den gewünschten Wert (laut der Darstellung 10 Euro) und eine Seriennummer, die möglichst groß sein soll (z.B.: 64 bit), auf einen Zettel schreiben. Über diesen Zettel wird ein Blatt Kohlepapier gelegt und dann beides in einen Umschlag gegeben. Der geschlossene Umschlag wird der Bank übergeben, damit diese diesen unterschreibt. Die Bank erfährt nur den gewünschten Betrag, wodurch im weiteren Verlauf sicher gestellt wird, dass der Kunde mit dem Geldschein nicht in Verbindung gebracht werden kann, da nirgends eine Registrierung der Seriennummer mit dem Besitzer vorgenommen wurde. Die Bank signiert den Umschlag ohne zu wissen, dass sich tatsächlich der genannte Betrag darin befindet, wodurch sich durch das Kohlepapier die Signatur auch auf den Geldschein überträgt. Bevor die Bank den Umschlag wieder an den Kunden zurückgibt, wird das Konto des Kunden mit dem gewünschten Betrag belastet. Nun kann der Kunde den Geldschein aus dem Umschlag nehmen und besitzt ein gültiges Zahlungsmittel, mit dem er bei jedem Händler, der dieses Zahlungsmittel akzeptiert, bezahlen kann. Möchte der Kunde nun etwas bezahlen, so schickt er den Geldschein einfach an die Person B (in dem Fall der Händler). Nun hat der Händler die Möglichkeit, dass er den Geldschein behält und diesen selber einmal für einen Einkauf verwendet, oder er löst ihn bei der Bank gegen reales Geld ein. Entschließt er sich für Letzteres, so sendet er den Geldschein einfach an die Bank. Diese überprüft die Seriennummer des Geldscheines. Ist in der Datenbank die Nummer schon registriert, so nimmt die Bank den Geldschein nicht als gültiges Zahlungsmittel an. Falls die Seriennummer noch nicht in der Datenbank registriert ist, so wird der Betrag auf das Konto des Händlers gutgeschrieben und die Seriennummer in die Datenbank eingefügt.

4.1.6 Digitale Umsetzung

Möchte eine Person A eine elektronische Münze beliebigen Wertes anfordern, so muss sie zunächst zwei große Zufallszahlen (C und V) wählen. Dabei dient die Zahl C zur Verschleierung der Kommunikation zwischen Bank und Person und V dient später zur Erzeugung des Bitstrings.
Jetzt muss die Person A die Zahl W bilden, indem sie einfach die Zahl V zweimal hintereinander aufschreibt.

$$W = VconcatV$$

Der nächste Schritt beinhaltet die Verschlüsselung von W mit dem öffentlichen Schlüssel der Bank.

$$S := C^e \cdot W \bmod n$$

Die erhaltene Zahl S schickt die Person A nun an die Bank, die diese in eine elektrische Geldmünze des entsprechenden Wertes umwandelt. Dazu muss die Person vorher Geld auf das Konto eingezahlt haben, damit die Bank den gewünschten Geldbetrag abbuchen kann.
Nach getätigter Abbuchung signiert die Bank die Zahl S mit ihrem persönlichen Schlüssel.

$$T := S^d \bmod n$$

Den Wert T schickt die Bank nun ihrem Kunden, der Person A, zurück. Diese kann in der weiteren Folge die Korrektheit der erhaltenen Daten überprüfen. Dazu wendet sie auf die Zahl T den öffentlichen Schlüssel der Bank an. Sind die Daten korrekt, so muss gelten:

$$T^e = S^{de} = S$$

Damit die Person die elektronische Münze erhält, muss sie den Verschleierungsfaktor C wieder entfernen.

$$F := T \cdot C^{-1} \bmod n$$

Nun ist die Person A im Besitz der elektronischen Geldmünze F.

Mit der folgenden Umformung soll gezeigt werden, dass F nicht vom Verschleierungsfaktor C, sondern nur vom geheimen Schlüssel d der Bank und von V abhängt. Da in der Praxis niemand in den Besitz dieser beiden Werte kommen kann ist dies auch kein Sicherheitsproblem.

$$F := T \cdot C^{-1} \bmod n = S^d \cdot C^{-1} \bmod n = C^{ed} \cdot W^d \cdot C^{-1} \bmod n = C \cdot W^d \cdot C^{-1} \bmod n = W^d \bmod n$$

Damit andere Personen die Echtheit der elektronischen Geldmünze überprüfen können, muss folgendes berechnet werden:

$$F^e \bmod n = W^{de} \bmod n = W \bmod n$$

Erhält man auf beiden Seiten von $F^e \bmod n$ ein identisches Ergebnis, so kann man mit absoluter Sicherheit davon ausgehen, dass F eine elektronische Geldmünze ist. Aufgrund der Mächtigkeit der Exponentialfunktion kann man aus F diese Information noch nicht gewinnen. Das heißt es ist nicht möglich eine Banknote herzustellen, ohne den geheimen Schlüssel der Bank zu kennen.
Die Anonymität des Bezahlenden ist gegeben, da keine direkte Verbindung zwischen der Geldmünze und der Person hergestellt werden kann. Auch die Bank hat keine diesbezüglichen Informationen, da in allen ihr bekannten Zahlen der Verschleierungsfaktor C steckt:
Zum einen kennt sie die Geldmünze F nicht, sondern nur

$$S = C^e \cdot W \bmod n$$

und zum anderen

$$T = C^{ed} \cdot W^d = C \cdot W^d \bmod n$$

wodurch sie keine Möglichkeit besitzt, um aus S und T die Geldmünze F zu berechnen.
Aufgrund dieser Tatsache wurde für den Kunde die vollständige Privatsphäre als auch ein hohes Maß an Sicherheit für die Bank realisiert.

4.1.7 Verbesserungs- und Ausbaustufen

Ein großes Problem stellt für die Bank die Tatsache dar, dass sie sich auf die Angaben des Antragstellers verlassen muss. Im Grunde könnte dieser der Bank sagen, dass er einen elektronischen 10 Euro Geldschein anfordert, obwohl er sich eigentlich einen 100 Euro Geldschein signieren lässt. Da die Bank den Geldschein nie zu Gesicht bekommt, könnte sie das nicht kontrollieren und würde so weniger vom Konto abbuchen als sie sollte. Dieses Problem könnte man lösen, indem der Kunde der Bank nicht einen Umschlag sondern zum Beispiel 100 Umschläge schickt. In jedem Umschlag ist ein vorbereiteter Geldschein, wobei alle denselben Wert haben. Die Bank öffnet von den 100 Umschlägen nur 99 und wenn alle denselben Wert haben, kann die Bank mit ziemlicher Sicherheit davon ausgehen, dass auch in diesem Umschlag derselbe Wert ist und signiert diesen. Sollte der Kunde ein Ganove sein, so hat er dadurch nur eine Wahrscheinlichkeit von 1:100, dass sein Betrug nicht auffällt. Um auch diese kleine Chance für Betrüger uninteressant zu machen, müssen von der Gesetzgebung entsprechend hohe Strafen auf solche Vergehen ausgearbeitet werden.

Ein weiteres Problem ist die mögliche Bereicherung von Personen durch die Erstellung von Duplikaten. Mit dem vorher beschriebenen System ist es nicht möglich Duplikate zu enttarnen. Möglich wäre eine Lösung, bei der der Kunde auf den Geldschein einen Zufallsstring schreibt. Dieser String sollte

möglichst groß sein, damit die Wahrscheinlichkeit vernachlässigbar klein ist, dass eine andere Person genau denselben String verwendet. Nachdem der Geldschein bei der Bank eingelöst wurde, wird der String in einer Datenbank gespeichert. Sobald versucht wird, den Geldschein nochmals einzulösen wird der String in der Datenbank entdeckt und die mehrfache Verwendung ist aufgedeckt.
Ein Duplikat zu erkennen ist sehr wichtig, aber nur das Erkennen ist nicht optimal. Es sollte auch möglich sein, dass man den Betrüger ausfindig macht, d.h. man muss erkennen können ob der Kunde oder der Händler der Täter ist. Aus diesem Grund wurde ein sogenannter Identifikationsstring eingeführt. Dieser wird von der bezahlenden Person auf den Geldschein geschrieben, nachdem der Empfänger die Bankunterschrift überprüft hat. Über den Identifikationsstring kann später herausgefunden werden, wer den Geldschein vervielfacht hat. Dabei gibt es zwei Szenarien, bei denen jeweils der Geldschein bereits in der Datenbank der Bank registriert ist und die Bank daher die Annahme des Geldscheins verweigert. Der erste Fall ist, dass der Identifikationsstring auf dem Geldschein ein anderer als in der Datenbank ist. Dadurch ist der Bezahlende der Übeltäter, da der Verkäufer den Geldschein erst kopieren kann, wenn der String bereits am Geldschein ist. Das ist dann bereits die andere Möglichkeit und zwar wenn die Strings in der Datenbank und auf dem Geldschein ident sind. Als Täter nimmt die Bank hier den Händler an.

Dieses System hat den gravierenden Nachteil, dass immer der Händler als Täter angenommen wird, sobald die Identifikationsstrings in der Datenbank und auf dem Geldschein ident sind. Ist der Kunde jedoch der Täter und verwendet immer den gleichen String, so kann ihm dies nicht nachgewiesen werden. Eine Verbesserung der Täterenttarnung wurde mit dem Verfahren des Secret Splitting realisiert.

4.1.8 Secret Splitting

Beim Secret Splitting wird eine beliebige Nachricht verschlüsselt und in Stringpaare zerlegt. Die Nachricht ist erst dann wieder lesbar, wenn man

alle Teile besitzt. Hat man nur die Hälfte der Stringpaare, so kann man diese zwar entschlüsseln, aber sie ergeben nicht die Daten selbst.
Bei elektronischen Geldscheinen wird dieser String so gewählt, dass er die Identität der Person enthält, die den Schein bei der Bank anfordert. Zur Veranschaulichung nehmen wir an, dass folgender String gewählt wurde: „Meine Name ist Alice und wohne in der Musterstraße 1 in 1010 Wien". Mittels eines Secret Splitting Protokolls erzeugt Alice x-Mal das Tupel (I1,I2). Auf jeden Geldschein werden (x/Anzahl der Scheine) dieser Tupel geschrieben. Die Zahlen x und y sollten dabei relativ groß sein, für die weitere Ausführung nehmen wir als 10000 an, wodurch sich durch eine Scheinanzahl von 100 ergibt, dass jeweils 100 Tupel auf einem Schein stehen.
Die Geldscheine werden dann wie gewohnt an die Bank geschickt. Die Bank öffnet wieder alle Scheine bis auf einen und lässt sich von Alice alle Tupel entschlüsseln. Die entschlüsselten Tupel müssen nun alle die gleiche Information ergeben und diese muss auch korrekt sein. Die Korrektheit kann von der Bank kontrolliert werden, da sie ihren Kunden kennt und dieser ein Konto besitzt. Ergibt sich überall die idente Information so signiert die Bank den letzten, nicht entschlüsselten, Geldschein. Diesen schickt sie zurück zu Alice, die mit diesem eine Bezahlung vornimmt. Die den Geldschein erhaltende Person überprüft zuerst die Signatur der Bank. Ist diese korrekt, so bittet sie den Bezahlenden um die Entschlüsselung der halben Anzahl der Tupel. Welche das sind wird von einem Zufallsgenerator ermittelt. Mit diesen entschlüsselten Tupel erhält der Händler keine Information über die Identität seines Kunden. Möchte nun der Händler den Geldschein bei der Bank einlösen, so schickt er diesen mit den entschlüsselten Tupeln an die Bank. Diese überprüft, ob der Schein schon einmal verwendet wurde. Wenn ja verweigert die Bank die Annahme des Scheins und kann zusätzlich feststellen wer den Betrug versucht hat. In der Datenbank ist auch schon die per Zufallsgenerator entschlüsselte Hälfte der Information gespeichert und es genügt ein vollständiges Tupel, damit die Information lesbar wird.

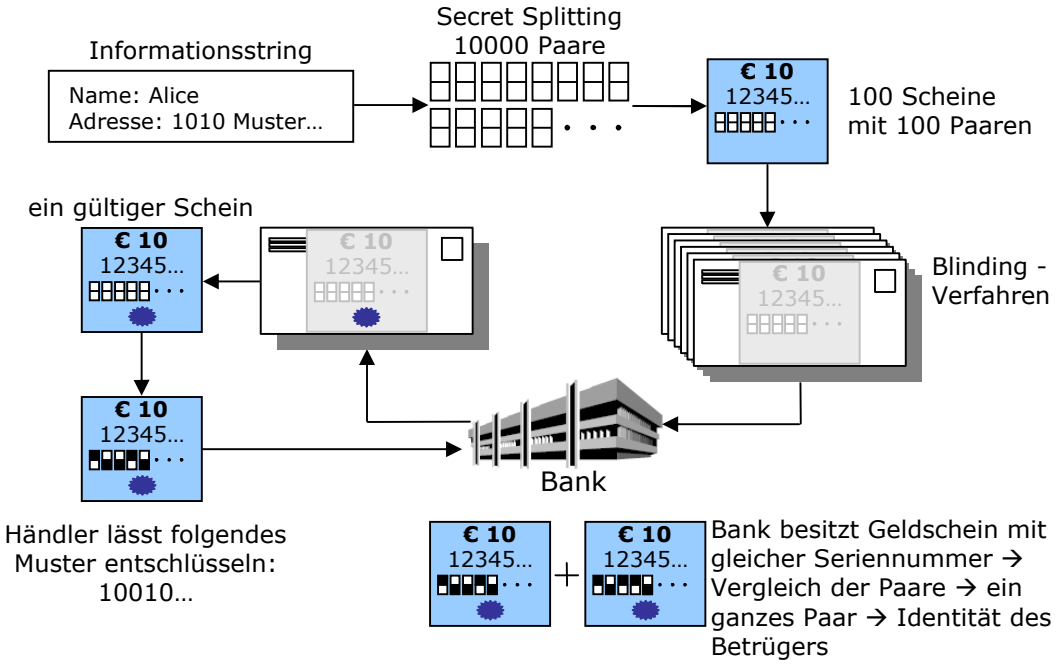


Abbildung 7: Ablauf Secret Splitting

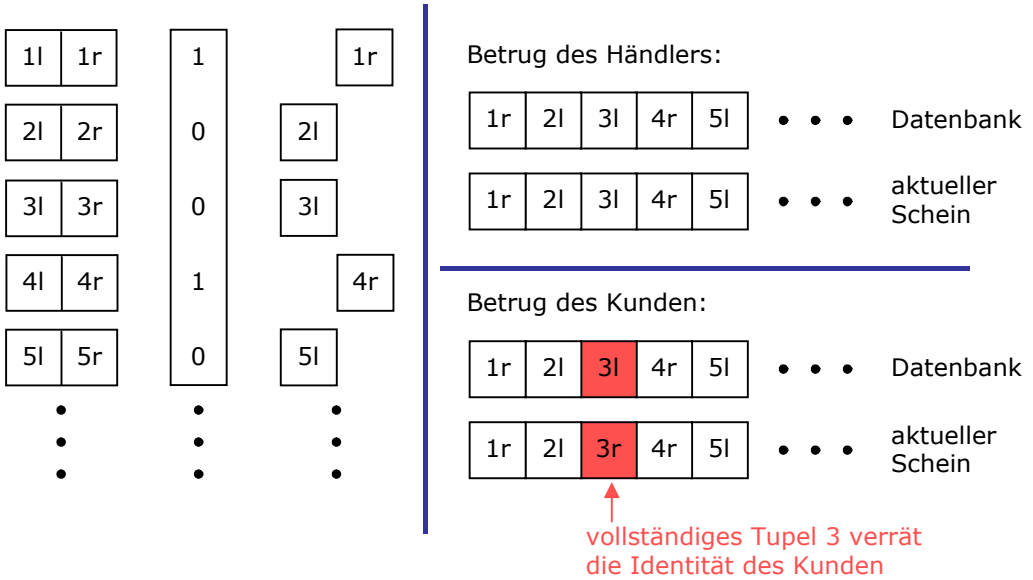


Abbildung 8: Beispiel für Secret Splitting

4.2 SET – Secure Electronic Transaction

Um einen sicheren Zahlungsverkehr über das Internet zu ermöglichen wurde das SET Protokoll von einer Vereinigung von Kreditkartengesellschaften, mit Vorsitz von Mastercard und Visa, entwickelt. Durch diese breite Unterstützung von einflussreichen Unternehmen sollte von Beginn an sichergestellt werden, dass sich dieses Protokoll etablieren kann. Weiters ist durch die Mitarbeit von Microsoft auch eine Integration in den am häufigsten verwendeten Webbrowser sichergestellt.
Das große Ziel aller beteiligten Unternehmen ist es, einen weltweiten Standard aufzubauen, der eine zuverlässige und sichere Abwicklung des Zahlungsverkehrs über das Internet ermöglichen soll. So soll es zu einer Vergrößerung der Akzeptanz von Kreditkarten im Internet kommen.

Das SET Protokoll realisiert drei zentrale Funktionen:

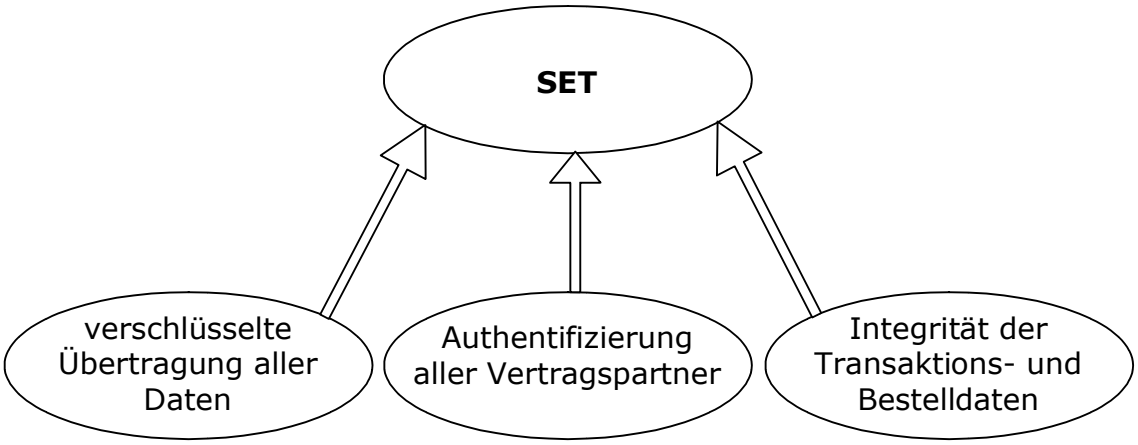


Abbildung 9: Zentrale Funktionen von SET

4.2.1 Ablauf

1. Kaufanfrage (purchase request): In diesem Abschnitt schickt der Kunde seine Bestellung an den Händler und dieser schickt die Rechnung zurück. Um diesen Vorgang nach den Vorgaben des SET Protokolls ablaufen zu lassen, müssen die beiden Parteien wie folgt vorgehen. Vom Kunden wird eine Initialisierungsnachricht an den

gewünschten Händler geschickt. Dieser erstellt eine, mit seinem Signatur-Zertifikat unterschriebene, Antwort und schickt diese gemeinsam mit dem zertifizierten Schlüssel seiner Kreditanstalt zurück an den Kunden. Der Kunde kann jetzt seinerseits die Gültigkeit der erhaltenen Zertifikate überprüfen. Fällt die Überprüfung positiv aus, so kann der Kunde davon ausgehen, dass der Händler vertrauenswürdig ist und der Zahlungsvorgang kann weitergeführt werden. Im nächsten Schritt werden die Bestellung und die Zahlungsinformationen vom Kunden mit der dualen Signatur versehen. Dabei wird die Zahlungsanweisung mittels DES verschlüsselt und dieser Schlüssel wird gemeinsam mit den Kreditkarteninformationen vom Kunden mittels RSA verschlüsselt. Als RSA Schlüssel wird der des Finanzinstitutes gewählt. Das verschlüsselte Paket wird nun vom Kunden an den Händler übertragen. Zusätzlich kann vom Kunden noch ein Zertifikat mitübertragen werden, sofern dieser eines besitzt. Hat der Händler die verschlüsselten Informationen erhalten, so schickt er eine Quittung zurück an den Kunden, der wiederum die Zertifikate überprüft und die Quittung abspeichert.

2. Zahlungsfähigkeitsüberprüfung: In diesem Abschnitt des Ablaufs geht es im Prinzip darum, dass der Händler von seiner Bank bestätigt bekommt, dass sein Kunde zahlungsfähig ist. Der Händler ist in der Lage, die DES Verschlüsselung aufzuheben, d.h. er kommt an die unverschlüsselten Bestelldaten nicht aber an die Kreditkarteninformationen seines Kunden. Darum schickt er die Zahlungsanweisung und zusätzlich eine Zahlungsfähigkeitsanfrage an das zuständige Finanzinstitut. Dort werden wieder alle Zertifikate überprüft und die Zahlungsanweisung des Kunden mit den angegebenen Daten des Händlers verglichen. Haben alle Tests ein positives Ergebnis, so übermittelt das Finanzinstitut an die Kundenbank die Anfrage, ob diese Zahlung in Ordnung geht. Die erhaltene Antwort übermittelt das Finanzinstitut an den Händler zurück und dieser verschickt seine Ware an den Kunden.

3. Zahlungsdurchführung: Der Händler schickt eine Bestätigung an das Kreditkartenunternehmen, welches sich um die weiteren Schritte kümmert. Zum einen veranlasst es eine Abbuchung am Konto des Kunden und zum anderen wird dem Händler die Summe minus den Spesen auf sein Konto verbucht. Die Abwicklung der Bezahlung zwischen dem Kreditkarteninstitut und den beiden Banken wird dabei nicht über das Internet, sondern über ein eigenes Bankennetz abgewickelt.

Schematische Darstellung der Abschnitte:

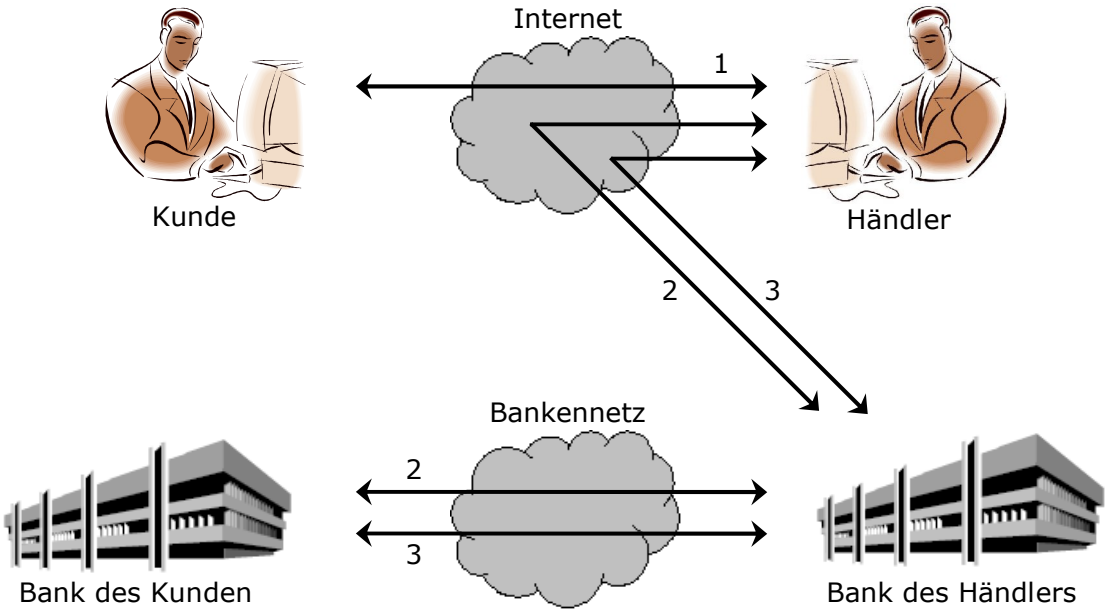


Abbildung 10: Abschnitte bei SET

Schematische Darstellung mit den einzelnen Schritten:

Abbildung 11: Funktionsablauf von SET

4.2.2 Probleme bei SET

Der Hauptgrund, dass nur sehr wenige Händler die Bezahlung mittels SET-Protokoll anbieten, ist, dass der Aufwand sowohl auf Händler- als auch auf Kundenseite enorm ist. Der Kunde muss ein eigenes Plug-In auf seinem Computer installiert haben, wodurch erst eine elektronische Brieftasche angelegt wird, indem sich auch das Kundenzertifikat befindet. Geschützt sind die Daten mittels Passwort. Bevor der Kunde jedoch die Software in Betrieb nehmen kann, muss er an seine Bank einen schriftlichen Antrag schicken, um für seine Kreditkarte einen SET-Code zu erhalten. Nun hat die Bank die Aufgabe, sich um den Zertifizierungsantrag weiter zu kümmern und schickt daher den Antrag an die entsprechende Kreditkartengesellschaft weiter. Diese erteilt den Auftrag zur Generierung des Zertifikats an ein Trust Center. Dieses Trust Center generiert das geforderte Zertifikat und

benachrichtigt im Gegenzug die Kreditkartengesellschaft, die wiederum die Informationen an die Bank weiterleitet. Nun kann die Bank an seinen Kunden den Zertifikatsbrief (bestenfalls gemeinsam mit der erforderlichen Software) senden. Erst jetzt kann der Kunde die Software installieren und dann das Zertifikat aktivieren. Das funktioniert meist über eine vorgegebene Internetseite, auf der man mit Hilfe eines geheimen Codes eine Aktivierungsmeldung (Wake-Up Message) erhält.
Der Händler braucht auf seinem Server eine SET – Handelssoftware, die von der Firma „SET Secure Electronic Transaction LLC" (SETCo) zertifiziert sein muss. Bei kleineren Unternehmen ist eine Auslagerung meist sinnvoller. Dazu muss der Server des Unternehmens über ein Netzwerk mit einem SET-Gateway Server verbunden sein.
Dieser komplizierte Ablauf hat hohe Transaktionskosten zur Folge, wodurch diese Zahlungsmethode nicht für kleine Geldbeträge sinnvoll ist.

4.3 Kreditkartenbezahlung mit SSL

Um im Internet mit Kreditkarte bezahlen zu können muss zum einen der Kunde eine Karte besitzen und zum anderen muss der Händler einen Vertrag mit der Kreditkartengesellschaft haben. Sind diese beiden Vorraussetzungen erfüllt, so genügt die Eingabe der Kreditkartennummer und des Ablaufdatums der Karte um zu bezahlen. Da beim Onlineeinkauf der Händler die Karte nicht in die Hände bekommt und kontrollieren kann ob die Unterschrift richtig ist, müssen die Daten bei der Übertragung vor Dritten geschützt sein. Um die Sicherheit zu gewährleisten werden die Daten mittels SSL verschlüsselt.

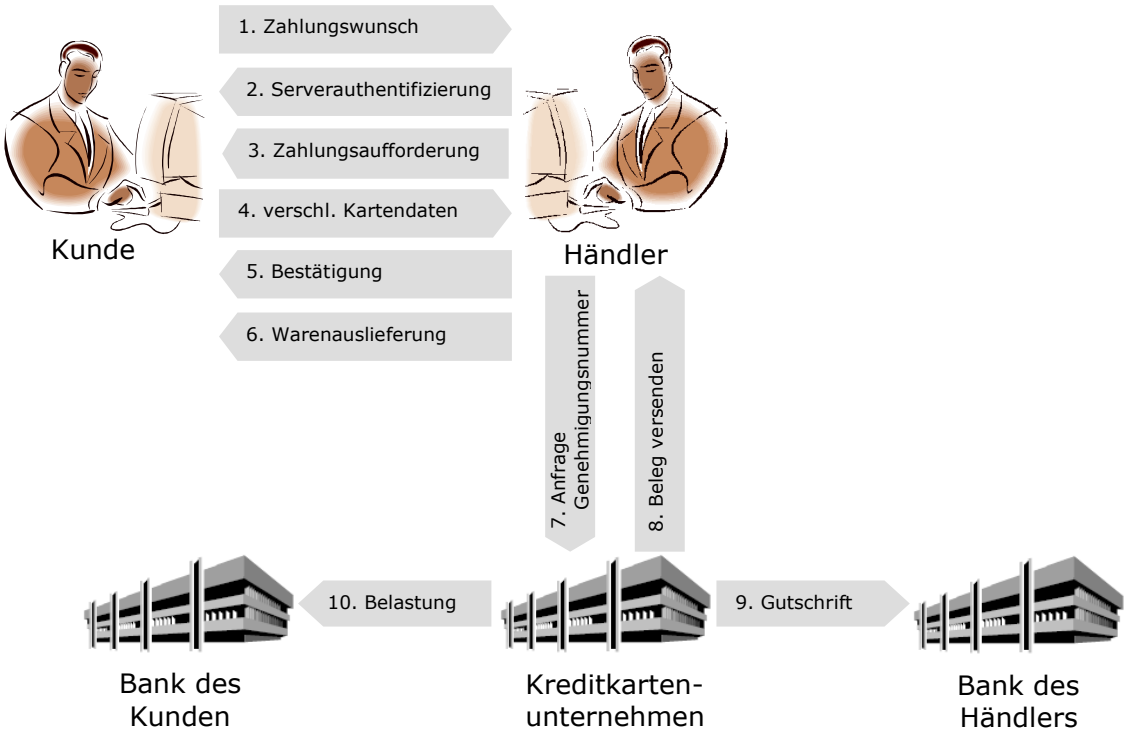


Abbildung 12: Ablauf einer Kreditkartenzahlung mit SSL

Kapitel II
Maschinenlesbare Reisedokumente

1. Entwicklung

Für die Entwicklung des Standards für die Einführung eines Machine Readable Travel Documents (engl. MRTD, deutsch: maschinenlesbare Reisedokumente) ist die International Civil Aviation Organization (engl. ICAO, deutsch: Internationale Zivilluftfahrt-Organisation) haupt-verantwortlich. Die ICAO ist eine Unterorganisation der Vereinten Nationen und befasst sich seit dem Jahre 1997 mit der Einführung eines einheitlichen, elektronisch auswertbaren und mit biometrischen Merkmalen ausgestatteten Reisedokument.
2003 wurde von der ICAO eine Empfehlung unter dem Namen Blueprint verabschiedet. Darin empfiehlt die ICAO allen UN-Mitgliedsstaaten, die Integration von biometrischen Merkmalen in den Reisepässen. Geeignete Techniken müssen dabei folgende fünf Kriterien erfüllen:

- weltweite Interoperabilität
- Einheitlichkeit
- technische Zuverlässigkeit
- Praktikabilität
- Haltbarkeit

Bei der Verwendung von Blueprint stehen vier zentrale Punkte im Mittelpunkt:

- Verwendung kontaktloser Chips (RFID)
- Chip dient zur Abspeicherung des Lichtbildes, eine Ergänzung weiterer Merkmale (z.B. Fingerabdruck, Irismuster) soll zu einem späteren Zeitpunkt möglich sein
- Verwendung einer definierten, logischen Datenstruktur (engl. Logical Data Structure, LDS)
- Public Key Infrastructure (PKI): Verfahren zur Verwaltung digitaler Zugangsschlüssel

Im weiterentwickelten Standard 9303 wurden von der ICAO alle Vorgaben zusammengefasst.
Von den USA ging immer mehr Druck aus, wodurch am 13. Dezember 2004 im Rat der Europäischen Union beschlossen wurde, dass die Mitgliedsstaaten alle Pässe nach dem Standard 9303 mit maschinenlesbaren biometrischen Daten ausstatten. Als Merkmal wurden dafür die Abdrücke der Zeigefinger beider Hände ausgewählt. Diese werden mittels Scanner eingelesen und auf den Chip gespeichert.
In Deutschland werden seit dem 1. November 2007 alle neu ausgestellten Reisepässe mit den biometrischen Daten ausgestattet. In Österreich ist noch kein genauer Termin zur Einführung bekannt.

2. Sicherheitsmechanismen beim ePass

Nach der ICAO müssen alle maschinenlesbaren Reisedokumente über folgende vier Sicherheitsmaßnahmen verfügen, damit ein absoluter Schutz gegen Missbrauch vorhanden ist:

- Passive Authentication (PA): das Lesegerät kann überprüfen, ob das Reisedokument von einer zulässigen Zertifizierungsstelle signiert wurde. Damit kann die Gültigkeit des Reisedokuments bestimmt werden.
- Basic Access Control (BAC): Schützt das Reisedokument vor unberechtigten Lesezugriffen. Somit können Angreifer das Reisedokument nicht unbemerkt auslesen und eventuell kopieren.
- Active Authentication (AA): Schützt die Einzigartigkeit und die Authentizität des im Reisedokument befindlichen Chips. Diese Sicherheitsmaßnahme sollte nur verwendet werden, wenn BAC bereits integriert ist und somit die Kommunikation in diesem Schritt entsprechend verschlüsselt wird.

- Extended Access Control (EAC): Ist ein Mechanismus, mit dem ermöglicht wird, dass auf gewisse Inhalte nur gewisse Lesegeräte Zugriff haben. Eingeführt wurde EAC um die, neu am Chip abgespeicherten, biometrischen Daten gegen unerlaubten Zugriff zu schützen.

3. Logical Data Structure

3.1 Allgemeines

Die Logical Data Structure (LDS) kann als logische Datenstruktur übersetzt werden. Gemeint ist damit die Speicherverteilung der Daten auf dem RF-Chip. Benötigt wird der Speicherstandard, um weltweite Interoperabilität gewährleisten zu können. Der größte Vorteil des Konzepts der LDS ist die Möglichkeit, dass die Speicherkapazität jederzeit erhöht werden kann.

3.2 Aufbau

Der Speicher am Chip ist in so genannte Data Groups (Datengruppen) unterteilt. Aktuell besteht die LDS aus 16 Datengruppen, wobei für die Zukunft eine Erweiterung geplant ist, um zum Beispiel Visadaten zu speichern. Bei der Einführung der maschinenlesbaren Reisepässe waren nur die Datengruppen 1 (maschinenlesbare Daten) und 2 (Gesichtbild) verpflichtend.

3.2.1 LDS Datengruppe 1

Bei den, in der DG1 gespeicherten Daten, handelt es sich um die Daten, die auch mit dem Auge erkennbaren sind. Das heißt es werden alle Daten in DG1 gespeichert, die auch bereits auf den alten Reisepässen abgedruckt waren. Dazu gehören:

- Dokumententyp
- Ausstellungsland
- Persönliche Informationen (Nachname, Vorname, Nationalität, Geburtsdatum, Geschlecht)

- Ablaufdatum
- zusätzliche Daten
- Ausweisnummer

3.2.2 LDS Datengruppe 2

In der DG2 wird das, auf den Reisepass gedruckten, Foto des Passinhabers digital abgespeichert. Um den Speicheraufwand möglichst gering zu halten wird das Foto komprimiert abgespeichert. Zusätzlich zum Foto werden noch weitere Informationen dazu abgespeichert. Da dieses Foto in der weiteren Folge automatisch von Gesichtserkennungssoftware verwendet werden soll, muss das Foto einer gewissen Norm entsprechen. Die deutsche Bundesdruckerei GmbH hat eine Schablone veröffentlicht, mit der man vorhandene Fotos auf Gültigkeit überprüfen kann.

3.2.3 LDS weitere Datengruppen

In der neuesten Generation des Reisepasses werden auch biometrische Daten auf den Chip gespeichert. Vorerst sollen nur die Fingerabdrücke verwendet werden und diese werden in die DG3 abgespeichert. Für die Zukunft wurde die DG4 bereits für das Abbild der Iris vorgesehen.

Noch nicht implementiert, aber bereits für die Zukunft beschlossen, sind die Datengruppen 17-19. Diese Gruppen sollen einmal für die Abspeicherung von Visadaten verwendet werden.
Die genaue Speicheraufteilung der einzelnen Datengruppen ist in „*ICAO Doc 9303 Part 1 Volume 2 E-passports*“ (Seite III-8 bis III-21) zu finden. Bei dieser Auflistung wird auch darauf eingegangen, welche Daten verpflichtend und welche optional sind und welches Format die Datenelemente haben müssen.
Im Zusammenhang mit dem Schwerpunkt dieser Arbeit ist noch die DG15 erwähnenswert. In dieser Datengruppe ist der öffentliche Schlüssel abgelegt, der für die Active Authentication benötigt wird.

3.2.4 Darstellung der LDS

In der nächsten Abbildung ist der, laut Spezifikation in ICAO (2005), vorgegebene Aufbau der LDS veranschaulicht.

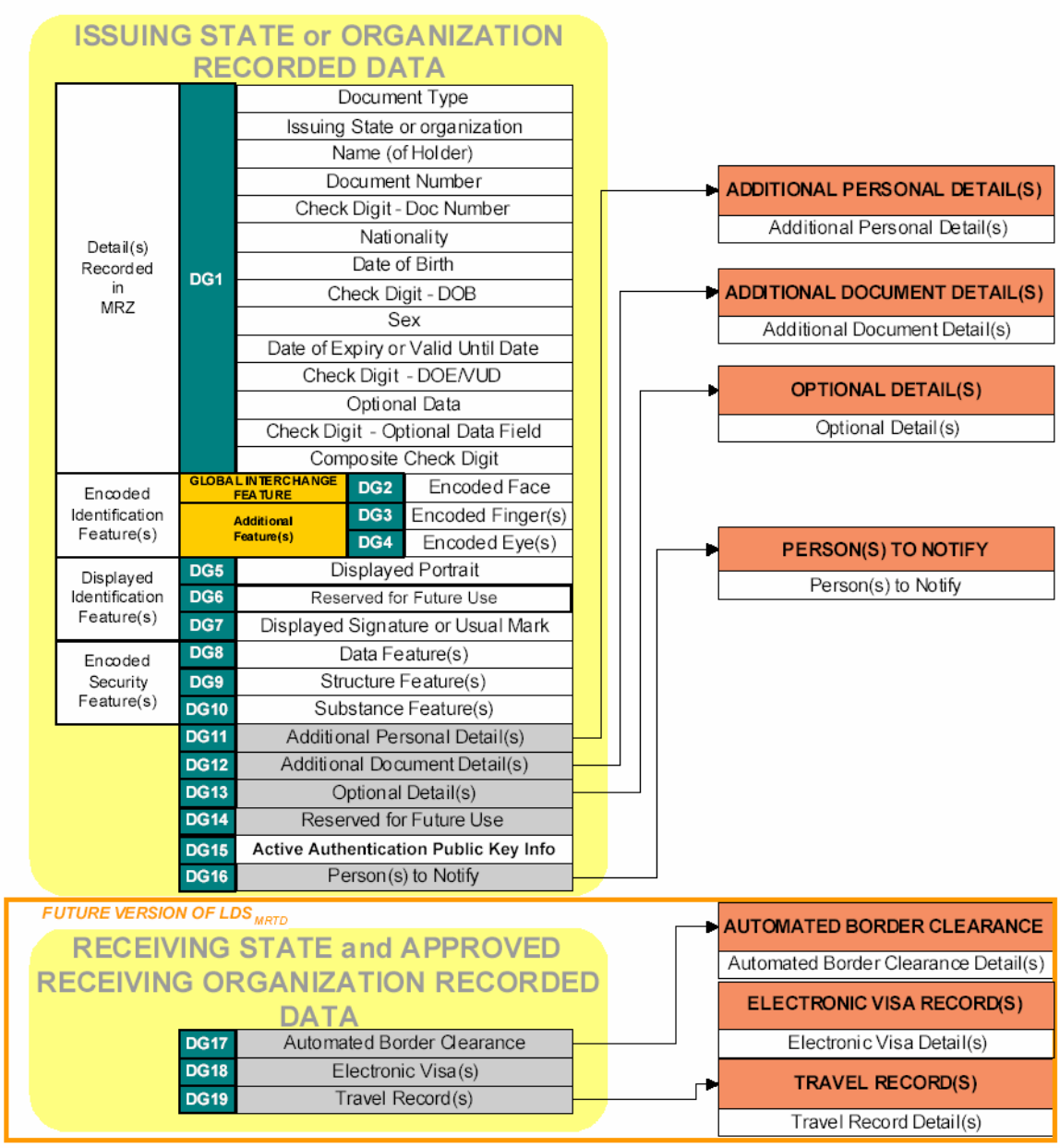


Abbildung 13: Aufbau der LDS

Für die weltweite Interoperabilität wurde im Bezug zur LDS beschlossen, dass eine zufällige Reihenfolge der Datengruppen auf dem Chip zu realisieren ist. Damit das Lesegerät beim Lesevorgang die richtigen Daten erfassen kann musste eine Lösung gefunden werden, um dem Inspektionsgerät mitzuteilen, wo sich welche Daten befinden. Dazu wurde ein System entwickelt, bei dem das Lesegerät zu Beginn einen Header und eine Data Group Presence Map ausliest.

Im Header befinden sich folgende Daten:

- Anwendungsidentifikation
- LDS Versionsnummer
- UNICODE Versionsnummer

Die Präsenzliste der Datengruppen besteht aus einer variablen Anzahl von Tags. Jeder dieser Tags steht für eine Datengruppe. Ist für eine Datengruppe kein Tag vorhanden, so befinden sich auch keine entsprechenden Daten auf dem Chip. Header und Presence Map müssen vorhanden sein.

Eine schematische Darstellung gibt die folgende Abbildung aus ICAO (2005).

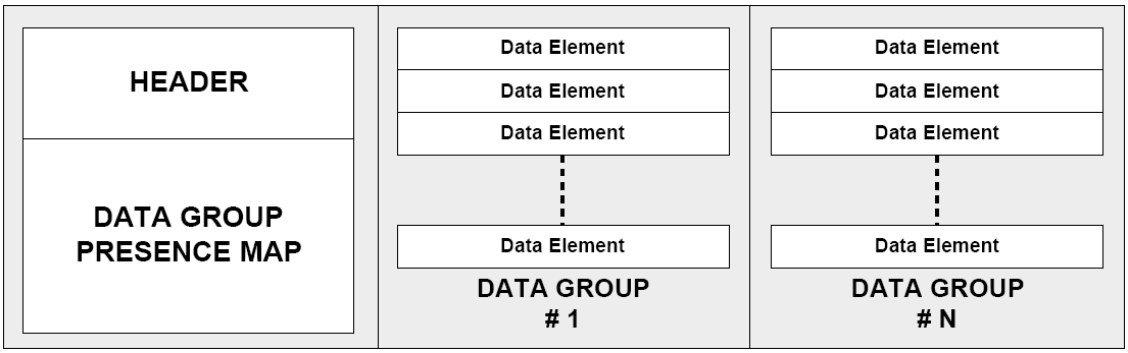


Abbildung 14: Header und Data Group Presence Information

Innerhalb der einzelnen Datengruppen arbeitet ein ähnliches Schema. Zu Beginn jeder Datengruppe ist eine so genannte Data Element Presence Map zu finden. Diese gibt Aufschluss über die, in dieser Gruppe, vorhandenen Datenelemente. In der folgenden Grafik aus ICAO (2005) wird das Schema dargestellt.

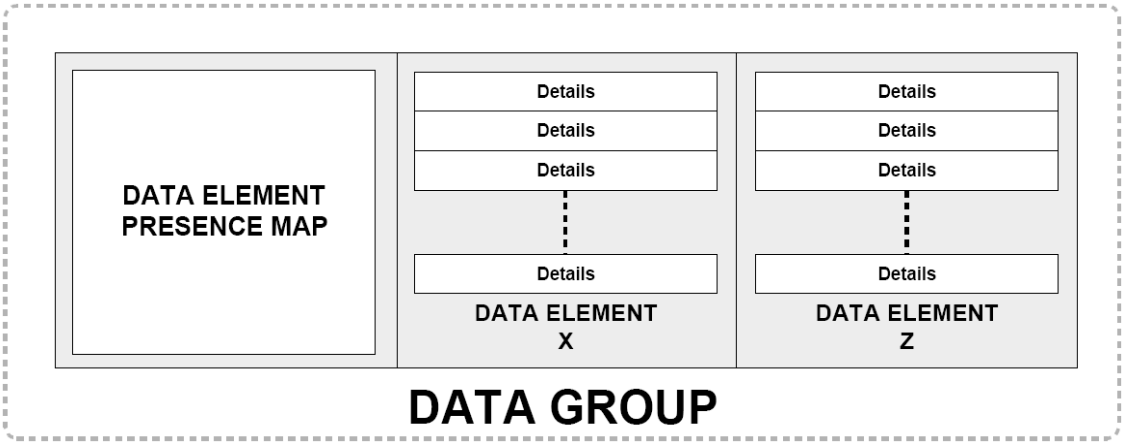


Abbildung 15: Data Element Presence Map

Um die Sicherheit der Daten zu gewährleisten, wird von der ICAO empfohlen, dass die Datengruppen 1 bis inklusive 16 schreibgeschützt sein sollen. Zusätzlich sollen digitale Signaturen über die Hashwerte in den Sicherheitsdaten (Security Data) abgespeichert werden. Auf diese Gruppen soll nur das herausgebende Land Schreibzugriff haben.
Die Spezifikation der Datengruppen 17 bis 19 wird erst in der 2. Version der Spezifikation der LDS definiert.

4. Extended Access Control

4.1 Allgemeines

Nachdem der Beschluss gefasst war, dass in der 2. Ausbaustufe auch biometrische Daten auf dem Reisepass gespeichert werden sollen, musste eine neue Methode zur Absicherung dieser Daten entwickelt werden. Diese Methode wird Extended Access Control (EAC) genannt. Der erweiterte Zugriffsschutz wurde durch eine Arbeitsgruppe, aufgrund einer technischen Richtlinie des Bundesamtes für Sicherheit in der Informationstechnik, entwickelt.
Biometrische Daten sind sensitive personenbezogene Daten und benötigen einen starken Schutz gegen unberechtigte Zugriffe. Genau dies hat die EAC als Hauptaufgabe. Zum Einen sollen nur nationale berechtigte Lesegeräte Zugriff auf die biometrischen Daten erhalten und zum Anderen soll auch für ausgewählten Kontrollsystemen anderer Ländern die Zugriffsmöglichkeit realisiert werden.
Für die Erfassung wurden die Abdrücke der beiden Zeigefinger gewählt. Diese werden jeweils in dreifacher Ausführung mittels Scanner eingelesen.

4.2 Grundprinzip

Das Grundprinzip von EAC ist, dass sich das Lesegerät beim, im Reisepass integrierten, Chip authentifizieren muss bevor es Zugang zu den sensiblen Daten bekommt. Dieser Schritt wird als Terminal-Authentisierung bezeichnet und muss nur dann zwingend durchlaufen werden, wenn ein Zugriff auf die biometrischen Daten erforderlich bzw. erwünscht ist. Die Terminal-Authentisierung basiert auf einer Public-Key-Infrastruktur (PKI) und wird erst nach erfolgreicher Chip-Authentisierung ausgeführt. Die Chip-Authentisierung ist bereits Teil der Basic Access Control (BAC) und ist für folgende zwei Dinge verantwortlich:

- Echtheitsüberprüfung des Chips
- Aufbau einer stark verschlüsselten Kommunikation zwischen Chip und Terminal

Aus dieser Reihenfolge ergibt sich, dass alle übertragenen persönlichen Daten nur verschlüsselt übertragen werden. Bei der Entwicklung wurde beschlossen, dass in der Europäischen Union alle Lesegeräte so realisiert werden müssen, dass die Chip-Authentisierung zwingend durchgeführt wird.

4.3 Erweiterte Sicherheitsmechanismen

In diesem Abschnitt werden zwei erweiterte Sicherheitsmechanismen erläutert, die benötigt werden um neue maschinenlesbare Reisedokumente ausreichend abzusichern. Die beiden Mechanismen sind:

- Chip Authentication (CA)
- Terminal Authentication (TA)

Chip Authentication kann als selbstständiges Protokoll eingesetzt werden. Terminal Authentication hingegen kann nur in Kombination mit Chip Authentication verwendet werden. Die Kombination der beiden Protokolle ist die Umsetzung der Extended Access Control.

4.3.1 Chip Authentication

Bei diesem Protokoll handelt es sich um eine Alternative zur optionalen Active Authentication (AA). Es ist also für die Überprüfung des RF-Chips zuständig und bietet zudem zwei Vorteile gegenüber AA:

- Es wird eine implizite Echtheitsprüfung der, auf dem Chip gespeicherten, Daten gewährleistet. Dazu wird dem Chip bereits während der Personalisierung ein Schlüsselpaar zugewiesen. Von diesem Schlüsselpaar wird der private Schlüssel direkt in den Speicher des Chips und der öffentliche wird in die Datengruppe 14 der LDS (Logical Data Structure) geschrieben. Da für den Aufbau einer sicheren Kommunikation mit dem Lesegerät beide Schlüssel notwendig sind, ist nur ein echter Chip in der Lage diese herzustellen. Dadurch wird durch die Chip Authentication auch ein Kopierschutz der Speicherinhalte des Chips implementiert.

- Neben der Authentikation des Chips stellt das Protokoll auch einen starken Schlüssel für eine sichere Kommunikation zur Verfügung.

Zu beachten ist, dass jeder Chip Authentication unterstützende Chip auch kompatibel zur Basic Access Control sein muss.

4.3.2 Terminal Authentication

Damit ausschließlich für die Kontrolle zugelassene Lesegeräte die Daten vom Chip auslesen können, gibt es das Terminal Authentication Protokoll. Dabei muss sich dass Lesegerät beim Chip als berechtigt ausweisen. Erst dann genehmigt der Chip einen Zugriff auf sensible Daten wie zum Beispiel die Fingerabdrücke. Jegliche Kommunikation zwischen dem Lesegerät und dem Chip muss verschlüsselt sein. Bevor die Berechtigung des Lesegeräts überprüft wird muss zuerst der Chip auf Gültigkeit überprüft worden sein.

4.4 Inspektionsprozedur

Es werden zwei verschiedene Inspektionsprozeduren unterschieden. Die Wahl der Prozedur ist davon abhängig, ob der MRTD Chip bzw. das Lesegerät konform zur Spezifikation sind. Sind beide konform, so wird die erweiterte Prozedur durchgeführt. Sobald ein Teil nicht konform ist wird nur die Standardprozedur ausgeführt. Die Auswahl der richtigen Prozedur wird in der nächsten Tabelle laut dem Bundesamt für Sicherheit in der Informationstechnik (2007) wie folgt verdeutlicht.

Inspektionssystem	MRTD Chip	
	konform	nicht konform
konform	erweitert	Standard
nicht konform	Standard	Standard

Tabelle1: Inspektionsprozedur

4.4.1 Standard ePass Inspektionsprozedur

Die Standardprozedur besteht aus folgenden Schritten:

1. Auswahl der ePass Anwendung (erforderlich)
   Vom MRTD Chip wird abhängig von der Verwendung von BAC, Zugriff auf die Daten erlaubt.
   - mit BAC: es soll, mit Ausnahme von generellen Systemdaten, kein Zugriff ermöglicht werden
   - ohne BAC: Lesezugriff auf wenig sensitive Daten soll möglich sein
2. Basis Access Control (unverbindlich)
   Falls erfolgreich abgewickelt, führt der Chip folgendes aus:
   - starten einer sicheren Übertragung
   - Zugriff auf wenig sensitive Daten zulassen
   - beschränkte Zugriffsrechte, um sichere Übertragung zu sichern
3. Passive Authentication (erforderlich)
   Vom Lesegerät müssen die Sicherheitsobjekte ausgelesen und verifiziert werden.
4. Active Authentication (optional)
   Das Lesegerät sollte, falls vorhanden, die Daten aus dem Feld DG15 auslesen und verifizieren und die Active Authentication durchführen.
5. Daten lesen und authentifizieren
   Das Lesegerät kann die wenig sensitiven Daten auslesen und verifizieren.

4.4.2 Erweiterte ePass Inspektionsprozedur

Die erweiterte Prozedur besteht aus folgenden Schritten:

1. Auswahl der ePass Anwendung (erforderlich)
   Der MRTD Chip soll keinen Zugriff auf die Daten zulassen. Ausnahme sind wieder die generellen Systemdaten.

2. Basic Access Control (erforderlich)
   Falls erfolgreich abgewickelt, führt der Chip folgendes aus:
   - starten einer sicheren Übertragung
   - Zugriff auf wenig sensitive Daten zulassen
   - eingeschränkte Zugriffsrechte um sichere Übertragung zu verlangen

3. Chip Authentication (erforderlich)
   Das Lesegerät soll die Daten des Feldes DG14 auslesen und die Chip Authentication durchführen.
   In diesem Schritt führt der Chip folgendes aus:
   - Neustart der sicheren Übertragung
   - beschränkte Zugriffsrechte, um neue sichere Übertragung zu sichern

4. Passive Authentication (erforderlich)
   In diesem Schritt muss das Lesegerät folgendes ausführen:
   - auslesen und verifizieren der Sicherheitsobjekte
   - Daten des Feldes DG14 verifizieren

5. Active Authentication (optional)
   Das Lesegerät sollte, falls vorhanden, die Daten aus dem Feld DG15 auslesen und verifizieren und die Active Authentication durchführen.

6. Terminal Authentication (unverbindlich)
   Falls ein Zugriff auf die sensiblen Daten ermöglicht werden soll, muss dieser Schritt durchgeführt werden.
   Falls erfolgreich abgewickelt muss der MRTD Chip folgende Schritte abarbeiten:

- Zugriff auf alle Datengruppen zulassen, für die das Lesegerät die entsprechenden Rechte besitzt
- beschränkte Zugriffsrechte, um die von der Chip Authentication gestartete sichere Übertragung, mit Verwendung des kurzlebigen öffentlichen Schlüssels, zu sichern

7. Daten lesen und authentifizieren

   Das Lesegerät liest je nach Zugriffsrechten die entsprechenden Datenfelder aus und verifiziert diese.

4.5 Ablauf der Extended Access Control

In der folgenden Abbildung wird der prinzipielle Ablauf der EAC vereinfacht dargestellt.

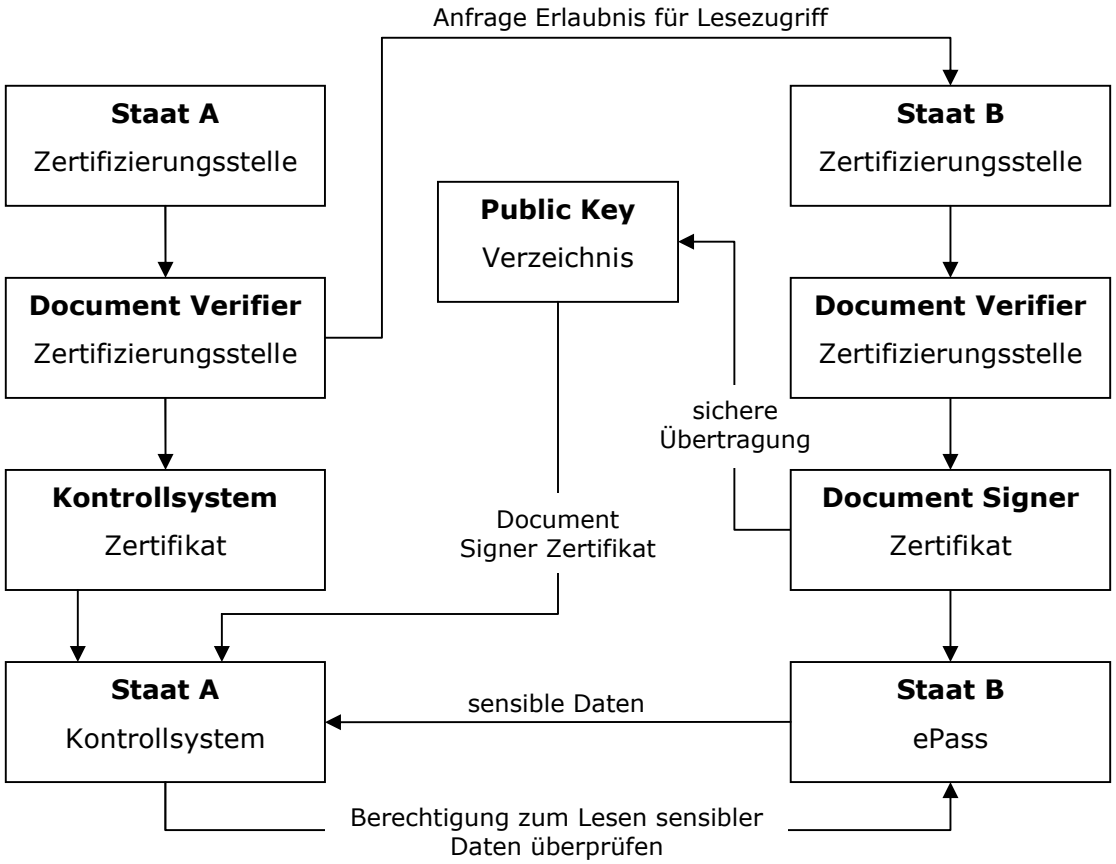


Abbildung 16: Ablauf der EAC

Im vorher dargestellten Szenario möchte eine Person aus dem Staat B in den Staat A einreisen. Dabei wird der Reisepass der Person von einem Lesegerät des Staates A überprüft.
Die grundlegende Arbeit passiert aber nicht erst bei der Passkontrolle sondern schon vorher. Es muss nämlich der Staat A beim Staat B die Erlaubnis für Lesezugriffe anfordern. Nur wenn der Staat A möchte, dass Lesegeräte des Staates B Zugriff auf sensible Daten haben sollen wird ein entsprechendes Zertifikat für die Lesegeräte ausgestellt.
Im ersten Schritt der Passkontrolle wird vom Lesegerät die Echtheit des Passes überprüft. Dieses Verfahren wird Chip Authentication genannt. Dabei wird über das Prinzip des Diffie-Hellmann Systems auf beiden Seiten der selbe private Schlüssel berechnet, ohne dass dieser übertragen werden muss. Der daraus entstandene private Schlüssel erhöht die Sicherheit enorm und schützt so gegen man-in-the-middle Angriffe. Alle weiteren Schritte werden verschlüsselt ausgeführt. Gegenüber der Basic Access Control bietet das neue System einen wesentlich höheren Grad an Sicherheit.
Bevor nun das Lesegerät Zugriff auf die Fingerabdrücke bekommt, muss es sich gegenüber dem Reisepass als berechtigt ausweisen. Dieser Schritt wird als Terminal Authentication bezeichnet. Dabei verwendet das Lesegerät eine Zertifikatkette und den bei der Chip Authentication berechneten privaten Schlüssel. Die Zertifiaktskette muss mit dem öffentlich Schlüssel der nationalen Wurzelinstanz enden, der sich auf dem Chip befindet. Erst wenn dieser Schritt erfolgreich absolviert wurde, lässt der Chip einen Zugriff auf die biometrischen Daten zu.

4.6 Public Key Infrastruktur

Alle beim maschinenlesbaren Reisepass verwendeten Signaturen und Zertifikate beruhen auf dem Grundprinzip einer Public Key Infrastruktur (PKI). Diese Grundform hat neben den sicheren Aspekten auch den, dass es über die Grenzen hinaus funktioniert und stellt damit eine Vertrauensstelle dar. Dadurch ergibt sich eine optimale Lösung aller technischen Aspekte

und zusätzlich wird die Sicherheit der, mit biometrischen Merkmalen ausgestatteten, Reisepässe über Staatsgrenzen hinaus gesichert.
Allgemein kann man drei große Möglichkeiten nennen, die eine Public Key Infrastruktur bietet:

- Bereitstellung öffentlicher Schlüssel: Benutzer können im System nach den öffentlich Schlüsseln anderer Benutzer suchen. Diese öffentlichen Schlüssel können auch auf deren Gültigkeit hin überprüft werden.
- Verwendung von Schlüssel: Zusätzlich zur einfachen Ansammlung verschiedener Schlüssel können diese auch gleich verwendet werden. Zumeist werden die Schlüssel für eine Verschlüsselung verwendet, da man für eine digitale Signatur den privaten Schlüssel benötigt und der sich nicht in der PKI gespeichert ist.
- Verwaltung der Schlüssel: Jegliche Tätigkeit, die einen Schlüssel betreffen, können in der PKI durchgeführt werden. Dazu zählen Vorgänge wie Schlüssel betrachten, anfordern oder zurückziehen. Auch der Vertrauensgrad von Schlüsselausstellern kann überprüft und festgelegt werden.

Damit ein Terminal die Terminal-Authentisierung durchführen kann, muss es mit zwei Dingen ausgestattet werden:

- Schlüsselpaar (geheimer und öffentlicher Schlüssel)
- Zertifikatskette (vom Chip verifizierbar)

In der Zertifikatskette sind die, dem Lesegerät erlaubten, Zugriffsrechte exakt definiert. Die Zugriffsrechte werden jeweils vom Land bestimmt, das den Reisepass ausstellt. Dadurch kann jedes Land festlegen, welche ausländischen Lesegeräte auf welche Daten zugreifen können.
In jedem Land gibt es eine oder mehrere Stellen, die solche Zertifikate ausstellen. Diese Stellen werden als Document Verifier (DV) bezeichnet. Jeder dieser DVs ist für eine bestimmte Reihe von Lesegeräten verantwortlich. Es ist zum Beispiel ein DV für alle Lesegeräte verantwortlich, die bei Grenzkontrollen eingesetzt werden.

Damit bei diesem System nicht der Überblick über alle Zertifikate verloren geht, ist auch eine so genannte nationale Wurzelinstanz vorhanden. Diese wird als Country Verifying Certification Authority (CVCA) bezeichnet und ist für die Ausgabe der DV Zertifikate verantwortlich. Als weitere wichtige Aufgabe der CVCA ist auch die Bereitstellung, des öffentlich Schlüssels, der auf allen Chips gespeichert wird. Dieser Schlüssel wird bei einer Kontrolle von dem Lesegerät ausgelesen und falls es die Zugriffsberechtigung besitz, kann es sich gegenüber des Chips mittels eines privaten Schlüssel und der Zertifikate authentisieren. Der öffentliche Schlüssel der CVCA muss dabei der letzte Teil der gespeicherten Zertifikatskette sein. Die folgende Abbildung stellt die Hierarchie der Zertifikatsverteilung dar.

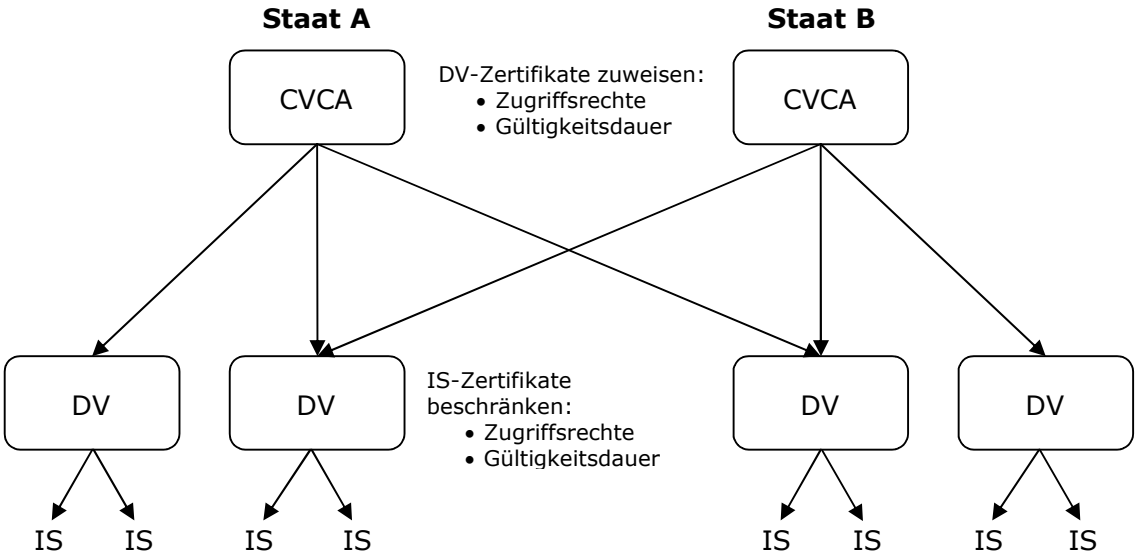


Abbildung 17: Zertifizierungshierarchie

Erklärung zur Grafik:

CVCA … Country Verifying Certification Authority
DV … Document Verifier
IS … Inspection Systems
⟶ … Zertifikatsvergabe

Aus dieser komplexen Vergabe von Zertifikaten ergibt sich die Möglichkeit der individuellen Vergabe von Zugriffsrechten. Soll nämlich ein

ausländisches Lesegerät Zugriff auf die biometrischen Daten erhalten, so muss vom CVCA dem entsprechenden DV ein passendes Zertifikat ausgestellt werden.

Zertifikatsverteilung für die Terminal Authentisierung:
Auf dem Chip des Reisepasses wird das CVCA Zertifikat des ausstellenden Landes (Staat B) gespeichert. Möchte nun ein Inspektionssystem von Staat A den Reisepass überzeugen, dass es die Erlaubnis besitzt, um auf die sensiblen Daten zuzugreifen, muss es über die korrekten Zertifikate verfügen. In unserem Fall wäre das ein DV Zertifikat des Staates A, das mit dem CVCA Zertifikat des Staates B signiert wurde. Nachdem der Reisepass die Zertifikatskette überprüft hat, ist es notwendig zu überprüfen, ob das Inspektionssystem Zugriff auf den, zum in der Kette befindlichen öffentlichen Schlüssel passenden, geheimen Schlüssel hat. Dies wird so realisiert, dass der Reisepass eine Nachricht an das Inspektionssystem schickt, die dieser mit dem geheimen Schlüssel verschlüsselt und zurück zum Reisepass schickt. Nun kann dieser mit Hilfe des öffentlichen Schlüssels überprüfen, ob das Lesegerät den passenden geheimen Schlüssel besitzt.

Abhören der Kommunikation
Es gibt zwei Möglichkeiten, wie Daten unberechtigt aus einem Reisepass ausgelesen werden könnten:

- aktiv: darunter versteht man das Auslesen der persönlichen Daten durch unauthorisierte Lesegeräte. Geschützt werden die Passdaten durch die Basic Access Control.
- passiv: damit ist das Abhören der Kommunikation zwischen dem Lesegerät und dem Pass gemeint. Dazu müsste ein möglicher Angreifer mit einer speziellen Abhöranlage eine oder mehrere Auslesevorgänge abhören. Aufgrund der starken Verschlüsselung der Daten müsste der Angreifer nachträglich mit extrem viel Rechenleistung die Verschlüsselung aufheben.

Rein theoretisch wäre ein Angriff durch passives Auslesen möglich, jedoch ist er in der Praxis eher unwahrscheinlich. Um jedoch die Passdaten absolut sicher zu schützen, stellen sich für die Sicherheit zwei relevante Fragen:

- In welchem Abstand kann die Kommunikation noch abgehört werden?
- Wie stark ist die Verschlüsselung der Kommunikation?

Wenn man der Sache der möglichen Mitleseentfernung auf den Grund geht stolpert man über viele Spekulationen, die aber meist nicht durch praktische Messreihen nachvollziehbar sind. Erst eine vom BSI durchgeführte Studie bringt verwertbare Ergebnisse über die Reichweite der verwendeten kontaktlosen Übertragung nach der ISO14443. In dieser Studie MARS wurden laut Dennis Kügler und Ingo Naumann (2006) neben der theoretischen Betrachtung des Problems auch unzählige Messreihen durchgeführt. Das Ergebnis dieser Studie ist, dass die Kommunikation bis etwa zwei Meter abhörbar ist. Ab etwa 2,7 Meter konnte kein erfolgreiches Abhören mehr durchgeführt werden.
Angenommen es wurde die Kommunikation abgehört, so muss im nächsten Schritt die Verschlüsselung aufgehoben werden. Zur Anwendung kommt das Triple-DES Verfahren im Cipher Block Chaining Modus (CBC). Dadurch ergibt sich bei Bitfehlern im abgehörten Chiffretext eine enorme Auswirkung im Klartext. Der verwendete symmetrische Sitzungsschlüssel ist 112 Bit lang, das entspricht der Sicherheit eines 2048 Bit RSA-Schlüssels.

Um einen nötigen Standard für die Sicherheit zu bieten, empfiehlt die ICAO folgende Schlüssellängen für verschiedene Verschlüsselungstechniken:

	RSA	DSA	Elliptic Curve DSA
Country Signer Certificat	3072 bit	256 bit	256 bit
Document Signer Certificate	2048 bit	224 bit	224 bit
Active Authentication Certificate	1034 bit	160 bit	160 bit

Tabelle 2: Vorschlag für Schlüssellängen

Bei diesen empfohlenen Schlüssellängen wurde nicht nur die Sicherheit mit dem Stand der Technik von heute, sondern auch die Weiterentwicklung in den nächsten Jahrzehnten einberechnet. Dies ist von Nöten, da der neue maschinenlesbare Reisepass eine Gültigkeit von 10 Jahren hat.

4.7 Protokoll Spezifikationen

In diesem Abschnitt werden kryptografische Protokolle für die Chip Authentication und die Terminal Authentication dargestellt, wie sie in der Technical Guideline TR-03110 (2007) spezifiziert wurden.
Bei der Ausführung der Protokolle sind zwei Parteien involviert:

- MRTD – Chip (im weiteren Verlauf als PICC bezeichnet)
- Inspektionssystem (wird als PCD benannt)

4.7.1 Schlüsselvereinbarung

Um einen universalen Überblick über die Schlüssel und Operationen beim Vereinbaren der Schlüssel zu erhalten, wird die Spezifikation auf einem verfahrensunabhängigen Weg erläutert. Mögliche Verfahren wären der Diffie-Hellman (DH) Schlüsselaustausch bzw. der auf Elliptischen Kurven basierende DH (ECDH).

Zur Verwendung kommen folgende Schlüsselpaare:

- Der MRTD – Chip hat ein statisches Diffie-Hellman Schlüsselpaar, beidem der öffentliche Schlüssel mit PK_{PICC} und der dazugehörige geheime Schlüssel mit SK_{PICC} bezeichnet wird. $\mathscr{D}_{PICC}$ bezeichnet die Domainparameter.
- Das Inspektionssystem generiert für jede neue Kommunikation ein kurzlebiges Diffie-Hellman Schlüsselpaar, beidem die Domainparameter der MRTD – Chips verwendet werden. Der öffentliche Schlüssel wird PK_{PCD} und der geheime SK_{PCD} genannt.

In der Technical Guideline TR-03110 (2007) wird empfohlen, dass der MRTD–Chip den öffentlichen Schlüssel des Inspektionssystems auf seine Gültigkeit hin überprüft. Die Berechnung des gemeinsam verwendeten geheimen Schlüssel K wird wie folgt definiert:

- MRTD – Chip: $KA(SK_{PICC},PK_{PCD},\mathscr{D}_{PICC})$
- Inspektionssystem: $KA(SK_{PCD},PK_{PICC},\mathscr{D}_{PICC})$

4.7.2 Signaturen

Auch hier wird die Spezifikation von einem systemunabhängigen Standpunkt betrachtet. Für die Signaturen werden beim Inspektionssystem der öffentliche Schlüssel mit PK_{PCD} und der dazugehörige private Schlüssel mit SK_{PCD} bezeichnet. Die Operationen um eine Nachricht zu signieren und zu verifizieren werden in der Technical Guideline TR-03110 (2007) wie folgt definiert:

- Eine Nachricht m mit dem geheimen Schlüssel SK_{PCD} signieren wird durch $s = Sign(SK_{PCD},m)$ definiert.
- Die Verifizierung der Signatur s mit dem öffentlichen Schlüssel PK_{PCD} wird durch $Verify(PK_{PCD},s,m)$ definiert.

4.7.3 Chip Authentication

Bei der Chip Authentication handelt es sich um ein kurzlebiges Diffie-Hellman Schlüsselvereinbarungsprotokoll, dass einerseits eine sichere Kommunikation und andererseits eine Gültigkeitsüberprüfung des Chips implementiert. Unter kurzlebig versteht man in diesem Zusammenhang, dass die Schlüssel eine sehr kurze Gültigkeitsdauer haben.
Das Protokoll wird in der Technical Guideline TR-03110 (2007) wie folgt spezifiziert:

MRTD Chip (PICC)		Inspection System (PCD)
static key pair: $(SK_{PICC}, PK_{PICC}, \mathcal{D}_{PICC})$		
	$\xrightarrow[\mathcal{D}_{PICC}]{PK_{PICC}}$	choose random ephemeral key pair $(\widetilde{SK_{PCD}}, \widetilde{PK_{PCD}}, \mathcal{D}_{PICC})$
	$\xleftarrow{\widetilde{PK_{PCD}}}$	
$K = \mathbf{KA}(SK_{PICC}, \widetilde{PK_{PCD}}, \mathcal{D}_{PICC})$		$K = \mathbf{KA}(\widetilde{SK_{PCD}}, PK_{PICC}, \mathcal{D}_{PICC})$

Abbildung 18: Chip Authentication

Bei der Abbildung 18 wird eine vereinfachte Version veranschaulicht, die wie folgt beschrieben wird:

1. Der MRTD – Chip sendet seinen öffentlichen Schlüssel PK_{PICC} und die Domain Parameter $\mathcal{D}_{PICC}$ zum Inspektionssystem.
2. Das Inspektionssystem berechnet sich aus den erhaltenen Daten das kurzlebige Diffie-Hellman Schlüsselpaar (SK_{PCD},PK_{PCD}, $\mathcal{D}_{PICC}$) und sendet den, nur kurz gültigen, öffentlichen Schlüssel PK_{PCD} zum MRTD – Chip.
3. Im folgenden Schritt müssen sowohl der Chip als auch das Inspektionssystem gleichzeitig folgende Berechnung ausführen:
   - Berechnung des gemeinsam verwendeten geheimen Schlüssel $K = KA(SK_{PICC},PK_{PCD},\mathcal{D}_{PICC})$
     $= KA(SK_{PCD},PK_{PCD},\mathcal{D}_{PICC})$
   - Für die sichere Nachrichtenübertragung werden K_{MAC} und K_{Enc} aus K berechnet.

- Zur Terminal Authentication wird der Hashwert H(PK_{PCD}) des öffentlichen Schlüssels des Inspektionssystems ermittelt.

In der Technical Guideline TR-03110 (2007) wird empfohlen, dass das Inspektionssystem direkt nach der Chip Authentication die Passive Authentication durchführt.
Betrachtet man den Sicherheitsstatus zu diesem Zeitpunkt, so gibt es zwei Möglichkeiten:

- Chip Authentication erfolgreich durchgeführt: Neustart der sicheren Nachrichtenübertragung mit den neu berechneten Sitzungsschlüssel K_{MAC} und K_{Enc}
- Chip Authentication nicht erfolgreich durchgeführt: verschlüsselte Übertragung der Daten wird mit den in der BAC berechneten Schlüssel weitergeführt

4.7.4 Terminal Authentication

Hierbei handelt es sich um ein Protokoll, dass ein „challenge–response" System die Authentizität des Inspektionssystems überprüft.
Durch folgende Grafik wird das Protokoll in der TR-03110 (2007) dargestellt:

MRTD Chip (PICC)		Inspection System (PCD)
choose r_{PICC} randomly	$\xrightarrow{r_{PICC}}$	
	$\xleftarrow{s_{PCD}}$	$s_{PCD} = \mathbf{Sign}(SK_{PCD}, ID_{PICC}\|\|r_{PICC}\|\|H(\widetilde{PK_{PCD}}))$
$\mathbf{Verify}(PK_{PCD}, s_{PCD}, ID_{PICC}\|\|r_{PICC}\|\|H(\widetilde{PK_{PCD}}))$		

Abbildung 19: Terminal Authentication

Bei der Terminal Authentication müssen folgende Schritte abgearbeitet werde:

1. Das Inspektionssystem sendet eine Zertifikatskette an den Chip. Dabei beginnt die Zertifikatskette mit einem öffentlichen Schlüssel einer CVCA, der am Chip gespeichert ist und endet mit dem IS Zertifikat des Inspektionssystems.
2. Der MRTD – Chip verifiziert die Zertifikate und liest den öffentlichen Schlüssel des Inspektionssystems (PK_{PCD}) aus. Danach schickt der Chip r_{PICC} an das Inspektionssystem.
3. Das Inspektionssystem reagiert mit der Antwort: $s_{PCD}=Sign(SK_{PCD},ID_{PICC}||r_{PICC}||H(PK_{PCD}))$
4. Im letzten Schritt überprüft der Chip die erhaltenen Daten: $Verify(PK_{PCD},s_{PCD},ID_{PICC}||r_{PICC}||H(PK_{PCD}))=true$

ID_{PICC} wird in diesem Protokoll die Dokumentnummer inklusive der Checkzahl genannt. Laut Spezifikation in der Technical Guideline TR-03110 (2007) muss jeglicher Datenaustausch verschlüsselt erfolgen und die Schlüssel aus der Chip Authentication verwendet werden.
Bei erfolgreicher Beendigung der Terminal Authentication wird dem Inspektionssystem Zugriff auf alle, den vorhandenen Zertifikaten entsprechende, Daten gewährleistet.

4.8 Funktionsbeispiele

In diesem Abschnitt werden Beispiele für die DG14 erläutert, die die Spezifikation erfüllen. Als erstes Beispiel wird eine DG14 auf ECDH (Elliptische Kurven Diffie-Hellman) Basis und als zweites funktionsfähiges ein auf Diffie-Hellman basierendes System. Die Beispiele stammen aus der Technical Guideline TR-03110 (Advanced Security Mechanisms for Machine ReadableTravel Documents – Extended Access Control (EAC), Version 1.1) (2007) der ICAO.

4.8.1 ECDH basierendes Beispiel

In der TR-03110 (2007) auf Seite 52 ist die DG14 hexadezimal dargestellt. Der für die Verschlüsselung benötigte geheime Schlüssel ist in Abbildung 21 hexadezimal angeführt.

```
0000 :  6E82014A 31820146 30820122 06090400 7F000702 02010230 82011330 81D40607
0020 :  2A8648CE 3D020130 81C80201 01302806 072A8648 CE3D0101 021D00D7 C134AA26
0040 :  4366862A 18302575 D1D787B0 9F075797 DA89F57E C8C0FF30 3C041C68 A5E62CA9
0060 :  CE6C1C29 9803A6C1 530B514E 182AD8B0 042A59CA D29F4304 1C2580F6 3CCFE441
0080 :  38870713 B1A92369 E33E2135 D266DBB3 72386C40 0B043904 0D9029AD 2C7E5CF4
00A0 :  340823B2 A87DC68C 9E4CE317 4C1E6EFD EE12C07D 58AA56F7 72C0726F 24C6B89E
00C0 :  4ECDAC24 354B9E99 CAA3F6D3 761402CD 021D00D7 C134AA26 4366862A 18302575
00E0 :  D0FB98D1 16BC4B6D DEBCA3A5 A7939F02 0101033A 0004680E C4FF3851 12D9A401
0100 :  76D36733 157B11FC 08B4A280 CE9B8246 4D765C38 C21CB883 6EE05724 3C1EBC7B
0120 :  B80EC484 41107C38 E4F545EB 213C300F 060A0400 7F000702 02030201 02010130
0140 :  0D060804 007F0007 02020202 0101.....................................
```

Abbildung 20: DG14 basierend auf ECDH

```
0000 :  12528622 D8947E85 E4988853 69ECDCAB F10E343A F7B95A99 DF610031
```

Abbildung 21: Privater Schlüssel für CA auf ECDH Basis

Im folgenden wird die Struktur und die Bedeutung der einzelnen Daten erklärt. In der Abbildung 22 werden dafür die Daten in folgende Spalten eingeteilt:

- Tag (Identifizierungsnummer)
- Length (Länge der Daten)
- Value (Daten)
- ASN.1 Type
- Comment (Kommentar zu Inhalt)

Tag	Length	Value	ASN.1 Type	Comment
6E	82 01 4A		-	Application specific tag “14”
31	82 01 46		SET	Set of SecurityInfos
30	82 01 22		SEQUENCE	SecurityInfo
06	09	04 00 7F 00 07 02 02 01 02	OBJECT IDENTIFIER	ChipAuthenticationPublicKeyInfo CA with ECDH
30	82 01 13		SEQUENCE	SubjectPublicKeyInfo
30	81 D4	siehe Abb. 23	SEQUENCE	AlgorithmIdentifier
03	3A	siehe Abb. 25	BIT STRING	SubjectPublicKey
–	–	–		optional keyId is unused
30	0F		SEQUENCE	SecurityInfo
06	0A	04 00 7F 00 07 02 02 03 02 01	OBJECT IDENTIFIER	ChipAuthenticationInfo CA with ECDH
02	01	01	INTEGER	Version 1
–	–	–		optional keyId is unused
30	0D		SEQUENCE	SecurityInfo
06	08	04 00 7F 00 07 02 02 02	OBJECT IDENTIFIER	TerminalAuthenticationInfo
02	01	01	INTEGER	Version 1
–	–	–		optional FileID is unused

Abbildung 22: Speicherstruktur der DG14 basierend auf ECDH

Zur Erklärung der oben angeführten Abbildung lässt sich anführen, dass der Tag die Art der Information angibt. Zum Beispiel weiß das Inspektionssystem, dass der Tag „6E“ den Beginn der DG14 bedeutet. Bei der Länge wird die Anzahl der Hexadezimalpärchen angegeben, dadurch ist für das Inspektionsgerät definiert, wie viele Daten zu diesem Tag gehören. Das Feld Value beinhaltet die in hexadezimal angegebenen Daten. Die Spalten ASN.1 Type und Comment sind nur zur Erklärung angeführt und werden so nicht am Chip gespeichert.

Damit die Übersicht nicht verloren geht, sind die Daten für AlgorithmIdentifier und SubjectPublicKey nicht in Abbildung 22 integriert. Diese beiden Datensätze beinhalten in sich nämlich wieder eine Struktur mit verschiedenen Daten.

Im AlgotihmIdentifier wird die Art des öffentlichen Schlüssels definiert und zusätzlich beinhaltet der Datensatz auch die Domain Parameter. Eine genauere Aufschlüsselung der Daten zeigt die Abbildung 23.

Tag	Length	Value	ASN.1 Type	Comment
30	81 D4		SEQUENCE	AlgorithmIdentifier
06	07	2A 86 48 CE 3D 02 01	OBJECT IDENTIFIER	Elliptic Curve Public Key
30	81 C8		SEQUENCE	Domain parameter
02	01	01	INTEGER	Version
30	28		SEQUENCE	Underlying field
06	07	2A 86 48 CE 3D 01 01	OBJECT IDENTIFIER	Prime field
02	1D	00 D7 C1 ... C8 C0 FF	INTEGER	Prime p
30	3C		SEQUENCE	Curve equation
04	1C	68 A5 E6 ... D2 9F 43	OCTET STRING	Parameter a
04	1C	25 80 F6 ... 6C 40 0B	OCTET STRING	Parameter b
–	–	–		Optional seed is unused
04	39	siehe Abb. 24	OCTET STRING	Group generator G
02	1D	00 D7 C1 ... A7 93 9F	INTEGER	Group order n
02	01	01	INTEGER	Cofactor f

Abbildung 23: AlgorithmIdentifier basierend auf ECDH

Der Aufbau der Daten erfolgt wie bereits zuvor erläutert. In der Sequenz AlgorithmIdentifier werden alle Informationen über den ausgewählten Algorithmus gespeichert.

Die nächste Abbildung zeigt den Group Generator. Dabei handelt es sich um einen Punkt auf der elliptischen Kurve, der für die Verschlüsselung verwendet wird. Dazu werden die x- und y-Koordinaten des Punktes in hexadezimaler Form gespeichert.

Tag	Length	Value	ASN.1 Type	Comment
04	39		OCTET STRING	Encoded group generator
		04	–	Uncompressed point
		0D 90 29 ... 12 C0 7D	–	x-coordinate
		58 AA 56 ... 14 02 CD	–	y-coordinate

Abbildung 24: Group Generator auf ECDH Basis

Die Sequenz Encoded public key enthält die Koordinaten eines Punktes auf der elliptischen Kurve. Als Speichertyp kommt hier ein Bitstring zur Anwendung.

Tag	Length	Value	ASN.1 Type	Comment
03	3A		BIT STRING	Encoded public key
		00		Number of unused bits
		04	–	Uncompressed point
		68 0E C4 ... 46 4D 76	–	x-coordinate
		5C 38 C2 ... EB 21 3C	–	y-coordinate

Abbildung 25: Punkt der Elliptischen Kurve als öffentlicher Schlüssel

In der nächsten Abbildung werden folgende Schlüssel zusammengefasst:

- zufällig gewähltes, nur kurz gültiges Schlüsselpaar → öffentlicher und privater Schlüssel
- durch Schlüsselaustausch berechneter, geheimer gemeinsamer Schlüssel

Private Key	7756F0C5 D1AB06C0 03672668 2B720C2F B1D5F789 B58244A6 DC07E5A2
Public Key	69D489F6 8A99ABC8 7106B3E1 3A52C6AF 2C57CEE5 72755FE3 712C8AC3, 8A6A3E9F E0694482 31BDC1BE FC826035 67E72602 EBA5C3EE EEAC3F15
Shared Secret	A770F66A CC78ED59 0581CC82 033C79F3 3BECE0A2 0C280244 79A4E97C

Abbildung 26: Schlüssel für ECDH basiertes Beispiel

Durch die in der Spezifikation angeführten Rechenvorgänge erhält man folgende zwei Sitzungsschlüssel:

K_{Enc}	DF94ED65 8A5CCB35 5FFFE612 8BADB584
K_{MAC}	1297F052 5DD9DE75 917DCD90 848465C1

Abbildung 27: Sitzungsschlüssel für ECDH Beispiel

Wird ein ECDH Algorithmus verwendet, so speichert der MRTD Chip die x-Koordinate des öffentlichen Schlüssels des Inspektionssystems.

$H(\widetilde{PK_{PCD}})$	69D489F6 8A99ABC8 7106B3E1 3A52C6AF 2C57CEE5 72755FE3 712C8AC3

Abbildung 28: Speicherung der x-Koordinate auf dem MRTD Chip

4.8.2 DH basiertes Beispiel

In der TR-03110 (2007) auf Seite 52 ist die DG14 hexadezimal dargestellt. Der für die Verschlüsselung benötigte geheime Schlüssel ist in Abbildung 30 hexadezimal angeführt.

```
0000 :  6E8201DC 318201D8 308201B4 06090400 7F000702 02010130 8201A530 82011A06
0020 :  092A8648 86F70D01 03013082 010B0281 8100DCB5 54DF8C69 31E865C1 B588273D
0040 :  80A2D87A B539C5E4 A074B402 49FF655A 9AB83063 3B457C4C F885E31C D79F8114
0060 :  8C8A68D1 DBFC2F7B 70ED55C0 387C23A0 479A9572 E8A6714F 418A6BF9 B00EC5BC
0080 :  4DEF255A 9485058A 4271008B A694AA62 CC18385E F9D7B6E8 33A7088A C817AA1F
00A0 :  9B93A86B 983EAB73 C15884E7 33665659 CA7D0281 802E69FE 94D3C0A4 378C8A47
00C0 :  9D83091A ED419234 25C10300 8C6AB3F6 E83E20CB 16C4AE0B 0E28ED9B C79CD7D7
00E0 :  E9DFD39D D0A39141 F2DD5714 9AB688DB AD177C68 6F771828 E5A04408 512F1564
0100 :  74B0BFD4 30CBBF91 C01589E7 21DDDFFC DF450043 EB771E61 084C597F 7AEA9048
0120 :  420A2180 EBFEC1B3 B93C1A6C B1AD38B3 984FF052 10020203 F9038184 00028180
0140 :  553CE735 ECF5CBF2 029D30FA A4F97335 DF404047 E4F8586D 76A7D221 A09E7F55
0160 :  BBE255C6 587BF288 5D41B786 BCEF2177 D52BF3CD BA785D37 D70B88D6 AB4E1CA6
0180 :  6A63B601 1376ED44 444A662B D0DC9524 176E9712 87AD41D2 9BED3D35 EAC7D39C
01A0 :  A73ECB2A 3B4D3967 1CE4125C 92658C5B F3DEDA91 5ED71B88 FC031BAB 887248A1
01C0 :  300F060A 04007F00 07020203 01010201 01300D06 0804007F 00070202 02020101
```

Abbildung 29: DG14 auf DH basierend

```
0000 :  01CD4A70 FFDC3D42 C862FD5E 3D781EB6 DE97677B 4FF61319 242E1499 B5CD1908
0020 :  A9B54221 135D1EDB AD787A5C E37586CF E86A61C4 78187157 267C97B4 0A7F2727
0040 :  B9B92FAC EC267CC0 1C883FA3 783BA07D C090EE04 99C9CE88 C684C874 0FEB84F4
0060 :  49B0F544 C1747716 46BDB7A3 0B0E3AB5 6C655F0E 83A98A4B A99EB9F1 0B0C0FBF
```

Abbildung 30: Privater Schlüssel für CA auf DH Basis

Im folgenden wird die Struktur und die Bedeutung der einzelnen Daten erklärt. Dazu werden die Daten aus Abbildung 29 zerlegt und ihre Bedeutung in der folgenden Abbildung in der Spalte Comment beschrieben. Die Felder AlgorithmIdentifier (siehe Abbildung 32) und SujectPublicKey (siehe Abbildung 33) werden nach der allgemeinen Speicherstruktur noch einzeln dargestellt.

Tag	Length	Value	ASN.1 Type	Comment
6E	82 01 DC		-	Application specific tag "14"
31	82 01 D8		SET	Set of SecurityInfos
30	82 01 B4		SEQUENCE	SecurityInfo
06	09	04 00 7F 00 07 02 02 01 01	OBJECT IDENTIFIER	ChipAuthenticationPublicKeyInfo CA with DH
30	82 01 A5		SEQUENCE	SubjectPublicKeyInfo
30	82 01 1A	siehe Abb. 32	SEQUENCE	AlgorithmIdentifier
03	81 84	siehe Abb. 33	BIT STRING	SubjectPublicKey
–	–	–	–	optional keyId unused
30	0F		SEQUENCE	SecurityInfo
06	0A	04 00 7F 00 07 02 02 03 01 01	OBJECT IDENTIFIER	ChipAuthenticationInfo CA with DH
02	01	01	INTEGER	Version 1
–	–	–		optional keyId unused
30	0D		SEQUENCE	SecurityInfo
06	08	04 00 7F 00 07 02 02 02	OBJECT IDENTIFIER	TerminalAuthenticationInfo
02	01	01	INTEGER	Version 1
–	–	–		optional FileID is unused

Abbildung 31: Speicherstruktur der DG14 basierend auf DH

Tag	Length	Value	ASN.1 Type	Comment
30	82 01 1A		SEQUENCE	AlgorithmIdentifier
06	09	2A 86 48 86 F7 0D 01 03 01	OBJECT IDENTIFIER	PKCS#3 dhKeyAgreement
30	82 01 0B		SEQUENCE	Domain parameter
02	81 81	00 DC B5 ... 59 CA D7	INTEGER	Prime p
02	81 80	2E 69 FE ... F0 52 10	INTEGER	Group generator g
02	02	03 F9	INTEGER	Private key length

Abbildung 32: AlgorithmIdentifier basierend auf DH

Tag	Length	Value	ASN.1 Type	Comment
03	81 84		BIT STRING	Encoded public key
		00		Number of unused bits
(02)	(81 80)	(55 3C E7 ... 72 48 A1)	(INTEGER)	Public key

Abbildung 33: Public Key auf DH Basis

In der nächsten Abbildung werden folgende Schlüssel zusammengefasst:

- Public und Private Key: zufällig gewählt, kurze Gültigkeitsdauer
- durch Schlüsselaustausch berechneter, geheimer gemeinsamer Schlüssel

Private Key	0170A377 AA4B612B 69A6762E CD71A91C 3D7CD149 A870F37 F357A196F F1134BF7 E0B33DDC EC645560 54EA9959 23189BDB 3893656F E05F8DA BE67F8998 3799E16F 9BF7A9CA 8050C949 31BAB4D8 CAA5F84B 33D71ACA 77A817C BC44CA92C 4B8960A2 034FBC31 999E7DEE 025E1001 EAF96113 BD06EFED FBBD5F2 E916ADC73 1971F019
Public Key	8EBC4457 EABBEF83 65D6EF83 9A1A3672 449486B2 779EF88E B0198ADD C64A096B 0AFC3C26 4D64EDFD F543C03B AEC7ED5D 58C2F2A1 4B63BB5D 280E62FE CAC0A6DD F255CEB9 2AB51C0D B672A251 68934F86 7B95552A 189A3244 4AF890B7 7509ED92 EC5A81A7 D787F5F5 51B37EB2 FA3A49C7 787B5F61 3527649C 151C1786 8417E9CA
Shared Secret	C30AAE5F DC23EFF6 E477734A C318D32D F128AF25 2542087F 1FA239DD 3734DE5C C7E154ED 93BDEC78 E87CD691 6307976F B2603425 133A61D4 10F7F050 EA0797B3 59A5009F 20C9BF0D 227C8866 B2C701FB 04ADF646 B138B1D6 D2623C17 AB3A910A 5A2C72E6 2B554B3F B5B2C310 A1F3334E D4C6AA77 919EA912 5A147C64 9C9E6556

Abbildung 34: Gewählte Schlüssel bei DH

Die, durch die Spezifikation berechneten, Schlüssel ergeben sich wie folgt:

K_{Enc}	EFF63AC6 29184F19 99C69B7C 3BFA4F17
K_{MAC}	7AD463F3 6997CB2B CB3D1B88 2CE8E4A7

Abbildung 35: Sitzungsschlüssel DH Beispiel

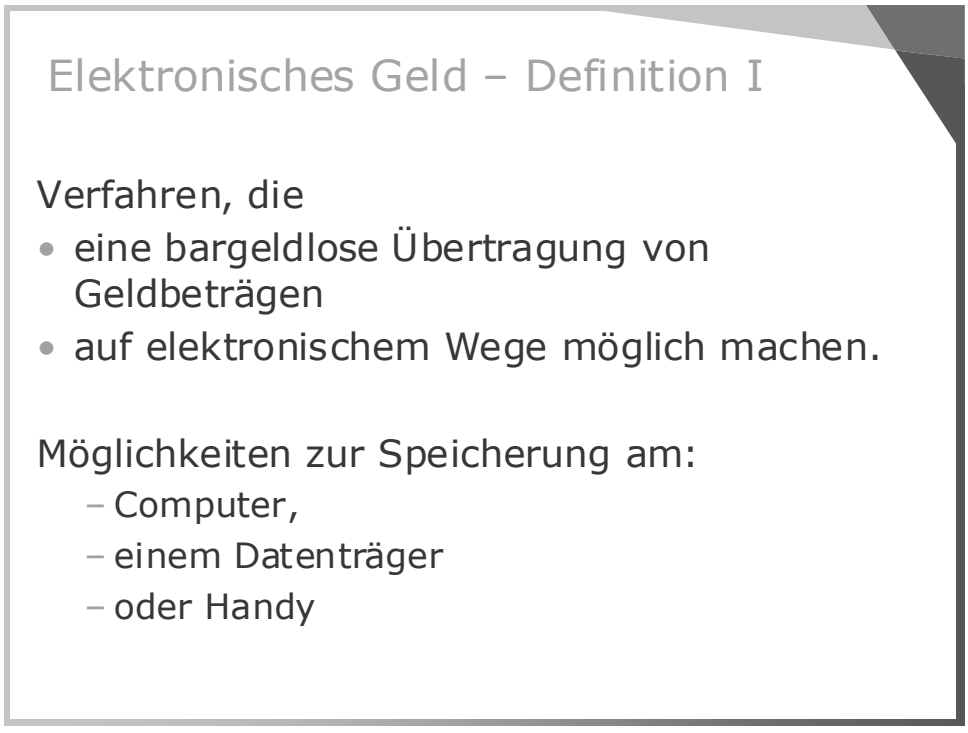
Für den Diffie-Hellman Schlüsselaustausch speichert der MRTD – Chip den SHA-1 Hash des öffentlichen Schlüssels, den er vom Inspektionssystem erhalten hat.

$H(\widetilde{PK_{PCD}})$	97D9AC36 0DCA6BB0 F2699B85 2DE37793 C29458CD

Abbildung 36: Speicherung des SCHA-1 Hash auf dem Chip

Anhang

1. Foliensatz Elektronisches Geld



Elektronisches Geld – Definition II

Für Europa geltende offizielle Definition für elektronisches Geld lautet wie folgt:

"ein monetärer Wert in Form einer Forderung gegen die ausgebende Stelle, der

- *auf einem Datenträger gespeichert ist,*
- *gegen Entgegennahme eines Geldbetrags ausgegeben wird, dessen Wert nicht geringer ist als der ausgegebene monetäre Wert,*
- *von anderen Unternehmen als der ausgebenden Stelle als Zahlungsmittel akzeptiert wird.„*

Quelle: E-Geld-Richtlinie 2000/46/EG

Eigenschaften elektronischer Zahlungssysteme

- Datenübertragung
- Höhe der Transaktionskosten
- Höhe des Transaktionsvolumens
- Rückverfolgbarkeit
- Überprüfbarkeit
- Akzeptanz
- Übertragbarkeit
- Teilbarkeit

Eigenschaft - Datenübertragung

- zwischen Banken – SWIFT Netzwerk
- kryptografische Verschlüsselung
  - Schutz der Daten
  - Authentizität und Unversehrtheit
- Kontrolle der Zugriffsberechtigung:
  - Passwort, PIN, Smartcard, Fingerabdruck…

Eigenschaft - Transaktionskosten

- Niedrig: Kosten im Centbereich, Minimierung der Ausgaben für Hard- und Software sowie Kommunikation
  - Beispiel: Telefonkarte
- Mittel: Kosten ca. 50 Cent, hohe Fixkosten durch automatisierte Transaktionen
  - Beispiel: POS- Systeme (Point Of Sale)
- Hoch: Kosten ca. 1-2 Euro, Systeme mit zum Teil manueller Abwicklung
  - Beispiel: Kreditkarten

Eigenschaft - Transaktionsvolumen

- Nanopayments (ca. 1 – 10 Cent)
- Micropayments (ca. 10 Cent – 10 Euro)
- Medium-Payments (ca. 10 – 10.000 Euro)
- Macropayments (ab ca. 10.000 Euro)

Eigenschaft - Rückverfolgbarkeit

- Uneingeschränkte Rückverfolgbarkeit
  - Beispiel: Bezahlung mit Kreditkarte
- Eingeschränkte Rückverfolgbarkeit
  - Beispiel: Chipkarten mit Wertspeicherung
- Nicht rückverfolgbare Rückverfolgbarkeit
  - Beispiel: Bezahlung mit Bargeld
- Benutzergesteuerte Rückverfolgbarkeit
  - Beispiel: verschlüsselte Quittung, die nur der Zahlende entschlüsseln und somit die Anonymität aufheben kann

Univ. Prof. Dr. Wolfgang Klas

Eigenschaft - Überprüfbarkeit

- Online: Überprüfung der Liquidität des Käufers während Bezahlvorgang → hoher Aufwand, dadurch hohe Transaktionskosten
  - Beispiel: kreditkartenbasierten Systemen
- Offline: fälschungssichere Hardware oder kryptografische Techniken, schnellere Abwicklung → geringere Kosten
  - Beispiel: Prepaid Systeme

Eigenschaft – Akzeptanz I

- maßgeblich entscheidend ob sich System durchsetzt
- laut Markus Stolpmann in „Elektronisches Geld im Internet" drei wichtige Faktoren
  - die Sicherheit des Systems
  - die entstehenden Kosten auf beiden Seiten
  - dem Verbreitungsgrad

Eigenschaft – Akzeptanz II

- Definition nach Univ. Prof. Dr. Wolfgang Klas:

  „Ein Zahlungssystem besitzt Akzeptanzfähigkeit, falls es überall angenommen wird, d.h. elektronische Beträge, die eine bestimmte Bank herausgibt, werden auch von anderen Banken angenommen."

Eigenschaft - Übertragbarkeit

- Weitergabe von Guthaben ohne eine Bank zu benötigen
- mit heutigen Mitteln schwer zu realisieren
- Hauptproblem: Kopien von Weitergaben könnten von verschiedenen Personen eingelöst werden

Eigenschaft - Teilbarkeit

- Guthaben kann in beliebige Teile zerlegt und weiterverwendet werden
  - Beispiel: kontobasierte Zahlungssysteme
- Definition nach Bernd Rothhaas

  „In einem elektronischen Zahlungssystem sollte ein Wert x in eine beliebige Anzahl von Beträgen jedweder Höhe unterteilt werden können. (Der Gesamtwert der Beträge muss natürlich dann wieder x entsprechen.)"

Eigenschaften - Zusammenfassung

- sehr viele Forderungen an elektronische Zahlungsmittel
- Auswahl des Systems durch Anforderungen der Aufgabe
- kein universelles System

Grundkonzepte elektronischer Zahlungssysteme

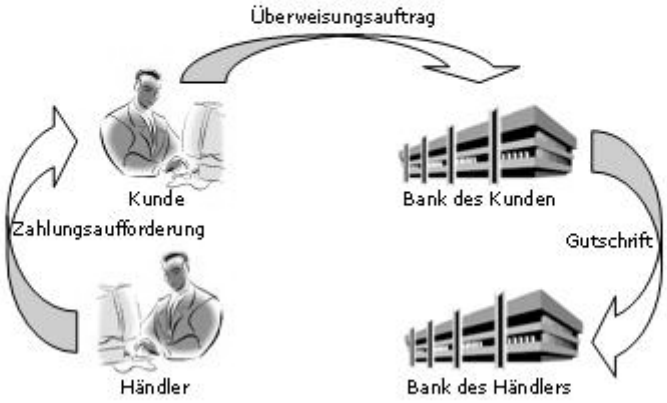
- Überweisungssystem
- Schecksystem
- Token System
  - Einweg Token System
  - Mehrweg Token System

Überweisungssystem

- kontobasiertes System
  - Kunde und Händler besitzen Konto
- Zahlungsaufforderung und Überweisungsauftrag werden elektronisch übertragen

Überweisungssystem - Funktionsweise

Schematische Darstellung:



Überweisungssystem - Sicherheit

- drei wichtige Faktoren
  - Wirksamkeit der Zugangskontrolle und der Identifikationsmechanismen bei der Autorisierung
  - Verlässlichkeit des Datentransfers
  - Infrastruktur der Übertragungskanäle
- bei Einhaltung → wirksamer Schutzmechanismus

Überweisungssystem - Eigenschaften

- hohe Transaktionskosten durch
  - großen technischen Aufwand
  - Bearbeitungszeiten
- uneingeschränkte Rückverfolgbarkeit
  - Bank kennt alle Daten der Überweisung (Quelle, Ziel, Betrag, Datum, Uhrzeit)

Schecksystem

- kontobasiertes System
- Schecknehmer kann Scheck bei Bank gegen Geld einlösen
- Scheck ist ein Schreiben, die eine Person befugt, bei der Bank eine gewisse Summe vom Guthaben einer anderen Person einzufordern

Schecksystem - Funktionsweise

Schematische Darstellung:

Schecksystem - Eigenschaften

- Sicherheit abhängig von
  - gewählten Verschlüsselungsmechanismen
  - integrierter digitale Signatur
- hohe Transaktionskosten hervorgerufen durch
  - aufwendiges Scheckbearbeitungsverfahren
- Datenschutz problematisch da
  - Aussteller zu jedem Zeitpunkt nachvollziehbar

Token System

- Token = elektronische Datenpakete, die realem Geld entsprechen
- Unterscheidung in
  - Einweg Token Systeme
  - Mehrweg Token Systeme
- Probleme:
  - dürfen weder fälschbar noch
  - manipulierbar sein und
  - einen hohen Grad an Anonymität realisieren

Token System - Problemlösungen

- Echtheit durch digitale Signatur der Bank sichergestellt
- mehrmalige Verwendung („double spending“) wird mittels two-part-lock Verfahren verhindert
- Sicherstellung der Anonymität durch das Verfahren der blinden Signatur

Problem „double spending“

- unerlaubte Vervielfältigung von Token
- 2 mögliche Täter:
  - Kunde
  - Händler

  → für die Bank relevant, um Strafverfahren gegen die richtige Person einleiten zu können
- Lösung: two-part-lock Vefahren

Problemlösung „double spending“ two – part – lock Verfahren I

- Kundendaten symmetrisch verschlüsselt
- verwendete Schlüssel mittels two-part-lock Verfahren verschlüsselt
- Token einlösen:

  → halbe Kundendaten entschlüsselt

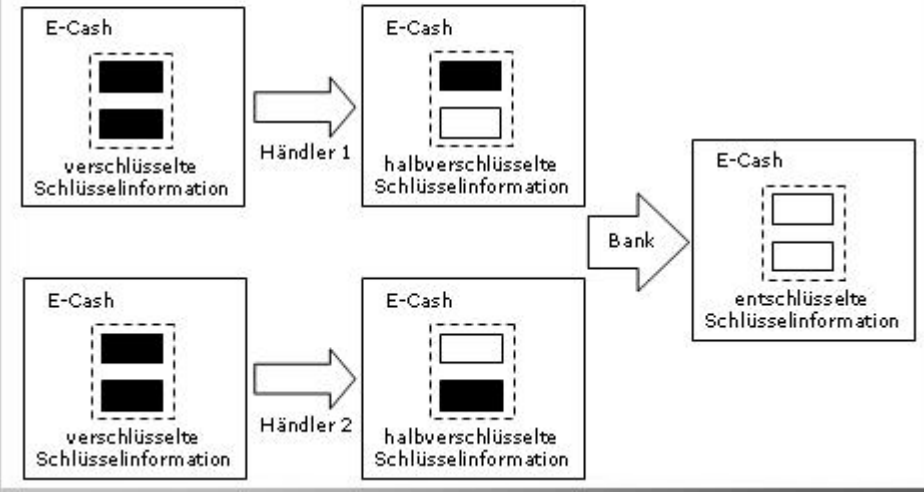
  → Überprüfung der Seriennummer in Datenbank

  → bei Treffer

  → Daten lesbar

Problemlösung „double spending“ two – part – lock Verfahren II

Schematische Darstellung:



Problemlösung „double spending“ two – part – lock Verfahren III

Funktionsweise:

a) Zerlegung der Kunden- und Schlüsseldaten in Byte Strings → N–mal kopieren (Sicherheit steigt mit N) → Stringpaare mit linker und rechter Seite → rekonstruieren mit einer Seite nicht möglich, nur durch Exklusiv-Oder Funktion auf beide Strings → verschlüsseln der einzelnen Stringpaare (2N Schlüssel nötig!)

Problemlösung „double spending“ two – part – lock Verfahren IV

b) Banknote mit verschlüsselten Strings an Begünstigten schicken → dieser fordert pro Stringpaar einen Schlüssel (Auswahl per Zufallsgenerator) → jeweils eine Seite entschlüsselt → Daten an Bank weiterleiten → durchsuchen der Datenbank → bei Treffer gibt es zwei Möglichkeiten:

1. idente Schlüsselverteilung: Händler wahrscheinlich der Kriminelle, da bei 2N Schlüssel die Wahrscheinlich gleicher Schlüssel äußerst gering ist
2. Nicht idente Schlüsselverteilung: Kunde ist der Schuldige, durch verschiedene Schlüssel wird mindestens ein Stringpaar vollkommen entschlüsselt und somit die Identität frei lesbar

Prinzip der blinden Signatur

- Signatur == Unterschrift
- blinde Signatur bedeutet:
  - Person A übergibt an Person B ein nicht lesbares Dokument
  - Person B unterschreibt dieses ohne den Inhalt zu kennen
  - Person B gibt das signierte Dokument zurück an Person A

Prinzip der blinden Signatur - Funktionsablauf

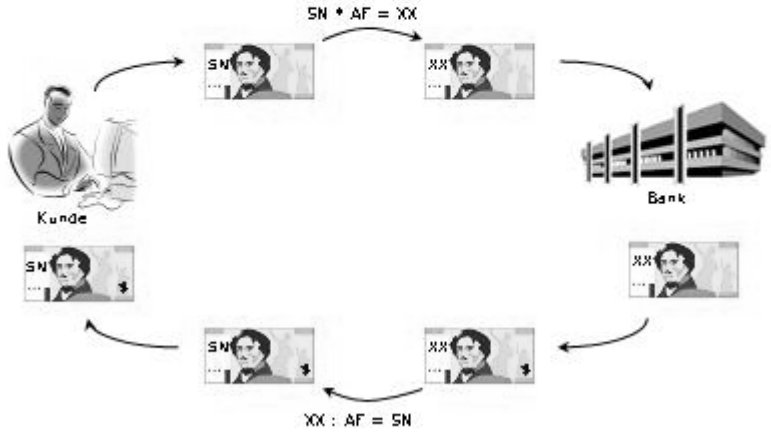
Kunde erstellt Banknote BN + Seriennummer → wählt zufälligen Ausblendfaktor AF und verschlüsselt damit die Seriennummer → protokolliert Tupel (BN,SN,AF) → schickt Banknote mit verschlüsselter SN an Bank

Bank bucht Betrag von Konto ab und unterzeichnet mit digitaler Signatur → Echtheit der Banknote garantiert, Anonymität des Kunden gewährleistet, da Bank SN nicht kennt → Bank protokolliert Tupel (SN,XX) und sendet BN zurück zum Kunden

Kunde blendet SN durch Division mit AF wieder ein → besitzt gültiges Zahlungsmittel

Prinzip der blinden Signatur - Funktionsschema

Schematische Darstellung:



Einweg Token System

Lebenszyklus:

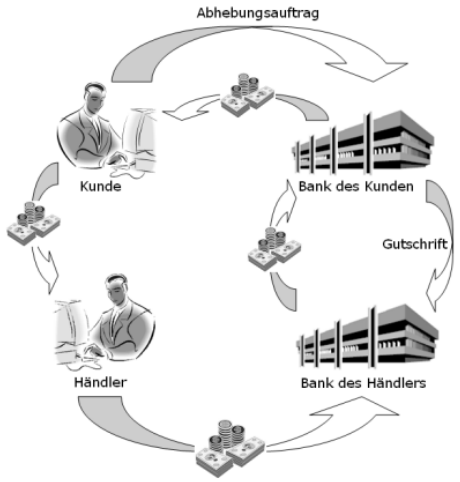
1. Generierung
2. Weitergabe
3. Umtausch in reales Geld

Ablauf:

- Person A schickt an seine Bank einen Abhebungsantrag
- Bank zieht den gewünschten Betrag vom Konto ab und schickt Token zurück
- Person A bezahlt bei einer Person B eine Rechnung mit diesen Token
- Person B bringt Token zu seiner Bank
- Bank schickt Token weiter an die Bank von Person A, diese sendet Guthaben zurück
- Guthaben wird auf Konto von Person B verbucht

Einweg Token System

Schematische Darstellung:



Elektronische Zahlungsmittel

- ECash
- SET – Secure Electronic Transfer
- Kreditkartenzahlung mittels
  - SET
  - SSL

ECash – Allgemeines I

- von Unternehmen DigiCash
- Gründer: David Chaum
- 3 Kriterien:
  - eine ausgebende Stelle
  - Anonymität des Benutzers
  - einmalige Verwendung
- besonders für Micropayment geeignet

ECash – Allgemeines II

- 1994 Pilotversuch mit 30000 Internetusern und einigen Banken und Onlinestores
- Chaum seiner Zeit voraus
- Durchbruch mangels Interesse nicht geschafft
- DigiCash Ende der 90er Jahre aufgelöst

ECash - Eigenschaften

- eigens entwickeltes Protokoll über TCP/IP
- Entwicklung der blinden Signatur durch Chaum für die Anonymität des Benutzers
- Erkennung von „double spending" plus Enttarnung des Täters
- Wiederherstellung bei Systemabstürzen durch integrierte
  - Backup Lösung bzw.
  - durch die Coin-ID
- RSA verschlüsselte Kommunikation
- Schlüssel der Bank in Software integriert
- Benutzer erhält bei Installation seine Schlüssel, per Zufallsgenerator erstellt

E-Cash – Blinde Signatur

- digitale Unterschrift auf elektronischem Weg
- kryptografische Lösung mittels RSA
- Funktionsablauf:
  - Person A wählt Zufallszahl r (teilerfremd zu n) und berechnet: $x = dokument \cdot r^e$
  - Zahl wird Person B zum Signieren vorgelegt → Signatur erstellen, indem sie die Zahl mit ihrem geheimen Schlüssel d potenziert: $y = x^d$
  - Dokument zurück zu Person A → aufheben der Verschlüsselung durch $y \cdot r^{-1} (= dokument^d)$

ECash - Anforderungen

- **Verifikation:** die Banknote muss für jeden Händler erkennbar sein und als Zahlungsmittel akzeptiert werden
- **Anonymität:** der Käufer soll nach dem Zahlungsvorgang nicht rückverfolgbar sein

E-Cash – Verfahrensbeschreibung I

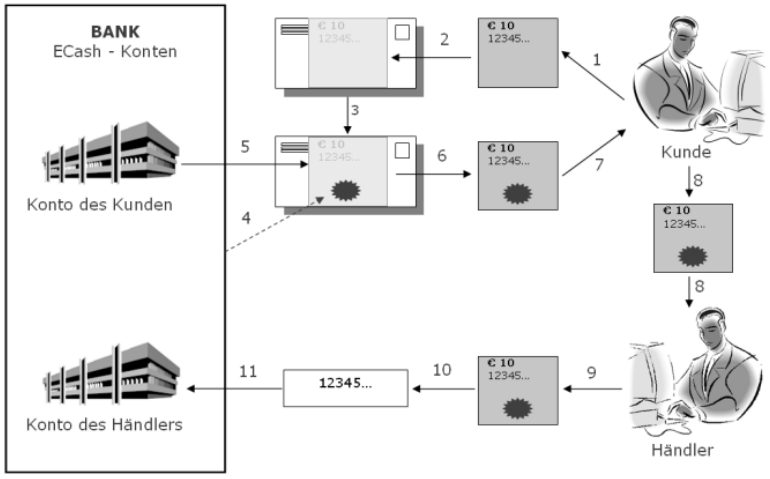
- drei agierende Parteien:
  1. Kunde: benötigt Konto und spezielle Software, elektronisches Geld als Datei auf Festplatte gespeichert
  2. Händler: benötigt Software, muss ECash akzeptieren, kann auch eine Privatperson sein
  3. Banken: tauschen reales in elektronisches Geld und umgekehrt, zuständig für Verwaltung der Konten, garantieren realen Gegenwert

E-Cash – Verfahrensbeschreibung II

Annahmen: Für die folgende Darstellung des Verfahrens von David Chaum wird angenommen, dass eine Person einen elektronischen Geldschein im Wert von 10 Euro erstellen und mit diesem bei einem Händler zahlen möchte. Die Bank der Person ist in diesem Fall befugt, dass sie elektronisches Geld erzeugt.

E-Cash – Verfahrensbeschreibung III

Schematische Darstellung:



E-Cash – Verfahrensbeschreibung IV

Beschreibung des Ablaufs:

- digitale Münze erzeugen: Wert und Seriennummer auf Zettel schreiben, Zettel gemeinsam mit Kohlepapier in Umschlag geben und der Bank übergeben
- Bank signiert geschlossenen Umschlag und belastet Konto der Person, die Seriennummer bleibt geheim, Bank gibt den Umschlag an die Person zurück
- Geldschein aus dem Umschlag nehmen → gültiges Zahlungsmittel → bezahlen beim Händler (dieser muss Zahlungsmittel akzeptieren)
- Händler löst elektronischen Geldschein bei Bank ein: Bank durchsucht Datenbank nach Seriennummer, falls noch nicht registriert → Gutschrift auf Konto und eintragen der Seriennummer in Datenbank

E-Cash – Verfahrensbeschreibung V

Digitale Umsetzung:

Person A möchte elektronische Münze anfordern → wählt dafür zwei Zufallszahlen C (dient zur Verschleierung) und V (für Erzeugung der Bitstrings)

bildet W, durch 2-maliges hintereinander schreiben von V:

$$W = VconcatV$$

verschlüsselt W mit öffentlichem Schlüssel der Bank:

$$S := C^e \cdot W \bmod n$$

Person A schickt S an Bank, diese bucht Betrag vom Konto ab und signiert S mit ihrem persönlichen Schlüssel:

$$T := S^d \bmod n$$

E-Cash – Verfahrensbeschreibung VI

Digitale Umsetzung Fortsetzung:

Bank schickt T an Person A zurück → diese überprüft Korrektheit der Daten durch anwenden des öffentlichen Schlüssel der Bank auf T

bei Korrektheit muss gelten:

$$T^e = S^{de} = S$$

Person A muss Verschleierungsfaktor entfernen um an die Münze zu gelangen:

$$F := T \cdot C^{-1} \bmod n$$

→ Person A ist nun im Besitz der signierten unverschlüsselten Münze

E-Cash – Verfahrensbeschreibung VII

Digitale Umsetzung: Erklärung der Abhängigkeiten

Folgende Umformung zeigt, dass F nicht von C, sondern von d und V abhängt:

$$F := T \cdot C^{-1} \bmod n = S^d \cdot C^{-1} \bmod n = C^{ed} \cdot W^d \cdot C^{-1} \bmod n = C \cdot W^d \cdot C^{-1} \bmod n = W^d \bmod n$$

Das stellt in der Praxis kein Problem dar, da niemand d und V kennt.

Die Echtheit kann von jeder Person überprüft werden:

$$F^e \bmod n = W^{de} \bmod n = W \bmod n$$

Bei identem Ergebnis auf beiden Seiten von (F^e *mod n*) ist F eine gültige Münze. Aufgrund der Mächtigkeit der Exponentialfunktion kann man aus F diese Information noch nicht gewinnen. D.h. es ist nicht möglich eine Banknote herzustellen, ohne den geheimen Schlüssel der Bank zu kennen

E-Cash – Verfahrensbeschreibung VIII

Digitale Umsetzung: Sicherstellung der Anonymität

Anonymität des Bezahlenden sichergestellt, da Bank nur

$$S = C^e \cdot W \bmod n$$

und zum anderen

$$T = C^{ed} \cdot W^d = C \cdot W^d \bmod n$$

kennt.

In beiden Zahlen steckt C, wodurch sie keine Möglichkeit besitzt die Geldmünze F zu berechnen.

ECash – Ausbau- und Verbesserungsstufen I

- Problem: Bank muss Antragsteller glauben, dass auf der Münze der tatsächliche Wert steht
- Lösung: Antragsteller schickt 100 gleiche Münzen an Bank, die davon 99 zufällige entschlüsselt
  → Wahrscheinlich sehr klein, dass ein Betrugsversuch nicht auffliegt
  → Restrisiko senken, indem bei erwischten Betrugsversuchen äußerst hohe Strafen vollzogen werden

ECash – Ausbau- und Verbesserungsstufen II

- Problem: erstellen und einlösen von Duplikaten
- Lösung: Identifikationsstring auf Geldmünze, Erfassung in Datenbank, Mehrfachverwendung aufgedeckt und Täter enttarnt
- Beschreibung: Identifikationsstring wird vom Bezahlenden auf die Geldmünze geschrieben, nachdem der Empfänger die Banksignatur überprüft hat. Bei positivem Treffer der Seriennummer in der Datenbank gibt es 2 Möglichkeiten:
  - Identifikationsstring gleich: Empfänger ist Übeltäter, da er Geldschein erst kopieren konnte, als der String schon vorhanden war
  - Identifikationsstring anders: Bezahlender ist Übeltäter
- Nachteil: sobald Strings ident wird Händler als Täter angenommen → Bezahlender könnte dies ausnutzen

ECash – Ausbau- und Verbesserungsstufen III

- Problem: erstellen und einlösen von Duplikaten
- Lösung: Secret Splitting
  - Nachricht verschlüsselt und in Stringpaare zerlegt
  - Nachricht erst lesbar, wenn man alle Teile besitzt
  - Hälfte der Stringpaare → keine Möglichkeit um an Daten zu kommen

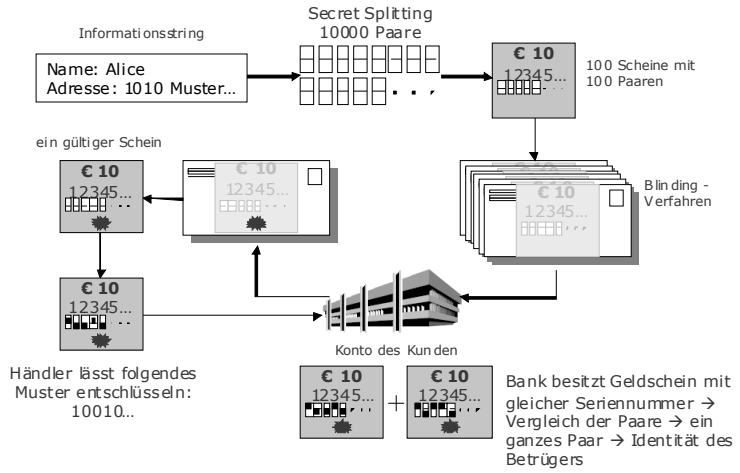
ECash – Secret Splitting I

- String so gewählt, dass er die Identität der Person enthält
- Beispielstring: „Meine Name ist Alice und wohne in der Musterstraße 1 in 1010 Wien"
- mittels Secret Splitting Protokoll wird x-Mal das Tupel (I1,I2) erzeugt
- pro Schein (x/Anzahl der Scheine) dieser Tupel
- Beispiel: x=10000, 100 Scheine → 100 Tupel auf jedem Schein

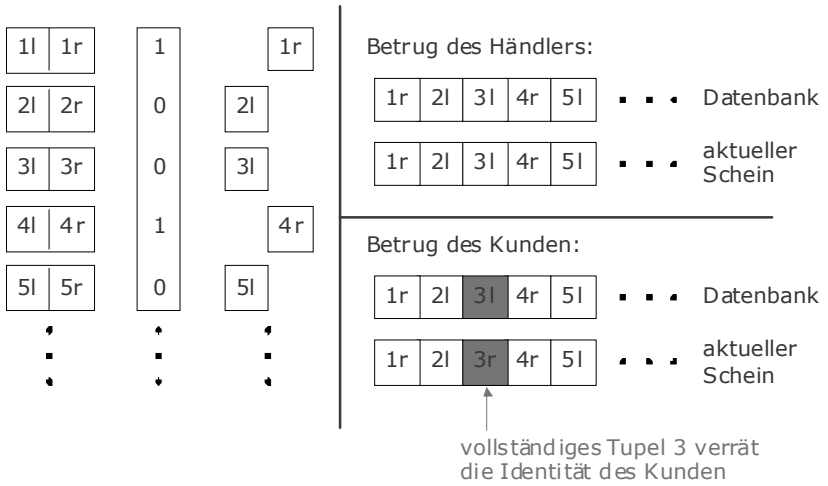
ECash – Secret Splitting II

- Geldscheine an Bank
- Bank entschlüsselt alle Scheine bis auf einen und lässt sich alle Tupel entschlüsseln
- alle Tupel ergeben idente Information → Bank signiert den letzten Schein
- Bank schickt Schein zurück
- Alice verwendet den Geldschein für eine Bezahlung
- der Händler überprüft Signatur der Bank → falls korrekt bittet er Alice um Entschlüsselung der halben Anzahl der Tupel (Auswahl per Zufallsgenerator)
- Händler löst Geldschein bei Bank ein
- Bank überprüft Seriennummer in Datenbank → falls vorhanden wird Annahme verweigert und durch Zusammenführung der beiden Scheine wird der Betrüger enttarnt

ECash – Secret Splitting III Funktionsschema



ECash – Secret Splitting IV

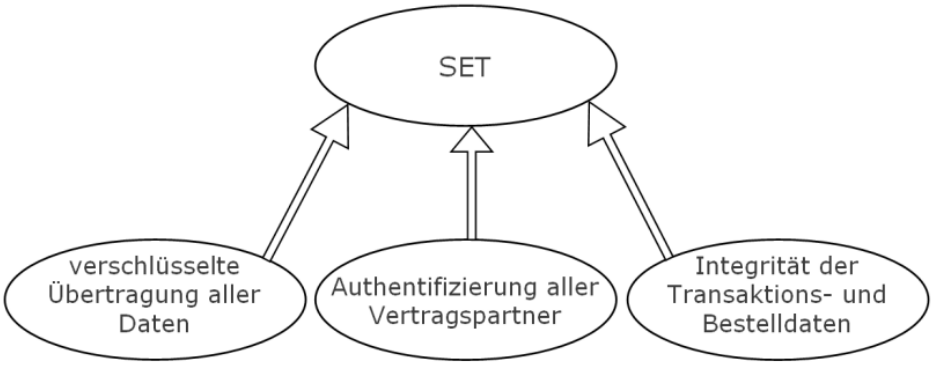


SET – Secure Electronic Transaction

- Protokoll für sicheren Zahlungsverkehr über das Internet
- Gemeinschaftsentwicklung (VISA, Mastercard, Microsoft, …)
- dadurch breite Unterstützung und
- Integration in häufigst verwendeten Browser gesichert
- Ziel:
  - Aufbau eines weltweiten Standards
  - Zuverlässige und sichere Abwicklung des Zahlungsverkehrs
  - Vergrößerung der Akzeptanz von Kreditkarten im Internet

SET – Secure Electronic Transaction

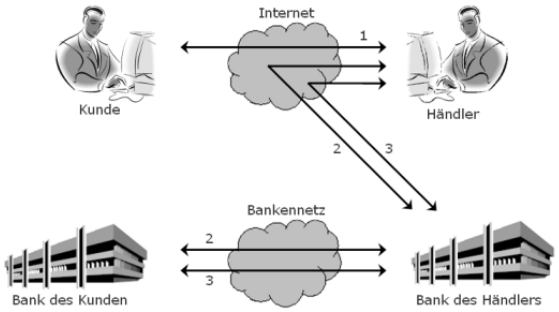
Zentrale Funktionen:



SET – Secure Electronic Transaction

Ablauf

1. Kaufanfrage
2. Zahlungsfähigkeitsüberprüfung
3. Zahlungsdurchführung



SET – Secure Electronic Transaction

1. Kaufanfrage
   - Kunde sendet Initialisierungsnachricht an Händler
   - Händler erstellt, mit Signatur-Zertifikat unterschriebene, Antwort und sendet diese mit dem Schlüssel seiner Kreditanstalt zurück an den Kunden
   - Kunde überprüft Zertifikate auf Gültigkeit
   - Kunde stattet Bestellung und Zahlungsinformationen mit der dualen Signatur aus (Zahlungsanweisung mittels DES verschlüsselt und dieser Schlüssel wird gemeinsam mit den Kreditkarteninformationen vom Kunden mitttels RSA verschlüsselt. Als RSA Schlüssel wird der des Finanzinstitutes gewählt)
   - Kunde übermittelt die Daten an den Händler, zusätzlich könnte ein Zertifikat des Kunden mitübertragen werden
   - Händler schickt nach dem Erhalt der Daten dem Kunden die Rechnung
   - Kunde überprüft die Zertifikate der Rechnung und speichert die Quittung ab

SET – Secure Electronic Transaction

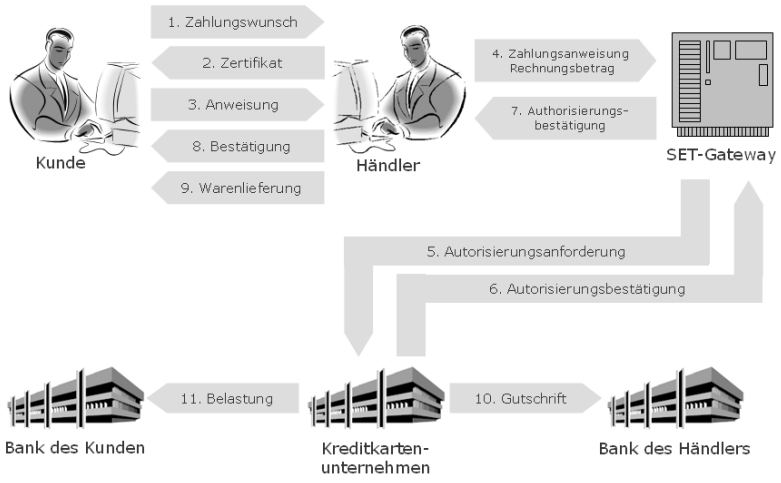
2. Zahlungsfähigkeitsüberprüfung
   - Händler hebt DES Verschlüsselung auf
   - Bestelldaten lesbar, Kreditkarteninformationen nicht
   - Händler schickt Zahlungsanweisung und zusätzlich eine Zahlungsfähigkeitsanfrage an das zuständige Finanzinstitut
   - Finanzinstitut überprüft alle Zertifikate und vergleicht die Zahlungsanweisung des Kunden mit den angegebenen Daten des Händlers
   - falls alle Tests ein positives Ergebnis haben, übermittelt das Finanzinstitut an die Kundenbank die Anfrage, ob diese Zahlung in Ordnung geht
   - Bank überprüft das Guthaben des Bezahlenden und schickt die Antwort zurück
   - Finanzinstitut erhält Antwort und schickt diese an den Händler zurück
   - Händler verschickt Ware

SET – Secure Electronic Transaction

3. Zahlungsdurchführung
   - Händler schickt Bestätigung an Kreditkartenunternehmen
   - Kreditkartenunternehmen kümmert sich um die weiteren Schritte
     - Abbuchung des Betrages vom Konto des Kunden
     - Gutschrift des Betrags minus den angefallenen Spesen auf dem Konto des Händlers
   - Abwicklung des Bezahlvorganges nicht über das Internet, sondern über eigenes Bankennetz realisiert

SET – Secure Electronic Transaction

Schematische Darstellung des Ablaufs:



SET – Secure Electronic Transaction

Problem: Aufwand für Händler und Kunde enorm

- Kunde benötigt Plug-In, welches elektronische Brieftasche anlegt indem sich auch Kundenzertifikat befindet (passwortgeschützt)
- Kunde muss Antrag bei Bank stellen um SET-Code zu erhalten, bevor er die Software in Betrieb nehmen kann
- Antrag wird von der Bank an Kreditkartengesellschaft weitergeleitet
- Kreditkartengesellschaft beauftragt Trust Center mit Generierung des Zertifikats
- Daten werden vom Trust Center über die Kreditkartengesellschaft zurück zur Bank gesendet
- Bank sendet Zertifikatsbrief an Kunden (beinhaltet Zertifikat und eventuell Software)
- erst jetzt ist die Software einsatzbereit
- Aktivierung des Zertifikats über Internetseite zwingend notwendig, damit man Aktivierungsmeldung (Wake-Up Message) erhält

SET – Secure Electronic Transaction

Probleme Fortsetzung:

- Händler benötigt ebenfalls SET-Software
- Händler muss von Firma „SET Secure Electronic Transaction LLC" (SETCo) zertifiziert sein
- Auslagerung bei kleineren Unternehmen sinnvoll
- Server des Unternehmens muss über Netzwerk mit SET-Gateway Server verbunden sein

→ hohe Transaktionskosten

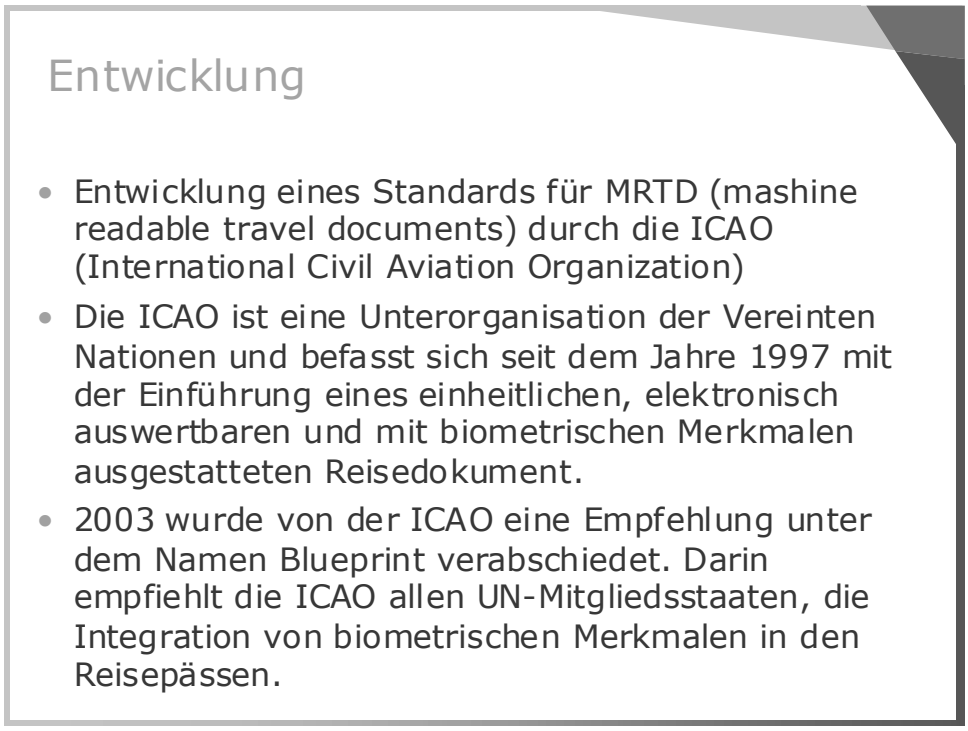
Kreditkartenbezahlung mit SSL

- Händler benötigt Vertrag mit Kreditkartengesellschaft
- Bezahlung per Eingabe von Kreditkartennummer und Ablaufdatum der Karte durch den Kunden
- Händler sieht Karte eigentlich nicht → Verschlüsselung der Daten unausweichlich, damit Daten nicht bei Übertragung gefälscht werden können
- Daten werden mittels SSL verschlüsselt

Kreditkartenbezahlung mit SSL

Schematische Darstellung des Ablaufs:

2. Foliensatz Maschinenlesbare Reisedokumente



Entwicklung II

Geeignete Techniken müssen dabei folgende fünf Kriterien erfüllen:

- weltweite Interoperabilität
- Einheitlichkeit
- technische Zuverlässigkeit
- Praktikabilität
- Haltbarkeit

Entwicklung III

Bei der Verwendung von Blueprint stehen vier zentrale Punkte im Mittelpunkt:

- Verwendung kontaktloser Chips (RFID)
- Chip dient zur Abspeicherung des Lichtbildes, eine Ergänzung weiterer Merkmale (z.B. Fingerabdruck, Irismuster) soll zu einem späteren Zeitpunkt möglich sein
- Verwendung einer definierten, logischen Datenstruktur (engl. Logical Data Structure, LDS)
- Public Key Infrastructure (PKI): Verfahren zur Verwaltung digitaler Zugangsschlüssel

Entwicklung IV

- Vorgaben durch die ICAO im weiterentwickelten Standard 9303 zusammengefasst
- Druck der USA für Einführung von Reisepässen mit maschinenlesbaren biometrischen Daten stieg stetig an → am 13.Dezember 2004 wurde im Rat der Europäischen Union beschlossen, dass alle Mitgliedsstaaten künftig Reisepässe nach dem Standard 9303, mit biometrischen Daten, ausstellen müssen
- Verwendung der Abdrücke beider Zeigefinger → einlesen mittels Scanner → auf Chip gespeichert.
- In Deutschland werden seit dem 1.November 2007 alle neu ausgestellten Reisepässe mit den biometrischen Daten ausgestattet. In Österreich ist noch kein genauer Termin zur Einführung bekannt.

Sicherheitsmechanismen beim ePass

ICAO definiert vier Sicherheitsmaßnahmen, um einen optimalen Schutz der Daten zu gewährleisten:

1. Passive Authentication (PA): das Lesegerät kann überprüfen, ob das Reisedokument von einer zulässigen Zertifizierungsstelle signiert wurde. Damit kann die Gültigkeit des Reisedokuments bestimmt werden.

2. Basic Access Control (BAC): Schützt das Reisedokument vor unberechtigten Lesezugriffen. Somit können Angreifer das Reisedokument nicht unbemerkt auslesen und eventuell kopieren.

Sicherheitsmechanismen beim ePass II

3. Active Authentication (AA): Schützt die Einzigartigkeit und die Authentizität des im Reisedokument befindlichen Chips. Diese Sicherheitsmaßnahme sollte nur verwendet werde, wenn BAC bereits integriert ist und somit die Kommunikation in diesem Schritt entsprechend verschlüsselt wird.

4. Extended Access Control (EAC): Ist ein Mechanismus, mit dem ermöglicht wird, dass auf gewisse Inhalte nur gewisse Lesegeräte Zugriff haben. Eingeführt wurde EAC um die, neu am Chip abgespeicherten, biometrischen Daten gegen unerlaubten Zugriff zu schützen.

Logical Data Structure

- Logical Data Structure (LDS) → logische Datenstruktur
- Speicherverteilung der Daten auf dem RF-Chip
- Speicherstandard benötigt um weltweite Interoperabilität gewährleisten zu können
- größter Vorteil ist die Möglichkeit, die Speicherkapazität jederzeit erhöhen zu können
- Speicher am Chip in so genannte Data Groups (Datengruppen, Abkürzung: DG) unterteilt
- aktuell besteht die LDS aus 16 Datengruppen
- für Zukunft ist eine Erweiterung geplant, um zum Beispiel Visadaten speichern zu können
- bei Einführung der maschinenlesbaren Reisepässe waren nur die DG 1 (maschinenlesbare Daten) und 2 (Gesichtbild) verpflichtend

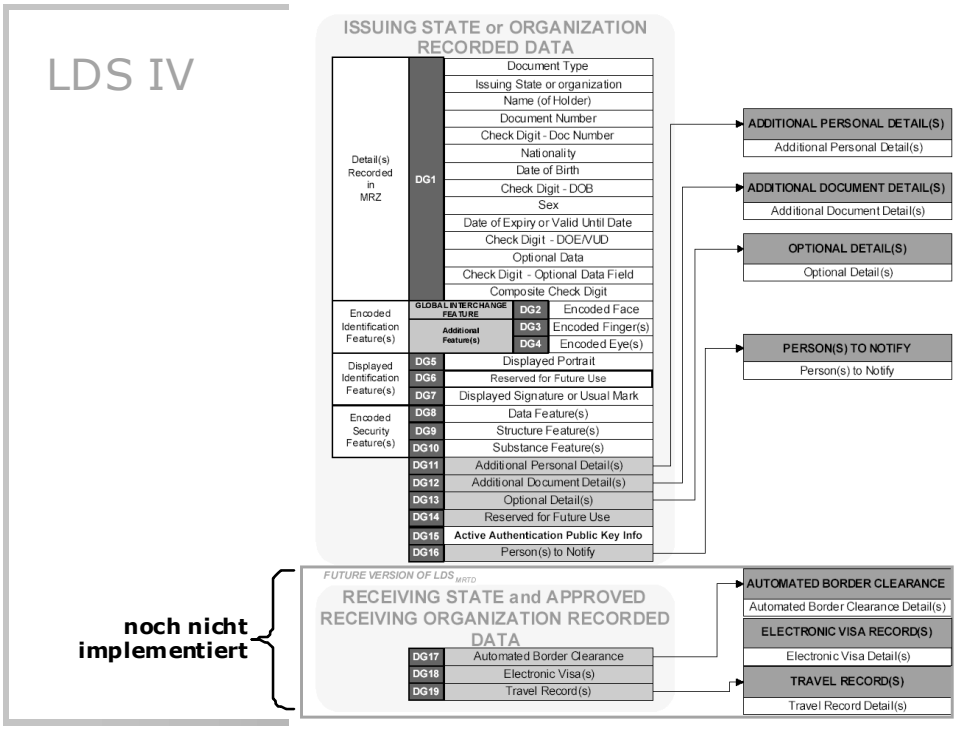
Logical Data Structure II

- die DG1 beinhaltet Daten, die bereits am alten Pass gedruckt waren
- dazu gehören:
  - Dokumententyp
  - Ausstellungsland
  - Persönliche Informationen (Nachname, Vorname, Nationalität, Geburtsdatum, Geschlecht)
  - Ablaufdatum
  - zusätzliche Daten
  - Gesamtprüfziffer

Logical Data Structure III

- DG2 beinhaltet digital abgespeichertes Foto und zusätzliche Informationen dazu
- Foto wird komprimiert gespeichert → geringerer Speicheraufwand
- Foto muss gewissen Anforderungen genügen (siehe Quellen), damit für automatische Gesichtserkennung mittels Software verwendet werden kann
- neueste Generation des Reisepasses soll auch biometrische Daten auf dem Chip integrieren → Abdrücke der Zeigefinger in der DG3 gespeichert
- in Zukunft Erweiterung mit Abbild der Iris möglich → DG4 bereits dafür vorgesehen

LDS IV



Logical Data Structure V

- Problem: Wie weiß das Lesegerät, wo sich welche Daten am Chip befinden?
- Lösung: Implementierung eines Systems, bei dem das Lesegerät zuerst einen Header und eine Data Group Presence Map ausliest.

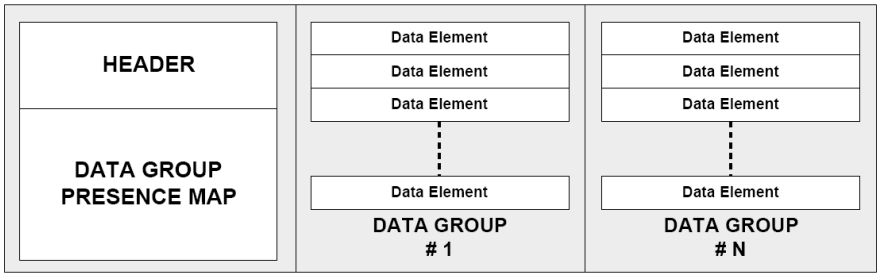
- Header beinhaltet:
  - Anwendungsidentifikation
  - LDS Versionsnummer
  - UNICODE Versionsnummer

- Präsenzliste der Datengruppen besteht aus:
  - variable Anzahl von Tags
  - jeder Tag steht für eine Datengruppe
  - für eine Datengruppe kein Tag vorhanden → keine entsprechenden Daten auf dem Chip.

- Header und Presence Map müssen vorhanden sein.

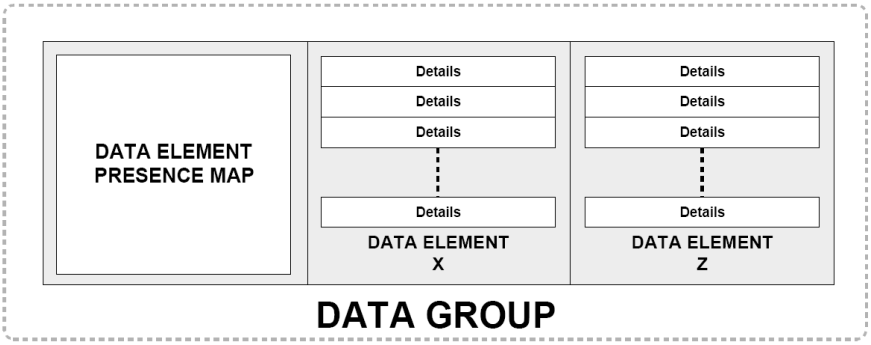
Logical Data Structure VI

Abbildung: Header und Data Group Presence Information



Logical Data Structure VII

- jede Datengruppe beginnt mit Data Element Presence Map zu finden
- gibt Aufschluss über die, in dieser Gruppe, vorhandenen Datenelemente.



Logical Data Structure VIII

- um Sicherheit der Daten zu gewährleisten, wird von der ICAO empfohlen, dass die Datengruppen 1 bis inklusive 15 schreibgeschützt sein sollen.
- zusätzlich sollen digitale Signaturen über die Hashwerte in den Sicherheitsdaten (Security Data) gespeichert werden → Zugriff nur für herausgebendes Land
- DG16 ebenfalls schreibgeschützt und Schreibzugriff nur durch herausgebendes Land
- Datengruppen 17 bis 19 werden erst in der Version 2 der LDS spezifiziert

Extended Access Control

- Ausbaustufe 2 → Integration biometrischer Daten → neuer Schutzmechanismus benötigt → Extended Access Control (EAC)
- EAC wurde, auf Basis einer technischen Richtlinie des Bundesamtes für Sicherheit in der Informationstechnik, von einer Arbeitsgruppe entwickelt
- EAC hat die Aufgabe, die sensitiven personenbezogenen Daten vor unberechtigten Zugriffen zu schützen
- EAC ermöglicht unterschiedliche Vergabe von Zugriffsrechten:
  - nationale Geräte haben automatisch Zugriff auf biometrische Daten
  - für Inspektionssysteme anderer Länder können Zugriffsrechte vergeben werde, oder nicht
- biometrische Daten sind zur Zeit die Abdrücke der Zeigefinger, in Zukunft wäre das Abbild der Iris denkbar

EAC - Grundprinzip

- im Reisepass befindet sich ein Chip mit allen Daten (MRTD Chip)
- bevor Inspektionssystem Zugang zu sensitiven Daten erhält muss es sich beim Chip authentifizieren
  → Terminal-Authentisierung (TA) genannt
  → für Zugriff auf biometrische Daten zwingend benötigt
  → TA basiert auf einer Public-Key-Infrastruktur (PKI, siehe später) und wird erst nach erfolgreich durchgeführter Chip-Authentisierung durchgeführt
- Chip-Authentisierung ist Teil der Basic Access Control (BAC) und hat folgende Aufgaben:
  - Echtheitsüberprüfung des Chips
  - Aufbau einer stark verschlüsselten Kommunikation zwischen Chip und Terminal
- Reihenfolge der Sicherheitsmechanismen garantiert eine sichere, verschlüsselte Übertragung aller persönlichen Daten
- in der EU ist Chip-Authentisierung für alle Inspektionssysteme zwingend vorgeschrieben

EAC –Chip Authentication (CA)

- Alternative zur Active Authentication (AA)
- dient zur Überprüfung des RF-Chips und bietet zwei Vorteile:
  - implizite Echtheitsprüfung der, auf dem Chip gespeicherten, Daten:
    - Zuweisung eines Schlüsselpaars während der Personalisierung
    - privater Schlüssel → in den Speicher des Chips
    - öffentlicher Schlüssel → DG14 der LDS
    - Chip benötigt beide Schlüssel, um Kommunikation mit Inspektionssystem aufbauen zu können → nur echter Chip dazu in der Lage → Kopierschutz der Speicherinhalte implementiert
  - Bereitstellung eines starken Schlüssels für eine sichere Kommunikation
- jeder Chip der CA unterstützt muss auch kompatibel zur Basic Access Control (BAC) sein muss

EAC – Terminal Authentication (TA)

- Hauptaufgabe: Schutz der sensiblen Daten (biometrische Daten wie Fingerabdrücke)
- überprüft, ob das Lesegerät
  - echt ist und
  - ob es über die Berechtigung verfügt, auf biometrische Daten zuzugreifen.
- Lesegerät muss sich beim Chip als berechtigt ausweisen → erst dann erlaubt Chip den Zugriff
- jede Kommunikation muss verschlüsselt erfolgen
- vor der TA muss der Chip überprüft worden sein → CA
- zur Terminal Authentisierung benötigt:
  - Schlüsselpaar (geheimer und öffentlicher Schlüssel)
  - Zertifikatskette (vom Chip verifizierbar)
    - beinhaltet exakte Definition der Zugriffsrechte

EAC - Inspektionsprozedur

- es werden zwei verschiedene Inspektionsprozeduren unterschieden:
  - Standard
  - erweitert
- Wahl der Prozedur abhängig von der Konformität von Chip und Lesegerät bezüglich der Spezifikation
- Auswahlschema:

Inspektionssystem	MRTD - Chip	
	konform	nicht konform
konform	erweitert	Standard
nicht konform	Standard	Standard

EAC – Standard Inspektionsprozedur

Standardprozedur besteht aus folgenden Schritten:

1. Auswahl der ePass Anwendung (erforderlich)
   Vom MRTD Chip wird abhängig von der Verwendung von BAC, Zugriff auf die Daten erlaubt.
   - mit BAC: es soll, mit Ausnahme von generellen Systemdaten, kein Zugriff ermöglicht werden
   - ohne BAC: Lesezugriff auf wenig sensitive Daten soll möglich sein

2. Basis Access Control (unverbindlich)
   Falls erfolgreich abgewickelt, führt der Chip folgendes aus:
   - starten einer sicheren Übertragung
   - Zugriff auf wenig sensitive Daten zulassen
   - beschränkte Zugriffsrechte, um sichere Übertragung zu sichern

EAC – Standard Inspektionsprozedur II

Fortsetzung Standardprozedur:

3. Passive Authentication (erforderlich)
   Vom Lesegerät müssen die Sicherheitsobjekte ausgelesen und verifiziert werden.

4. Active Authentication (optional)
   Das Lesegerät sollte, falls vorhanden, die Daten aus dem Feld DG15 auslesen und verifizieren und die Active Authentication durchführen.

5. Daten lesen und authentifizieren
   Das Lesegerät kann die wenig sensitiven Daten auslesen und verifizieren.

EAC – Erweiterte Prozedur

Erweiterte Inspektionsprozedur besteht aus folgenden Schritten:

- Auswahl der ePass Anwendung (erforderlich)
  MRTD Chip soll keinen Zugriff auf Daten zulassen, ausgenommen sind generelle Systemdaten

2. Basic Access Control (erforderlich)
   Falls erfolgreich abgewickelt, führt der Chip folgendes aus:
   - starten einer sicheren Übertragung
   - Zugriff auf wenig sensitive Daten zulassen
   - eingeschränkte Zugriffsrechte um sichere Übertragung zu verlangen

EAC – Erweiterte Prozedur II

Fortsetzung erweiterte Inspektionsprozedur:

3. Chip Authentication (erforderlich)
   Lesegerät liest Daten des Feldes DG14 und führt CA
   Chip führt folgendes aus:
   - Neustart der sicheren Übertragung
   - beschränkte Zugriffsrechte, um neue sichere Übertragung zu sichern
4. Passive Authentication (erforderlich)
   Lesegerät muss folgendes ausführen:
   - auslesen und verifizieren der Sicherheitsobjekte
   - Daten des Feldes DG14 verifizieren
5. Active Authentication (optional)
   Lesegerät sollte, falls vorhanden, die Daten aus DG15 lesen und verifizieren und AA durchführen

EAC – Erweiterte Prozedur III

Fortsetzung erweiterte Inspektionsprozedur:

6. Terminal Authentication (unverbindlich)
   bei Zugriff auf sensible Daten zwingend vorgeschrieben
   MRTD Chip arbeitet folgende Schritte ab:
   - Zugriff auf alle Datengruppen zulassen, für die das Lesegerät die entsprechenden Rechte besitzt
   - beschränkte Zugriffsrechte, um die von der CA gestartete sichere Übertragung, mit Verwendung des kurzlebigen öffentlichen Schlüssels, zu sichern

7. Daten lesen und authentifizieren
   Lesegerät liest je nach Zugriffsrechten die entsprechenden Datenfelder aus und verifiziert diese

EAC - Ablauf

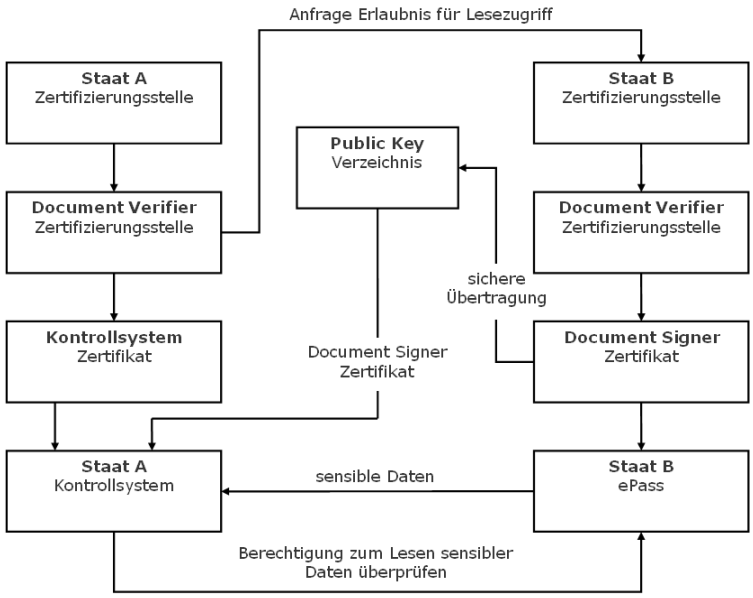
- Person aus Staat b möchte in Staat A einreisen
- Reisepass wird von Lesegerät aus A überprüft

- Arbeit bei Passausstellung:
  - Staat A fordert von B Zugriffsrechte für Reisepassdaten an
  - falls Staat B einverstanden → entsprechende Zertifikate an A übermitteln

1. Schritt bei Passkontrolle: Echtheit des Passes überprüfen (→ CA)
   - Schlüsselgenerierung mittels Diffie-Hellman
   - geheimer Schlüssel erhöht Sicherheit enorm
   - weitere Kommunikation verschlüsselt
   - gegenüber BAC höherer Grad an Sicherheit

2. Schritt bei Passkontrolle: Echtheit des Lesegeräts überprüfen (TA)
   - Verwendung einer Zertifikatskette und aus CA berechneter Schlüssel
   - Zertifikatskette endet mit Public Key der nationalen Wurzelinstanz
   - erfolgreiche Authentifizierung → Zugriff auf biometrische Daten

EAC – Ablauf II



EAC – Public Key Infrastruktur

Public Key Infrastruktur (PKI)

- Verwaltung von
  - Zertifikaten und
  - Signaturen
- Hauptaspekte:
  - Sicherheit
  - Funktion über Grenzen hinaus sichergestellt
  - →stellt Vertrauensstelle dar

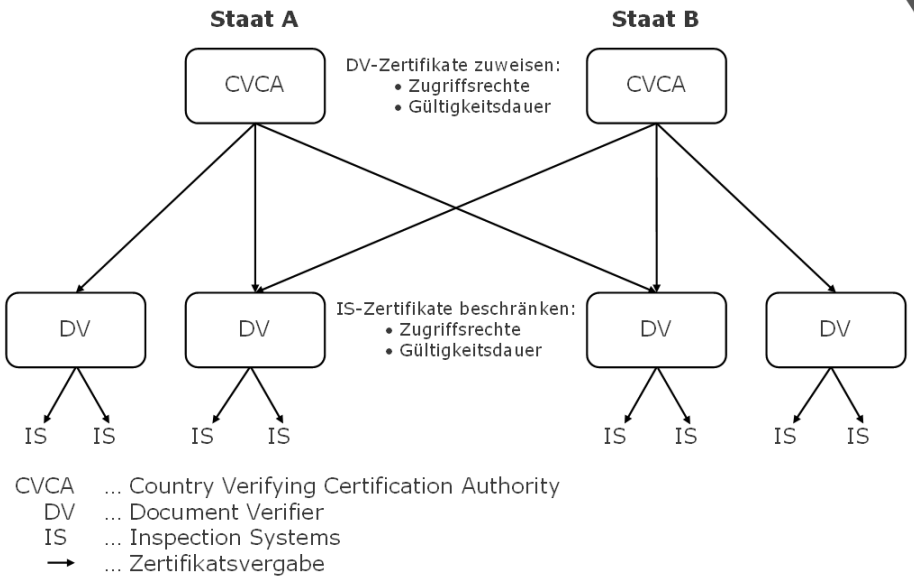
→Lösung aller technischen Anforderungen
→Sicherheit der Reisepässe über Staatsgrenzen hinaus gewährleistet

EAC – PKI Möglichkeiten

PKI bietet drei große Möglichkeiten

1. Bereitstellung öffentlicher Schlüssel:
   - System kann nach Public Keys anderer Benutzer durchsucht werden
   - Gültigkeit öffentlicher Schlüssel kann überprüft werden
2. Verwendung von Schlüssel:
   - zusätzlich zur Ansammlung können Schlüssel auch verwendet werden
   - Hauptverwendung → Verschlüsselung; nicht für Signatur (da privater Schlüssel nötig
3. Verwaltung der Schlüssel:
   - jegliche Tätigkeit, die Schlüssel betreffen, können in PKI durchgeführt werden
   - mögliche Vorgänge: Schlüssel betrachten, anfordern oder zurückziehen oder Vertrauensgrad von Schlüsselausstellern überprüfen und festlegen

PKI - Hierarchie



CVCA ... Country Verifying Certification Authority
DV ... Document Verifier
IS ... Inspection Systems
→ ... Zertifikatsvergabe

PKI – Zertifikatsvergabe für TA

Ablauf:

- CVCA Zertifikat des ausstellenden Landes (Staat B) auf Chip des Reisepasses gespeichert
- Inspektionssystem (IS) von Staat A möchte auf biometrische Daten zugreifen → folgender Ablauf:
  - IS muss Pass beweisen, dass er die Zugriffsrechte besitzt → IS muss also über entsprechende Zertifikate verfügen
  - in unserem Fall:
    - DV Zertifikat des Staates A, das mit dem CVCA Zertifikat des Staates B signiert wurde
  - nachdem Zertifikatskette vom Reisepass überprüft wurde → Kontrolle, ob IS Zugriff auf geheimen Schlüssel, der zum in der Zertifikatskette gespeicherten öffentlichen Schlüssel passt, hat → Überprüfung mittels challenge–response Protokoll → Nachricht wird vom Reisepass an das IS gesendet → Reisepass verschlüsselt mit geheimen Schlüssel → sendet verschlüsselte Nachricht zurück → Reisepass kann mit öffentlichem Schlüssel entschlüsseln → Schlüssel ok → Zugriff auf biometrische Daten wird gewährt

EAC – Abhören der Kommunikation

- zwei Möglichkeiten, wie Daten unberechtigt aus einem Reisepass ausgelesen werden könnten:
  - aktiv: Auslesen der persönlichen Daten durch unauthorisierte Lesegeräte; Schutz durch die Basic Access Control
  - passiv: Abhören der Kommunikation zwischen dem Lesegerät und dem Pass; möglicher Angreifer müsste mit einer speziellen Abhöranlage eine oder mehrere Auslesevorgänge abhören → entschlüsseln mit enormer Rechenleistung verbunden → nicht in sinnvoller Zeit entschlüsselbar

EAC – Abhören der Kommunikation II

- passives Auslesen theoretisch möglich → 2 Fragen:
  1. In welchem Abstand kann die Kommunikation noch abgehört werden?
     - unzählige Spekulationen zu finden, meist aber nicht mit Praxistests untermauert
     - Studie „MARS" [DuD2] durch BSI → Reichweite kontaktloser Übertragungen nach der ISO14443 → ab ca. 2,7m Abstand ist mithören nicht mehr möglich
  2. Wie stark ist die Verschlüsselung der Kommunikation?
     - Triple-DES Verfahren im Cipher Block Chaining Modus → kleiner Bitfehler ergibt großen Unterschied
     - Sitzungsschlüssel: 112 Bit lang entspricht 2048 RSA-Schlüssel
     - ePass 10 Jahre gültig → ICAO empfiehlt:

	RSA	DSA	Elliptic Curve DSA
Country Signer Certificat	3072 bit	256 bit	256 bit
Document Signer Certificate	2048 bit	224 bit	224 bit
Active Authentication Certificate	1034 bit	160 bit	160 bit

EAC - Protokollspezifikation

- Darstellung der kryptografischen Protokolle für die Chip Authentication und die Terminal Authentication
- in Technical Guideline TR-03110 spezifiziert
- involvierte Parteien:
  - MRTD – Chip (als PICC bezeichnet)
  - Inspektionssystem (als PCD benannt)

EAC - Schlüsselvereinbarung

- Spezifikation auf einem verfahrensunabhängigen Weg erläutert
- mögliche Verfahren:
  - Diffie-Hellman (DH) Schlüsselaustausch
  - auf Elliptischen Kurven basierende DH (ECDH)
- folgende Schlüsselpaare werden verwendet:
  - MRTD – Chip: statisches Diffie-Hellman Schlüsselpaar
    - öffentlicher Schlüssel → PK_{PICC}
    - geheimer Schlüssel → SK_{PICC}
    - $\mathcal{D}_{PICC}$ bezeichnet die Domainparameter
  - Inspektionssystem: generiert für jede neue Kommunikation ein kurzlebiges Diffie-Hellman Schlüsselpaar
    - öffentliche Schlüssel → PK_{PCD}
    - geheimer Schlüssel → SK_{PCD}
    - Domainparameter der MRTD – Chips
- ICAO empfiehlt, dass PK_{PCD} vom MRTD-Chip überprüft wird
- Berechnung des gemeinsam verwendeten geheimen Schlüssel K:
  - MRTD – Chip: $KA(SK_{PICC}, PK_{PCD}, \mathcal{D}_{PICC})$
  - Inspektionssystem: $KA(SK_{PCD}, PK_{PICC}, \mathcal{D}_{PICC})$

EAC - Signaturen

- Spezifikation auf einem verfahrensunabhängigen Weg erläutert
- für Signaturen hat Inspektionssystem:
  - öffentlichen Schlüssel → PK_{PCD}
  - privaten Schlüssel → SK_{PCD}
- Operationen zum Nachricht signieren und verifizieren werde wie folgt definiert:
  - Nachricht m mit dem geheimen Schlüssel SK_{PCD} signieren wird durch $s = Sign(SK_{PCD}, m)$ definiert
  - Verifizierung der Signatur s mit dem öffentlichen Schlüssel PK_{PCD} wird durch $Verify(PK_{PCD}, s, m)$ definiert

EAC – Chip Authentication

- Abkürzung: CA
- kurzlebiges Diffie-Hellman Schlüsselvereinbarungsprotokoll
- kurzlebig → Gültigkeitsdauer der Schlüssel sehr kurz
- implementiert:
  - sichere Kommunikation
  - Gültigkeitsüberprüfung des Chips

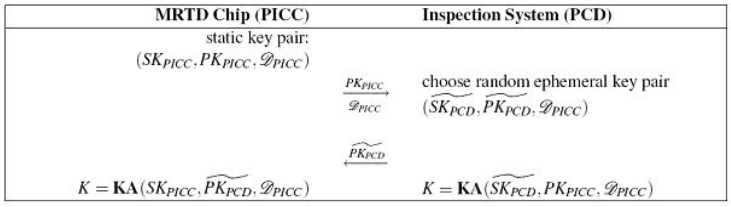
EAC – CA II

Protokollspezifikation:

1. MRTD – Chip sendet öffentlichen Schlüssel PK_{PICC} und Domain Parameter $\mathcal{D}_{PICC}$ zum Inspektionssystem
2. Inspektionssystem berechnet aus erhaltenen Daten das Diffie-Hellman Schlüsselpaar (SK_{PCD},PK_{PCD},$\mathcal{D}_{PICC}$) und sendet seinen öffentlichen Schlüssel PK_{PCD} zum MRTD – Chip
3. Chip und Inspektionssystem berechnen gleichzeitig:
   - Berechnung des gemeinsam verwendeten geheimen Schlüssel $K=KA(SK_{PICC},PK_{PCD},\mathcal{D}_{PICC})=KA(SK_{PCD},PK_{PCD},\mathcal{D}_{PICC})$
   - für sichere Nachrichtenübertragung werden KMAC und K_{Enc} aus K berechnet
   - zur Terminal Authentication wird der Hashwert $H(PK_{PCD})$ des öffentlichen Schlüssels des Inspektionssystems ermittelt

EAC – CA III

- Spezifikation der CA nach [xx]:



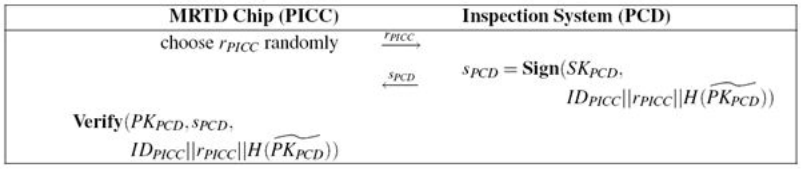
- Sicherheitsstatus
  - CA erfolgreich durchgeführt: Neustart der sicheren Nachrichtenübertragung mit den neu berechneten Sitzungsschlüssel K_{MAC} und K_{Enc}
  - CA nicht erfolgreich durchgeführt: verschlüsselte Übertragung der Daten wird mit den in der BAC berechneten Schlüssel weitergeführt

EAC – Terminal Authentication

- Abkürzung: TA
- die TA beinhaltet folgende Schritte:
  1. Inspektionssystem sendet Zertifikatskette an den Chip; Zertifikatskette beginnt mit öffentlichem Schlüssel einer CVCA und endet mit dem IS Zertifikat des Inspektionssystems
  2. MRTD – Chip verifiziert die Zertifikate und holt sich den öffentlichen Schlüssel des Inspektionssystems (PK_{PCD}) → Chip schickt r_{PICC} an das Inspektionssystem
  3. Inspektionssystem reagiert mit der Antwort: $s_{PCD}=Sign(SK_{PCD},ID_{PICC}||r_{PICC}||H(PK_{PCD}))$
  4. Chip überprüft die erhaltenen Daten: $Verify(PK_{PCD},s_{PCD},ID_{PICC}||r_{PICC}||H(PK_{PCD}))=true$

EAC – TA II

- Spezifikation der TA nach [xx]:



- ID_{PICC} → Dokumentnummer inklusive der Checkzahl
- laut Spezifikation [xx] MUSS jeglicher Datenaustausch verschlüsselt erfolgen
- verwendete Schlüssel aus CA
- erfolgreiche Beendigung der TA → Inspektionssystem erhält Zugriff auf sensible Daten

EAC - Funktionsbeispiele

- Beispiele für DG14 die Spezifikation erfüllen:
  1. Beispiel für DG14 auf Basis von Elliptische Kurven Diffie-Hellman
  2. Beispiel für funktionsfähiges System auf Diffie-Hellman basierend

  → Abbildungen aus Technical Guideline TR-03110 (Advanced Security Mechanisms for Machine ReadableTravel Documents – Extended Access Control (EAC), Version 1.1).

EAC – Beispiel ECDH I

- Hexadezimale Darstellung der DG14:

```
0000 : 6E82014A 31820146 30820122 06090400 7F000702 02010230 82011330 81D40607
0020 : 2A8648CE 3D020130 81C80201 01302806 072A8648 CE3D0101 021D00D7 C134AA26
0040 : 4366862A 18302575 D1D787B0 9F075797 DA89F57E C8C0FF30 3C041C68 A5E62CA9
0060 : CE6C1C29 9803A6C1 530B514E 182AD8B0 042A59CA D29F4304 1C2580F6 3CCFE441
0080 : 38870713 B1A92369 E33E2135 D266DBB3 72386C40 0B043904 0D9029AD 2C7E5CF4
00A0 : 340823B2 A87DC68C 9E4CE317 4C1E6EFD EE12C07D 58AA56F7 72C0726F 24C6B89E
00C0 : 4ECDAC24 354B9E99 CAA3F6D3 761402CD 021D00D7 C134AA26 4366862A 18302575
00E0 : D0FB98D1 16BC4B6D DEBCA3A5 A7939F02 0101033A 0004680E C4FF3851 12D9A401
0100 : 76D36733 157B11FC 08B4A280 CE9B8246 4D765C38 C21CB883 6EE05724 3C1EBC7B
0120 : B80EC484 41107C38 E4F545EB 213C300F 060A0400 7F000702 02030201 02010130
0140 : 0D060804 007F0007 02020202 0101…. …… …… …… ……
```

- Privater Schlüssel für CA:

```
0000 : 12528622 D8947E85 E4988853 69ECDCAB F10E343A F7B95A99 DF610031
```

EAC – Beispiel ECDH II

- Speicherstruktur der DG14

Tag	Length	Value	ASN.1 Type	Comment
6E	82 01 4A		-	Application specific tag "14"
31	82 01 46		SET	Set of SecurityInfos
30	82 01 22		SEQUENCE	SecurityInfo
06	09	04 00 7F 00 07 02 02 01 02	OBJECT IDENTIFIER	ChipAuthenticationPublicKeyInfo CA with ECDH
30	82 01 13		SEQUENCE	SubjectPublicKeyInfo
30	81 D4	★	SEQUENCE	AlgorithmIdentifier
03	3A	★	BIT STRING	SubjectPublicKey
–	–	–		optional keyId is unused
30	0F		SEQUENCE	SecurityInfo
06	0A	04 00 7F 00 07 02 02 03 02 01	OBJECT IDENTIFIER	ChipAuthenticationInfo CA with ECDH
02	01	01	INTEGER	Version 1
–	–	–		optional keyId is unused
30	0D		SEQUENCE	SecurityInfo
06	08	04 00 7F 00 07 02 02 02	OBJECT IDENTIFIER	TerminalAuthenticationInfo
02	01	01	INTEGER	Version 1
–	–	–		optional FileID is unused

★ **... siehe auf den nächsten Folien**

EAC – Beispiel ECDH III

- AlgorithmIdentifier

Tag	Length	Value	ASN.1 Type	Comment
30	81 D4		SEQUENCE	AlgorithmIdentifier
06	07	2A 86 48 CE 3D 02 01	OBJECT IDENTIFIER	Elliptic Curve Public Key
30	81 C8		SEQUENCE	Domain parameter
02	01	01	INTEGER	Version
30	28		SEQUENCE	Underlying field
06	07	2A 86 48 CE 3D 01 01	OBJECT IDENTIFIER	Prime field
02	1D	00 D7 C1 ... C8 C0 FF	INTEGER	Prime p
30	3C		SEQUENCE	Curve equation
04	1C	68 A5 E6 ... D2 9F 43	OCTET STRING	Parameter a
04	1C	25 80 F6 ... 6C 40 0B	OCTET STRING	Parameter b
–	–	–		Optional seed is unused
04	39	★ siehe nächste Folie	OCTET STRING	Group generator G
02	1D	00 D7 C1 ... A7 93 9F	INTEGER	Group order n
02	01	01	INTEGER	Cofactor f

EAC – Beispiel ECDH IV

- Group Generator
  - Punkt auf der elliptischen Kurve, der für die Verschlüsselung verwendet wird
  - x- und y-Koordinaten des Punktes in hexadezimaler Form gespeichert

Tag	Length	Value	ASN.1 Type	Comment
04	39		OCTET STRING	Encoded group generator
		04	–	Uncompressed point
		0D 90 29 ... 12 C0 7D	–	x-coordinate
		58 AA 56 ... 14 02 CD	–	y-coordinate

EAC – Beispiel ECDH V

- Sequenz Encoded public key enthält die Koordinaten eines Punktes auf der elliptischen Kurve
- Speichertyp kommt Bitstring zur Anwendung

Tag	Length	Value	ASN.1 Type	Comment	
03	3A		BIT STRING	Encoded public key	
		00			Number of unused bits
		04	–		Uncompressed point
		68 0E C4 ... 46 4D 76	–		x-coordinate
		5C 38 C2 ... EB 21 3C	–		y-coordinate

EAC – Beispiel ECDH VI

- Zusammenfassung verwendeter Schlüssel:
  - zufällig gewähltes, nur kurz gültiges Schlüsselpaar → öffentlicher und privater Schlüssel
  - durch Schlüsselaustausch berechneter, geheimer gemeinsamer Schlüssel

Private Key	7756F0C5 D1AB06C0 03672668 2B720C2F B1D5F789 B58244A6 DC07E5A2
Public Key	69D489F6 8A99ABC8 7106B3E1 3A52C6AF 2C57CEE5 72755FE3 712C8AC3, 8A6A3E9F E0694482 31BDC1BE FC826035 67E72602 EBA5C3EE EEAC3F15
Shared Secret	A770F66A CC78ED59 0581CC82 033C79F3 3BECE0A2 0C280244 79A4E97C

  - berechnete Sitzungsschlüssel

K_{Enc}	DF94ED65 8A5CCB35 5FFFE612 8BADB584
K_{MAC}	1297F052 5DD9DE75 917DCD90 848465C1

  - Chip speichert Public Key vom Inspektionssystem

$H(\widetilde{PK_{PCD}})$	69D489F6 8A99ABC8 7106B3E1 3A52C6AF 2C57CEE5 72755FE3 712C8AC3

EAC – Beispiel DH I

- Hexadezimale Darstellung der DG14:

```
0000 : 6E8201DC 318201D8 308201B4 06090400 7F000702 02010130 8201A530 82011A06
0020 : 092A8648 86F70D01 03013082 010B0281 8100DCB5 54DF8C69 31E865C1 B588273D
0040 : 80A2D87A B539C5E4 A074B402 49FF655A 9AB83063 3B457C4C F885E31C D79F8114
0060 : 8C8A68D1 DBFC2F7B 70ED55C0 387C23A0 479A9572 E8A6714F 418A6BF9 B00EC5BC
0080 : 4DEF255A 9485058A 4271008B A694AA62 CC18385E F9D7B6E8 33A7088A C817AA1F
00A0 : 9B93A86B 983EAB73 C15884E7 33665659 CA7D0281 802E69FE 94D3C0A4 378C8A47
00C0 : 9D83091A ED419234 25C10300 8C6AB3F6 E83E20CB 16C4AE0B 0E28ED9B C79CD7D7
00E0 : E9DFD39D D0A39141 F2DD5714 9AB688DB AD177C68 6F771828 E5A04408 512F1564
0100 : 74B0BFD4 30CBBF91 C01589E7 21DDDFFC DF450043 EB771E61 084C597F 7AEA9048
0120 : 420A2180 EBFEC1B3 B93C1A6C B1AD38B3 984FF052 10020203 F9038184 00028180
0140 : 553CE735 ECF5CBF2 029D30FA A4F97335 DF404047 E4F8586D 76A7D221 A09E7F55
0160 : BBE255C6 587BF288 5D41B786 BCEF2177 D52BF3CD BA785D37 D70B88D6 AB4E1CA6
0180 : 6A63B601 1376ED44 444A662B D0DC9524 176E9712 87AD41D2 9BED3D35 EAC7D39C
01A0 : A73ECB2A 3B4D3967 1CE4125C 92658C5B F3DEDA91 5ED71B88 FC031BAB 887248A1
01C0 : 300F060A 04007F00 07020203 01010201 01300D06 0804007F 00070202 02020101
```

- Privater Schlüssel für CA:

```
0000 : 01CD4A70 FFDC3D42 C862FD5E 3D781EB6 DE97677B 4FF61319 242E1499 B5CD1908
0020 : A9B54221 135D1EDB AD787A5C E37586CF E86A61C4 78187157 267C97B4 0A7F2727
0040 : B9B92FAC EC267CC0 1C883FA3 783BA07D C090EE04 99C9CE88 C684C874 0FEB84F4
0060 : 49B0F544 C1747716 46BDB7A3 0B0E3AB5 6C655F0E 83A98A4B A99EB9F1 0B0C0FBF
```

EAC – Beispiel DH II

- Speicherstruktur der DG14:

Tag	Length	Value	ASN.1 Type	Comment
6E	82 01 DC		-	Application specific tag "14"
31	82 01 D8		SET	Set of SecurityInfos
30	82 01 B4		SEQUENCE	SecurityInfo
06	09	04 00 7F 00 07 02 02 01 01	OBJECT IDENTIFIER	ChipAuthenticationPublicKeyInfo CA with DH
30	82 01 A5		SEQUENCE	SubjectPublicKeyInfo
30	82 01 1A	★	SEQUENCE	AlgorithmIdentifier
03	81 84	★	BIT STRING	SubjectPublicKey
–	–	–	–	optional keyId unused
30	0F		SEQUENCE	SecurityInfo
06	0A	04 00 7F 00 07 02 02 03 01 01	OBJECT IDENTIFIER	ChipAuthenticationInfo CA with DH
02	01	01	INTEGER	Version 1
–	–	–		optional keyId unused
30	0D		SEQUENCE	SecurityInfo
06	08	04 00 7F 00 07 02 02 02	OBJECT IDENTIFIER	TerminalAuthenticationInfo
02	01	01	INTEGER	Version 1
–	–	–		optional FileID is unused

★ **... siehe auf den nächsten Folien**

EAC – Beispiel DH III

- AlgorithmIdentifier

Tag	Length	Value	ASN.1 Type	Comment
30	82 01 1A		SEQUENCE	AlgorithmIdentifier
06	09	2A 86 48 86 F7 0D 01 03 01	OBJECT IDENTIFIER	PKCS#3 dhKeyAgreement
30	82 01 0B		SEQUENCE	Domain parameter
02	81 81	00 DC B5 ... 59 CA D7	INTEGER	Prime *p*
02	81 80	2E 69 FE ... F0 52 10	INTEGER	Group generator *g*
02	02	03 F9	INTEGER	Private key length

- Encoded Public Key

Tag	Length	Value	ASN.1 Type	Comment
03	81 84		BIT STRING	Encoded public key
		00		Number of unused bits
(02)	(81 80)	(55 3C E7 ... 72 48 A1)	(INTEGER)	Public key

EAC – Beispiel DH IV

- Schlüssel zusammengefasst:
  - Public und Private Key: zufällig gewählt, kurze Gültigkeitsdauer
  - durch Schlüsselaustausch berechneter, geheimer gemeinsamer Schlüssel

Private Key	0170A377 AA4B612B 69A6762E CD71A91C 3D7CD149 A870F37 F357A196F F1134BF7 E0B33DDC EC645560 54EA9959 23189BDB 3893656F E05F8DA BE67F8998 3799E16F 9BF7A9CA 8050C949 31BAB4D8 CAA5F84B 33D71ACA 77A817C BC44CA92C 4B8960A2 034FBC31 999E7DEE 025E1001 EAF96113 BD06EFED FBBD5F2 E916ADC73 1971F019
Public Key	8EBC4457 EABBEF83 65D6EF83 9A1A3672 449486B2 779EF88E B0198ADD C64A096B 0AFC3C26 4D64EDFD F543C03B AEC7ED5D 58C2F2A1 4B63BB5D 280E62FE CAC0A6DD F255CEB9 2AB51C0D B672A251 68934F86 7B95552A 189A3244 4AF890B7 7509ED92 EC5A81A7 D787F5F5 51B37EB2 FA3A49C7 787B5F61 3527649C 151C1786 8417E9CA
Shared Secret	C30AAE5F DC23EFF6 E477734A C318D32D F128AF25 2542087F 1FA239DD 3734DE5C C7E154ED 93BDEC78 E87CD691 6307976F B2603425 133A61D4 10F7F050 EA0797B3 59A5009F 20C9BF0D 227C8866 B2C701FB 04ADF646 B138B1D6 D2623C17 AB3A910A 5A2C72E6 2B554B3F B5B2C310 A1F3334E D4C6AA77 919EA912 5A147C64 9C9E6556

EAC – Beispiel DH V

- weitere Schlüssel:
  - berechnete Sitzungsschlüssel

K_{Enc}	EFF63AC6 29184F19 99C69B7C 3BFA4F17
K_{MAC}	7AD463F3 6997CB2B CB3D1B88 2CE8E4A7

  - auf Chip abgespeicherter Public Key des Inspektionssystems

$H(\widetilde{PK_{PCD}})$	97D9AC36 0DCA6BB0 F2699B85 2DE37793 C29458CD

Literatur

Andreas Wolf (2007), *Angewandte Biometrie 7. Veranstaltung: Elektronischer Pass*; Friedrich-Schiller-Universität Jena, Zugriff am 14.04.2008, http://www.inf-cv.uni-jena.de/uploads/media/FSU-Biometrie-2007-VL7_01.pdf

Alexander Klink (2005), *Die Anforderungender ICAO an Reisedokumente mit Biometrie*; Technische Universität Darmstadt

Bruce Schneier (2006), *Angewandte Kryptographie*; München: Pearson Studium

Bundesamt für Sicherheit in der Informationstechnik (2007)*, Technical Guideline TR-03110, Advanced Security Mechanisms for Machine Readable Travel Documents – Extended Access Control (EAC) Version 1.1*; Zugriff am 14.04.2008, http://www.bsi.de/fachthem/epass/EACTR03110_v110.pdf

Bundesdruckerei (2006), *A BUNDESDRUCKEREI POCKET GUIDE TO ePASSPORT SYSTEMS*; Zugriff am 14.04.2008, http://www.bundesdruckerei.de/de/wissen/download/untern_epassport_system.pdf

Bundesdruckerei (2007), *A BUNDESDRUCKEREI POCKET GUIDE TO BORDER CONTROL*; Zugriff am 14.04.2008, http://www.bundesdruckerei.de/de/wissen/download/unter_epassport_border_control.pdf

Bundesdruckerei (2008), Trustcenter Pocketguide; Zugriff am 14.04.2008, http://www.bundesdruckerei.de/de/wissen/download/service_trustcenter.pdf

Chaos Computer Club e.V. (2004), CCC: Biometrische Merkmale in Ausweisen erhöhen Sicherheit nicht; Pressemitteilung vom 21. Oktober 2004, Zugriff am 14.04.2008, http://www.ccc.de/press/releases/2004/CCC20041020-PE-BIOM.html

Dennis Kügler und Ingo Naumann (2006), *Sicherheitsmechanismen für kontaktlose Chips im deutschen Reisepass*; Datenschutz und Datensicherheit 31 (2007), Seite 176-180

Europäisches Parlament und Rat (2000), *Richtlinie 2000/46/EG Des EUROPÄISCHEN PARLAMENTS UND DES RATES vom 18.September 2000 über die Aufnahme, Ausübung und Beaufsichtigung der Tätigkeit von E-Geld-Instituten*; Zugriff am 14.04.2008, http://eur-lex.europa.eu/LexUriServ/site/de/oj/2000/l_275/l_27520001027de00390043.pdf

Günter Müller und Martin Reichenbach (2001), *Sicherheitskonzepte für das Internet*; Berlin Heidelberg: Springer

ICAO (2004), *Machine Readable Travel Documents – Development of a Logival Data Structure LDS*; Version 1.7, Zugriff am 14.04.2008, http://mrtd.icao.int/images/stories/Doc/ePassports/Logical%AC%AC_Data_Structure(LDS)_version1.7.pdf

ICAO (2004), *Machine Readable Travel Documents - PKI for Machine Readable Travel Documents offering ICC Read-Only Access*; Version 1.1, http://mrtd.icao.int/images/stories/Doc/ePassports/PKI_for_Machine_Readable_Travel_Documents_offering_ICC_read-only_access_v1.1.pdf

ICAO (2005), *Machine Readable Travel Documents - Part 1: Machine Readable Passports, Volume 2: Specifications for Electronically Enabled Passports with Biometric Identification Capability*; 6th Edition

Manhard Schlifni (1996), Diplomarbeit: *Electronic Cash in verteilten Systemen*; Technische Universität Wien

Markus Stolpmann (1997), *Elektronisches Geld im Internet: Grundlagen, Konzepte, Perspektiven*; Köln[u.a.]: O'Reilly

Markus Ullman, *Extended Access Control (EAC)*; Zugriff am 14.04.2008, traumtenza.tr.funpic.de/seiten/studie/daten/mob_si_II/05_ExtendedAccessControl.pdf

Simon Lofthouse (2006), *ePassport Extended Access Control*, White Paper; Zugriff am 14.04.2008, www.securitydocumentworld.com/client_files/eac_white_paper_210706.pdf

Thomas Gasser (2005), Magisterarbeit: *Der elektronische Reisepass mit Chip und PKI*; Technische Universität Wien

Wolfgang Klas (2006), Electronic Commerce 1; Fakultät für Informatik an der Universität Wien

Zdeněk Říha und Jan Löschner (2007*), Electronic Passports – the Interoperability Aspects*; Zugriff am 14.04.2008, http://www.enisa.europa.eu/doc/pdf/Workshop/June2007/Papers/ENISA_JRC_ePassport_paper_v3.pdf