Einsatz von Kryptographie zum Schutz von Daten

Mathias Neuhaus
bis 02/2011 bei: cv cryptovision GmbH
Munscheidstr. 14, 45886 Gelsenkirchen
mathias@dokom.net

Der Beitrag bietet einen Überblick über in der modernen Kryptographie eingesetzte Verfahren und bewertet deren Eignung für die Zwecke der Verschlüsselung oder Signatur. Im zweiten Teil werden die Aufgaben der im INSIKA Projekt eingesetzte Smart Card (TIM) beschrieben. Die Smart Card bildet die zentrale Instanz zur Absicherung der in der Kasse erfassten Umsatzdaten gegen eine nachträgliche Manipulation.

1 Kryptographie

1.1 Grundlagen der Kryptographie

1.1.1 Ziele beim Einsatz von Kryptographie

Zur Erreichung unterschiedlicher Ziele werden unterschiedliche kryptographische Verfahren eingesetzt. Typische Ziele umfassen:

Geheimhaltung: Zum Schutz gegen unbefugtes Lesen können Dokumente verschlüsselt werden.

Integrität: Eine Veränderung an einem Dokument lässt durch einen Message Authentication Code (MAC) oder eine digitale Signatur zweifelsfrei nachweisen.

Authentizität: Der Urheberschaft eines Dokumentes lässt sich durch eine digitale Signatur dokumentieren.

Nicht-Bestreitbarkeit: Durch eine digitale Signatur lässt sich der Verfasser eines Dokumentes zweifelsfrei ermitteln. Damit ist es für diesen auch nicht möglich die Urheberschaft abzustreiten.

1.1.2 Kryptographische Paradigmen

Bei der Auswahl eines geeigneten Verfahrens sollten einige grundlegende Überlegungen nicht außer Acht gelassen werden.

Auguste Kerkhoffs formulierte schon 1883 seinen Grundsatz für die moderne Kryptographie, dass die Sicherheit eines kryptographischen Verfahrens durch Geheimhaltung des Schlüssels, nicht aber durch alleinige Geheimhaltung des Verfahrens beruhen darf (Kerkhoffs' Paradigma). Dies steht im klaren Gegensatz zum leider viel zu häufig angewandten „Security by Obscurity".

Die Forderungen nach Praxisnähe und absoluter Sicherheit schließen einander weitgehend aus. So bieten Einmalschlüssel zwar eine absolute Sicherheit. Leider ist dieses Verfahren in der Praxis nicht anwendbar, da die sichere Übertragung der Schlüssel denselben Aufwand erfordern würde wie die sichere Übertragung der Daten selbst.

Die Sicherheit heute praktisch einsetzbarer Verfahren basiert auf Annahmen aus der Zahlen- und Komplexitätstheorie.

1.2 Kryptographische Verfahren

Die in der modernen Kryptographie verwendeten Verfahren lassen sich grob in symmetrische, asymmetrische und sonstige Verfahren einteilen.

1.2.1 Symmetrische Verfahren

Symmetrische Verfahren verwenden einen einzelnen geheimen Schlüssel. Sie basieren typischerweise auf einfachen Bitoperationen.

Bild 1 stellt den typischen Verlauf einer Ver- und Entschlüsselung mit symmetrischer Kryptographie dar. Dabei wird die zu sichernde Nachricht mittels des geheimen Schlüssels verschlüsselt. Die Nachricht wird an den Empfänger übertragen. Dieser kann die Nachricht dann mit demselben geheimen Schlüssel wieder in eine lesbare Form entschlüsseln.

Die Vorteile der symmetrischen Verfahren liegen

PTB-Bericht IT-18
doi:10.7795/210.20130206a

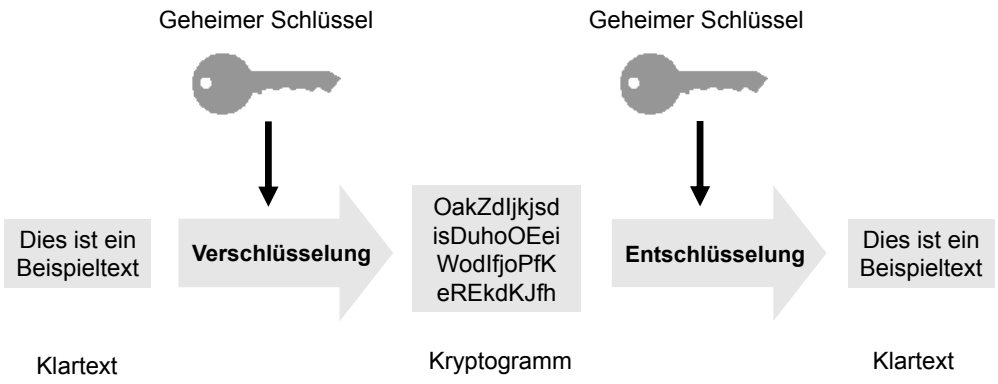

Abbildung 1: Symmetrische Verschlüsselung

in der leichten Implementierbarkeit (in Hardware und Software) und der erzielbaren hohen Performance.

Der wesentliche Nachteil der symmetrischen Verfahren ist die Verwaltung der benötigten Schlüssel.

Wird ein einziger geheimer Schlüssel für alle Kommunikationsteilnehmer verwendet, ist die eindeutige Zuordnung des Schlüssels zu einem Teilnehmer nicht mehr möglich und damit die Forderung nach der Nicht-Bestreitbarkeit nicht erfüllbar. Darüber hinaus wäre durch die Offenlegung des Schlüssels die gesamte Kommunikation kompromittiert.

Alternativ kann man für je zwei Kommunikationsteilnehmer einen eigenen Schlüssel verwenden. Damit werden bei n Teilnehmern $n^2 - 1$ Schlüssel benötigt – was nur mit erheblichem Aufwand zu verwalten ist.

Typische Verfahren dieser Gruppe sind DES, Triple-DES, AES oder RC4. Im Einsatz sind diese Verfahren wegen der mangelhaften Schlüsselverwaltung praktisch nur für die Verschlüsselung.

1.2.2 Asymmetrische Verfahren

Asymmetrische Verfahren verwenden ein Schlüsselpaar aus einem privaten und einem öffentlichen Schlüssel. Diese Verfahren basieren auf komplexer Langzahlenarithmetik.

Bild 2 stellt den Vorgang einer digitalen Signatur (und Verifikation) unter Verwendung asymmetrischer Kryptographie grafisch dar. Dabei wird die abzusichernde Nachricht zunächst gehashed und der Hashwert mit dem privaten Schlüssel des Absenders signiert. Nachricht und Signatur werden an den Empfänger übertragen. Der Empfänger führt nun dieselbe Hashberechnung durch und kann dann mit dem öffentlichen Schlüssel des Absenders die Gültigkeit der Signatur überprüfen.

Der große Vorteil asymmetrischer Verfahren liegt in der einfachen Schlüsselverwaltung. Der öffentliche Schlüssel kann problemlos an alle Teilnehmer verteilt werden, ohne die Sicherheit des Verfahrens zu gefährden. Durch die eindeutige Zuordnung eines Schlüssels zu einem Teilnehmer sind auch die Forderungen nach Authentizität und Nicht-Bestreitbarkeit erfüllbar.

Nachteile sind die aufwändige Implementierung und die geringe Performance.

Typische Vertreter dieser Gruppe sind RSA und ECC. Asymmetrische Verfahren werden aufgrund der begrenzten Performance praktisch nur für elektronische Signaturen und den Schlüsselaustausch eingesetzt.

1.2.3 Sonstige Verfahren

Aus der Gruppe der sonstigen Verfahren sollen hier nur die Hashfunktionen und Zufallszahlengeneratoren erwähnt werden.

Hashfunktionen werden eingesetzt, um einen kurzen „kryptographischen Fingerabdruck“ eines Datensatzes (z.B. einer Nachricht) zu erzeugen. Realisiert wird das durch eine kollisionsfreie Einwegfunktion. Kollisionsfrei bedeutet dabei, dass unterschiedliche Eingabedaten zu unterschiedlichen Ergebnissen führen müssen; der erzeugte Fingerabdruck lässt nicht auf die ursprünglichen Daten zurückschließen (Einwegfunktion). Beispiele für Hashfunktionen sind RIPEMD160, SHA-1 oder SHA-2 (SHA-256) [1].

Zufallszahlengeneratoren (RNG) erzeugen „kryptographisch nutzbare“ Zufallszahlen. Diese Zufallszahlen müssen insbesondere statistisch zufällig und nicht voraussagbar sein. Im Einsatz sind Hardware-RNG und Pseudo-RNG. Pseudo-RNG werden in Software realisiert und bedürfen einer möglichst zufälligen Initialisierung. Ein häufig eingesetzter Pseudo-RNG ist im Standard FIPS 186-2 definiert [2].

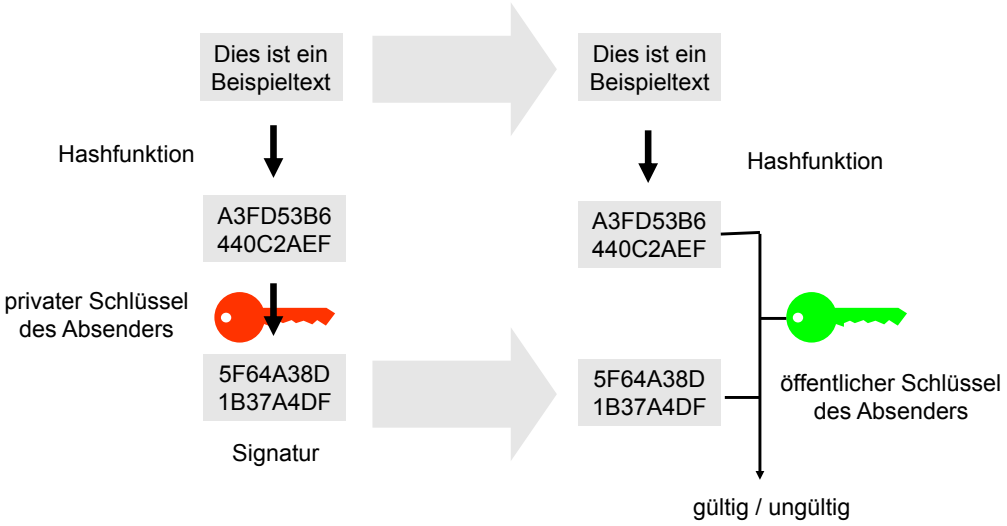

Abbildung 2: Digitale Signatur mit asymmetrischer Kryptographie

1.3 Public Key Infrastruktur (PKI)

Eine Public Key Infrastruktur dient der Verifikation und sicheren Verteilung von öffentlichen Schlüsseln für asymmetrische Verfahren. Eine zentrale Instanz – das TrustCenter (TC) – bietet dazu eine gemeinsame Basis für alle Kommunikationsteilnehmer.

Bild 3 stellt die Abläufe beim Einsatz einer PKI zur Schlüsselverwaltung dar. Das TrustCenter erstellt aus dem öffentlichen Schlüssel des Teilnehmers A zusammen mit einem eindeutigen Identifikationsmerkmal (Name, Adresse) ein Zertifikat und signiert dieses Zertifikat mit seinem privaten Schlüssel. Jeder andere Teilnehmer (hier B benannt) kann die Echtheit des Zertifikates – und damit auch die Echtheit und die Zuordnung des enthaltenen Schlüssels – anhand der Signatur des TC validieren.

Wichtige Voraussetzung für die Nutzung einer PKI ist damit natürlich das Vertrauen aller Teilnehmer in die Integrität des TrustCenters.

2 Signaturverfahren für INSIKA

2.1 RSA

RSA war das erste asymmetrische kryptographische Verfahren. Es wurde im Jahr 1977 durch Ron Rivest, Adi Shamir und Leonard Adleman am MIT entwickelt und ist seit 2000 patentfrei nutzbar. RSA bietet Algorithmen für typische kryptographische Anwendungen wie Signatur und Verschlüsselung, aber kein generisches Verfahren zum Schlüsselaustausch.

RSA basiert auf dem „Problem der Faktorisierung“. Dabei wird genutzt, dass die Multiplikation zweier langer Zahlen (mehr als 100 Dezimalstellen) sehr leicht berechenbar ist, die Umkehroperation – die Zerlegung einer Langzahl (mit mehr als 200 Dezimalstellen) in ihre Primfaktoren – aber nur mit erheblich höherem Aufwand zu leisten ist.

2.2 ECC

Elliptic Curve Cryptography (ECC) wurde im Jahr 1985 „erfunden“ und ist eine heute sehr populäre Alternative zu RSA. ECC bietet Algorithmen für Signatur, Verschlüsselung und Schlüsselaustausch.

ECC basiert auf dem „Problem des Diskreten Logarithmus“. Es nutzt aus, dass eine (modulare) Exponentiation leicht berechenbar ist, die Umkehrung – die Berechnung des „Diskreten Logarithmus“ – aber wesentlich höheren Aufwand erfordert.

2.3 Vergleich RSA – ECC

Für die Erreichung vergleichbarer Sicherheit werden bei ECC wesentlich kürzere Parameter als bei RSA benötigt. Durch die Verwendung kürzerer Parameter ist eine höhere Performance erreichbar. So bietet ECC mit 192 Bit Schlüssellänge eine vergleichbare Sicherheit wie RSA mit 2048 Bit Schlüsseln.

Darüberhinaus skaliert ECC besser als RSA (siehe Bild 4). Die notwendige Schlüssellänge steigt bei ECC linear mit der geforderten Sicherheit, bei RSA jedoch exponentiell, so dass in Zukunft der Vorteil von ECC gegenüber RSA noch größer wird.

Als Beispiele für den Einsatz von ECC seien hier der elektronische Reisepass und der neue Personalausweis genannt.

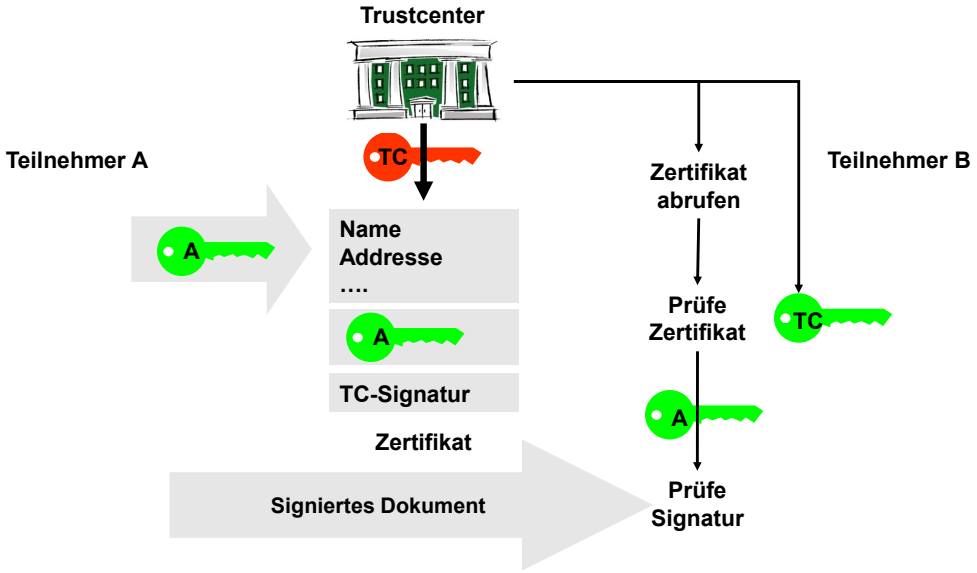

Abbildung 3: Public Key Infrastruktur

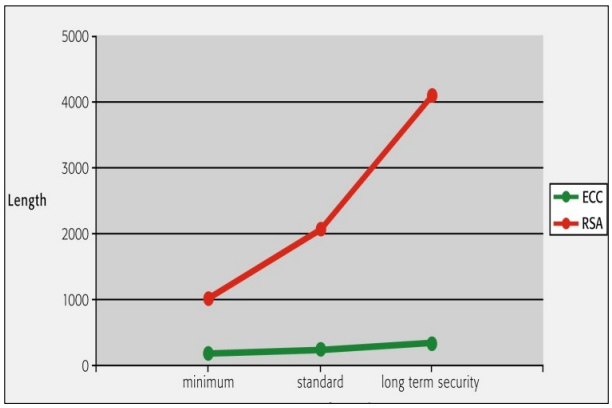

Abbildung 4: Skalierung RSA & ECC

Die Sicherheit einer Signatur steht und fällt mit der verwendeten Schlüssellänge. Die Bundesnetzagentur und das BSI empfehlen für elektronische Signaturen folgende Schlüssellängen:

Tabelle 1: Empfohlene Schlüssellängen (Stand 2012, siehe [3])

	Früher	Heute	Zukunft
RSA	1024 Bit	2048 Bit	4096 Bit
ECC	160 Bit	224 Bit	224 / 256 Bit

2.4 INSIKA nutzt ECC mit 192 Bit

Für die Auswahl des verwendeten Verfahrens für INSIKA waren verschiedenste Aspekte zu berücksichtigen:

Hohe Sicherheit: Für den Einsatz bei INSIKA kommen ausschließlich als sicher anerkannte kryptographische Verfahren in Frage. Da hier lediglich Signaturen verwendet werden, ist ein asymmetrisches Verfahren zu wählen.

Einfache Schlüsselverwaltung: Diese Forderung ist nur durch Verwendung einer PKI erfüllbar; diese wiederum ist nur mit asymmetrischen Verfahren sinnvoll einsetzbar.

Preiswerte Hardware: Für den Einsatz bei INSIKA sollte eine preiswerte und praktikable Lösung gefunden werden. Daher kam die Neuentwicklung eines „Fiskalspeichers“ nicht in Frage. Es gibt zwar sogenannte „Hardware Security Module“ (HSM) zur sicheren digitalen Signatur; diese kosten aber mehrere tausend Euro. Als preiswerte Alternative bietet sich eine Smart Card an.

Einfache Integration in Kassensysteme: Smart Cards lassen sich auch in aktuell verfügbare Kassensysteme mit geringem Aufwand integrieren. Im einfachsten Fall wird dazu lediglich eine serielle Schnittstelle benötigt.

Performance: Die für die Signatur einer Buchung (eines Beleges) verfügbare Zeit liegt bei deutlich unter einer halben Sekunde – sonst wird sie beim Kassiervorgang als störend empfunden. Diese Forderung kann lediglich ECC erfüllen. Bei der verwendeten Smart Card werden mit ECC 192 Bit Signaturzeiten von etwa 160 ms erreicht – bei RSA mit 2048 Bit benötigt eine Signatur etwa 2 Sekunden.

Länge der Signatur: Da die Signatur auf die Belege gedruckt werden soll und für eine Überprüfung

ggf. auch wieder eingetippt werden muss, sollte eine möglichst kurze Signatur verwendet werden.

Aufgrund der Abwägung zwischen Sicherheit und Druckbarkeit wird für INSIKA ECC mit 192 Bit Schlüssellänge verwendet [4]. Eine Änderung ist dabei leicht möglich (siehe 3.7).

3 INSIKA TIM

Die zentrale Instanz zur Absicherung der Umsatzdaten bildet eine Smart Card. Diese Karte – das Tax Identification Module (TIM) – erfüllt mehrere Aufgaben, die sich aber nicht getrennt voneinander realisieren lassen.

3.1 Funktionen des TIM

Hauptaufgabe des TIM ist die Plausibilisierung, Speicherung und Signatur jedes einzelnen Kassenumsatzes. Zusätzlich sorgt das TIM für eine eindeutige Identifikation jeder Buchung und des Steuerpflichtigen.

3.2 Plausibilisierung der Umsatzdaten

Für jede Buchung müssen dem TIM der Umsatz, der Umsatzsteuersatz und der Umsatzsteuerbetrag übergeben werden. Die Übergabe kann als Brutto- oder Nettoumsatz erfolgen.

Zur Plausibilisierung der Umsätze berechnet das TIM aus den übergebenen Daten (Umsatz und Umsatzsteuersatz) den Umsatzsteuerbetrag und bei Brutto-Buchungen zusätzlich den Nettoumsatz. Der errechnete Umsatzsteuerbetrag wird mit dem übergebenen Wert verglichen. Bei Abweichungen wird die Buchung als „ungültig“ abgewiesen. Der (ggf. berechnete) Nettoumsatz und die berechnete Umsatzsteuer werden anschließend zu den gespeicherten Umsatzdaten addiert.

Um die Kumulierung von Rundungsfehlern zu vermeiden, werden alle Währungsbeträge auf zehntausendstel Cent genau berechnet und gespeichert.

Das TIM ist in der Lage, Umsätze getrennt nach verschiedenen Umsatzsteuersätzen – auch mehrere in einer einzigen Buchung – zu verarbeiten. Zusätzlich können mit dem TIM Umsätze im Agenturgeschäft, über Lieferscheine oder Trainingsbuchungen verarbeitet werden.

3.3 Aufzeichnung der Umsatzdaten

Die Umsatzdaten werden auf dem TIM als Summen monatsweise aufgezeichnet. Die Aufzeichnung erfolgt getrennt für verschiedene Umsatzsteuersätze.

Bild 5 stellt den grundsätzlichen Aufbau des Umsatzspeichers auf dem TIM dar. Für jeden Monat werden dort die Umsätze – getrennt nach Umsatzsteuersätzen – gespeichert. Diese Speicher sind mit „Container 1“ bis „Container 6“ bezeichnet. Zusätzlich zu diesen werden getrennte Speicher für Agenturgeschäfte, Lieferschein-Umsätze und Trainingsbuchungen vorgehalten. Diese Umsätze werden vom TIM als „nicht umsatzsteuerrelevant“ behandelt. Die Umsatzsteuer wird in einer – nicht von der Kasse erstellten – Rechnung ausgewiesen.

3.4 Signatur der Umsatzdaten

Nach Plausibilisierung der Umsätze erstellt das TIM eine Signatur für diese Buchung. Dabei werden folgende Daten signiert:

- Datum und Uhrzeit
- ID des Steuerpflichtigen
- ID des Bedieners (z.B. Kellner)
- Buchungsdaten (der Hashwert über alle Positionen einer Buchung)
- Kennzeichen Brutto- / Nettoumsatz
- Kennzeichen Trainingsbuchung
- Eindeutige Sequenznummer
- Umsätze getrennt nach Umsatzsteuersätzen

Dabei ist die ID des Steuerpflichtigen fest auf dem TIM gespeichert. Die Sequenznummer wird für jede Buchung vom TIM selbst erzeugt. Alle anderen Daten werden dem TIM übergeben.

Über diesen Datensatz wird zunächst der Hashwert (Verfahren SHA-1) gebildet und dieser anschließend signiert. Die so erstellte Signatur wird im Kassenjournal gespeichert und auf dem Beleg ausgedruckt.

3.5 Sonstige Funktionen

3.5.1 Identifikation einer Buchung

Zur eindeutigen Identifikation jeder einzelnen Buchung führt das TIM eigenständig eine Sequenznummer. Diese wird bei jeder Signatur erhöht und – von außen unveränderlich – auf dem TIM gespeichert.

3.5.2 Identifikation des Steuerpflichtigen

Jedes TIM wird mit der Umsatzsteuer-Identifikationsnummer des Steuerpflichtigen personalisiert. Dieses Merkmal geht in jede mit dem TIM erstellte Signatur ein.

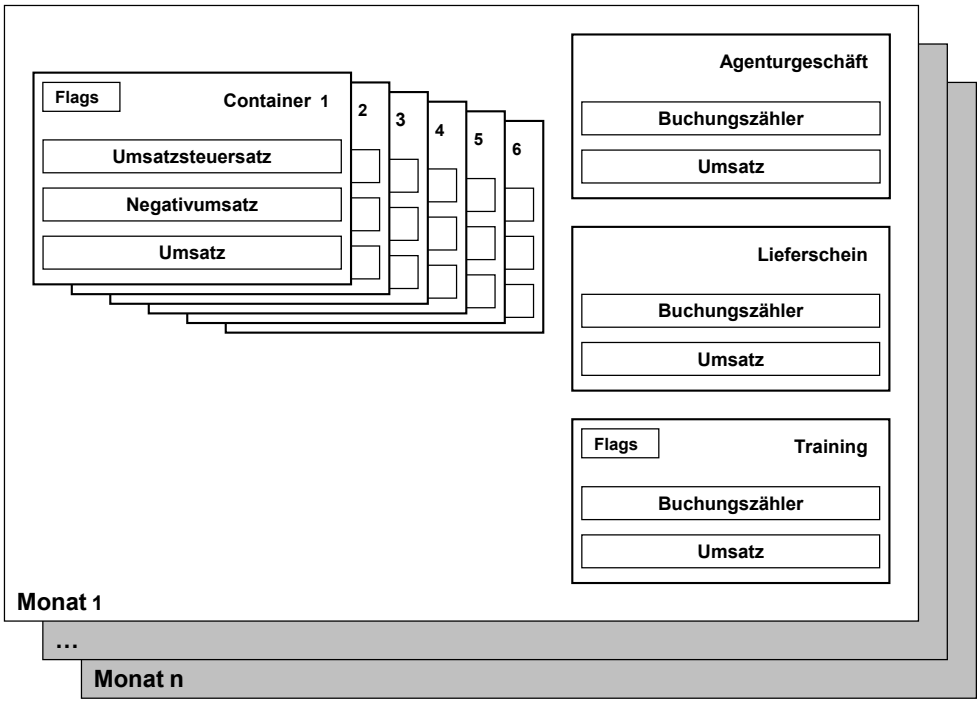

Abbildung 5: Umsatzspeicher des TIM

3.5.3 Reportfunktion

Über die Reportfunktion können Tagesabschlüsse realisiert werden. Jeder Tagesabschluss wird mit einer eindeutigen Sequenznummer versehen und signiert. Dadurch erfordert eine Revision deutlich weniger Aufwand. Weiterhin liefert diese Funktion auf Anfrage monatsgenaue Umsatzsummen.

3.5.4 Umgang mit verschiedenen Steuersätzen

Das TIM ist für die gleichzeitige Verwendung von 6 verschiedenen Umsatzsteuersätzen ausgelegt. Diese Sätze sind aber nicht fest konfiguriert, sondern werden dem TIM mit jeder Buchung von außen neu vorgegeben. Das TIM speichert Umsatzsteuersätze mit in dem zugehörigen Umsatz-Container. Werden in einem Monat verschiedene Steuersätze in einen Umsatz-Container gebucht, so zeichnet das TIM diese Umsatzsteueränderung auf. Bei Revision / Auswertung können (müssen) so gebuchte Speicher dann gesondert behandelt werden.

3.6 Sicherung gegen Manipulationen

Zur Sicherung gegen Manipulationen werden alle Daten auf der Smart Card als „Nur Lesbar“ gespeichert. Lediglich die im TIM realisierten Befehle können diese Daten ändern.

Das verwendete ECC Schlüsselpaar wird auf der Smart Card selbst erzeugt. Der private Schlüssel ist nicht lesbar auf der Smart Card gespeichert. Der öffentliche Schlüssel wird in einem Zertifikat auf dem TIM selbst gespeichert.

Jedes TIM hat eine eindeutige Seriennummer, die schon bei der Produktion des Chips festgelegt wird. Über diese ist ein TIM immer eindeutig identifizierbar.

3.7 Referenzimplementierung des TIM

Die Referenzimplementierung des TIM basiert auf einer Smart Card mit CardOS V4.3b Betriebssystem von Siemens. Die kryptographischen Algorithmen stellt das ECC-Package von cryptovision zur Verfügung.

Die Funktionen des TIM werden durch das INSIKA TIM-Package von cryptovision realisiert. Die Umstellung auf „längere“ ECC-Schlüssel oder das SHA-2 Hashverfahren ist jederzeit mit minimalem Aufwand möglich.

4 Fazit

Mit dem INSIKA Konzept und dem INSIKA TIM wird erstmals eine praxisgerecht einsetzbare Plattform für revisionssichere Kassensysteme geschaffen.

Der Einsatz des kryptographischen Verfahrens ECC bietet eine von anderen Systemen bisher nicht erreichte Sicherheit gegen Manipulationen an Kassenumsätzen.

Die Verwendung der TIM Smart Card ist nicht nur sicher, sondern darüber hinaus kostengünstig und auch für ältere Kassensysteme mit überschaubarem Aufwand zu realisieren.

Literatur

[1] NIST. *FIPS Publication 180-4: Secure Hash Standard (SHS)*. National Institute of Standards und Technology, März 2012. URL: http://csrc.nist.gov/publications/PubsFIPS.html.

[2] NIST. *FIPS Publication 186-2: Digital Signature Standard (DSS)*. National Institute of Standards und Technology, Jan. 2000. URL: http://csrc.nist.gov/publications/PubsFIPSArch.html.

[3] Damien Giry. *Keylength - Cryptographic Key Length Recommendation*. 2012. URL: http://www.keylength.com/ (besucht am 27. 09. 2012).

[4] NIST. *FIPS Publication 186-3: Digital Signature Standard (DSS)*. National Institute of Standards und Technology, Juni 2009. URL: http://csrc.nist.gov/publications/PubsFIPS.html.