Ein mißbrauchfreies anonymes elektronisches Zahlungssystem

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr. rer. nat.)

von

Dipl.-Inform. Dennis Kügler

aus Hamburg

Referenten:	Prof. Dr. Johannes Buchmann Prof. Dr. Albrecht Beutelspacher
Tag der Einreichung:	26. Juni 2002
Tag der mündlichen Prüfung:	13. August 2002

Darmstadt, 2002

D 17

Für Stefanie und Yannis

Danksagung

An dieser Stelle möchte ich mich zuerst bei meiner Frau Stefanie für ihre Liebe, Geduld und Unterstützung bedanken, die sie mir in der Zeit meiner Dissertation entgegengebracht hat.

Ganz besonderer Dank gebührt natürlich Herrn Prof. Johannes Buchmann für die hervorragende Betreuung dieser Arbeit und für die Gelegenheit, an seinem Lehrstuhl an einem interessanten und herausfordernden Thema zu arbeiten.

Ich bedanke mich ebenfalls bei Herrn Prof. Albrecht Beutelspacher, der sich die Zeit für die Zweitbegutachtung meiner Dissertation genommen hat.

Außerdem möchte ich hier Holger Vogt erwähnen sowie die unzählbaren Diskussionen, die wir in unserem gemeinsamen Büro geführt haben.

Schließlich möchte ich mich bei meinen Eltern dafür bedanken, daß sie meine Neugier geweckt und meine Interessen stets gefördert haben.

Diese Arbeit wurde von der Deutschen Forschungsgemeinschaft (DFG) im Rahmen des Graduiertenkollegs “Infrastruktur für den elektronischen Markt” finanziell unterstützt.

Zusammenfassung

Anonymes Bezahlen ist aus Gründen des Datenschutzes zwar unbedingt notwendig, kann aber auch für kriminelle Zwecke mißbraucht werden: Geschützt durch anonyme Zahlungen können Verbrechen begangen werden, die mit Bargeld nicht möglich wären. Da ein möglicher Mißbrauch von anonymen Zahlungen nicht toleriert werden kann, muß die gewährte Anonymität eingeschränkt werden.

Zahlungssysteme mit eingeschränkter Anonymität basieren in der Regel auf dem Konzept des Treuhänders, der die Anonymität jeder Zahlung nachträglich aufheben kann. Wurde die Anonymität mißbraucht, kann auf diese Weise der Täter immer ermittelt werden. Jedoch müssen sowohl die Kunden, als auch die Strafverfolgungsbehörden dem Treuhänder vertrauen, daß er genau dann die Anonymität einer Zahlung aufdeckt, wenn eine richterliche Anordnung vorliegt. Aus diesem Grund muß der Treuhänder strengen Auflagen und Kontrollen unterliegen. Es stellt sich die Frage, wer diese Kontrollen durchführen soll und wie überhaupt kontrolliert werden kann, ob ein Treuhänder seine Aufgabe korrekt erfüllt. Wir sind daher der Meinung, daß der treuhänderbasierte Ansatz für anonyme Zahlungssysteme nicht geeignet ist.

In dieser Dissertation beschreiben wir FlexiCash, ein neues anonymes elektronisches Zahlungssystem, das Mißbrauch von Anonymität auf neue und elegante Weise verhindert, ohne dabei eine zusätzliche dritte Partei zu benötigen. Obwohl FlexiCash sehr gut mit Bargeld vergleichbar ist, bietet FlexiCash sowohl stärkere Anonymität als auch wesentlich bessere Deanonymisierungsmechanismen. Diese Eigenschaften erreichen wir durch einen Kontrollmechanismus für Deanonymisierungen, der in keinem anderen Zahlungssystem existiert: In regelmäßigen Abständen können vorhergehende Zahlungen durch den Zahlenden selbst überprüft werden. Bei dieser Überprüfung kann der Zahlende feststellen, ob die Anonymität einer Zahlung in der Vergangenheit aufgehoben wurde. In diesem Fall muß eine richterliche Anordnung dafür vorhanden sein. Andernfalls wurde beweisbar eine illegale Deanonymisierung durchgeführt, die rechtlich verfolgt werden kann. Wird bei einer Überprüfung keine Deanonymisierung festgestellt, hat der Zahlende die Garantie, daß diese Zahlungen tatsächlich anonym durchgeführt wurden und anonym bleiben werden. Kein anderes Zahlungssystem mit eingeschränkter Anonymität kann diese starke Anonymität garantieren.

Abstract

While anonymous payments are necessary for privacy reasons, they can as well be misused for criminal activities: Anonymous electronic cash can be used to conduct crimes that would be impossible with physical cash. As potential misuse of anonymity cannot be tolerated, the allowed anonymity has to be restricted.

Payment systems with restricted anonymity are commonly based on the concept of a trusted third party that is always able to subsequently revoke the anonymity of any payment. If anonymity was misused, it is always possible to investigate the delinquent. However, both the customers as well as law enforcement have to rely on the trusted third party to revoke the anonymity of a payment if and only if a judical warrant was issued. Thus, the trusted third party must be subject to rigorous restrictions and controls. Creating an instance that is able to control the trusted third party is still an open problem. Therefore, in our opinion, the trusted third party approach is not suitable for anonymous payment systems.

In this dissertation we introduce FlexiCash, a new anonymous payment system that uses a novel approach to prevent misuse of anonymity independent of trusted third parties. Although FlexiCash is well comparable to physical cash, it does not only offer stronger anonymity but has also superior deanonymization mechanisms. This is possible by integrating an audit mechanism in FlexiCash that cannot be found in any other payment system: At regular intervals previous payments can be audited by the payer himself. Then the payer can check, whether the anonymity of his payments has been revoked. In this case, a judical warrant must be available otherwise the deanonymization was provably illegal and can be prosecuted. If no deanoymization is detected by the payer, it is guaranteed that the payments have been and will remain anonymous. No other payment system with restricted anonymity can guarantee such strong anonymity.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Einleitung

In der folgenden Arbeit stellen wir das neue anonyme elektronische Zahlungssystem FlexiCash vor. Anonymes Bezahlen ist aus Gründen des Datenschutzes unbedingt notwendig und wird z.B. im Teledienstedatenschutzgesetz [TDD] gefordert. Anonymität kann aber auch für kriminelle Zwecke mißbraucht werden. Geschützt durch anonyme Zahlungen können Verbrechen begangen werden, die mit Bargeld nicht möglich wären. Ein Beispiel für den Mißbrauch von Anonymität ist das "perfekte Verbrechen" [vSN92]. Unter dem perfekten Verbrechen versteht man eine Erpressung, in der die Geldübergabe über ein anonymes elektronisches Zahlungssystem erfolgt. Auf diese Weise hinterläßt der Täter keine verwertbaren Spuren. Andere Beispiele, die wir später betrachten werden, sind Geldwäsche und Bankraub.

FlexiCash löst das Problem des Mißbrauchs von Anonymität auf neue Weise. Wir erreichen folgende Eigenschaften:

- FlexiCash bietet dem Zahlenden eine mit Bargeld vergleichbare Anonymität.
- Bei Verdacht auf Mißbrauch können zukünftige Zahlungen deanonymisiert werden.
- Deanonymisierungen sind nur mit richterlicher Genehmigung möglich.
- Es werden keine Treuhänder zum Deanonymisieren benötigt.

Kein anderes anonymes Zahlungssystem besitzt alle diese Eigenschaften. Bisherige Zahlungssysteme bieten entweder vollständige Anonymität und ignorieren den möglichen Mißbrauch (z.B. [Cha83, CFN88, Bra93, Sch97]) oder schränken die Anonymität zu stark ein (z.B. [BGK95, JY96, CMS96, FTY96, DFTY97, CMS97, PP97, JY97, JM98, JM99, Poi01]).

Zahlungssysteme mit eingeschränkter Anonymität basieren in der Regel auf dem Konzept des Treuhänders, der die Anonymität jeder Zahlung nachträglich aufheben kann. Wurde die Anonymität mißbraucht, kann auf diese Weise der Täter immer ermittelt werden. Jedoch müssen sowohl die Kunden, als auch die Strafverfolgungsbehörden dem Treuhänder vertrauen, daß er genau dann die Anonymität einer Zahlung aufdeckt, wenn eine richterliche Anordnung vorliegt. Aus diesem Grund muß der Treuhänder strengen Auflagen und Kontrollen unterliegen. Es stellt sich die Frage, wer diese Kontrollen durchführen soll und wie überhaupt kontrolliert werden kann, ob ein Treuhänder seine Aufgabe korrekt erfüllt. Wir sind daher der Meinung, daß der treuhänderbasierte Ansatz für anonyme Zahlungssysteme nicht geeignet ist.

Bei FlexiCash werden Deanonymisierungen durch die Bank ohne Hilfe eines Treuhänders durchgeführt. Im Prinzip kann die Bank daher jede Zahlung ohne rechtliche Grundlage deanonymisieren. Wir nennen das eine illegale Deanonymisierung. Wir zeigen nun, warum die Bank keine illegale Deanonymisierungen durchführen wird. FlexiCash hat einen Kontrollmechanismus für Deanonymisierungen. Jede Deanonymisierung kann immer nach einer bestimmten, zuvor festgelegten Zeit durch die deanonymisierte Person selbst entdeckt werden. Entdeckte Deanonymisierungen können gegenüber jedem Dritten bewiesen werden. Stellt sich heraus, daß eine Deanonymisierung ohne rechtliche Grundlage durchgeführt wurde, kann die Bank dafür bestraft werden, z.B. mit Entzug der Banklizenz. Aber nicht nur aufgrund einer möglichen Strafe wird die Bank keine illegalen Deanonymisierungen durchführen. Da es der Bank nicht möglich ist, eine durchgeführte illegale Deanonymisierung abzustreiten, findet eine Selbstregulierung statt. Eine Bank, die nachweislich illegale Deanonymisierungen durchführt, wird mit Sicherheit einen Großteil ihrer Kunden verlieren, woran natürlich keine Bank interessiert ist.

In den folgenden vier Kapiteln beschreiben wir die Ideen, die Realisierung und eine Implementierung von FlexiCash. Wir fassen die Inhalte der einzelnen Kapitel hier kurz zusammen.

Kapitel 1

In diesem Kapitel stellen wir die Grundlagen von anonymen Zahlungssystemen vor. Zunächst beschreiben wir die Teilnehmer, die Vorgänge und die Eigenschaften eines anonymen Zahlungssystems. Anschließend stellen wir eine mögliche Realisierung auf Basis des allgemeinen kryptographischen Primitivs der *blinden Signatur* vor. Wir definieren die Eigenschaften und die Operationen von blinden Signaturen. Für jeden der möglichen Vorgänge spezifizieren wir die Protokolle zwischen den Teilnehmern auf Basis von allgemeinen blinden Signaturen. Wir zeigen für diese Realisierung, wie weit die von uns zuvor geforderten Eigenschaften erfüllt werden. Danach diskutieren wir die aus der vollständigen Anonymität des Zahlungssystems resultierenden Bedrohungen. Anhand von Erpressung, Geldwäsche und Bankraub demonstrieren wir, wie Anonymität für Verbrechen mißbraucht werden kann, so daß die Ermittlung des Täters unmöglich ist.

Anschließend beschreiben wir Zahlungssysteme mit eingeschränkter Anonymität. Bei dieser Art von anonymen Zahlungssystemen kann ein Treuhänder die Anonymität von Zahlungen gezielt aufheben. Wir zeigen, wie weit die Probleme Erpressung, Geldwäsche und Bankraub durch Deanonymisierungsmechanismen gelöst werden können. Wir stellen verschiedene aus der Literatur bekannte treuhänderbasierte Zahlungssysteme vor und kommen zu dem Schluß, daß derartige Zahlungssysteme keine geeignete Lösung für anonymitätsbezogene Probleme bieten. Schließlich stellen wir anonyme Zahlungsysteme vor, die nicht auf dem Paradigma der blinden Signatur basieren und zeigen, daß auch diese Zahlungssysteme das Problem des Mißbrauchs von Anonymität nicht zufriedenstellend lösen können.

Kapitel 2

Das Ziel dieses Kapitels ist es, die Grundideen unseres neuen Zahlungssystems vorzustellen. FlexiCash ist ein anonymes Zahlungssystem, das sich an den Vorgaben von Bargeld orientiert: Analog zu Geldscheinen, die mit einer “unsichtbaren” Farbe gekennzeichnet werden können, führen wir *markierbare* elektronische Münzen ein. Durch die markierbaren Münzen sind wir in der Lage, überprüfbare Deanonymisierungen anzubieten. Wir präsentieren eine Realisierung von FlexiCash, die wir in zwei Varianten unterteilen:

1. Die Basisvariante ist sehr anschaulich und dient als Grundlage für die folgende erweiterte Variante. In der Basisvariante werden Münzen von der Bank beim Abheben gegebenenfalls unsichtbar markiert. Auf diese Weise kann die Bank diese Münze später wiedererkennen. Das ist exakt das gleiche Prinzip, wie es bei Bargeld existiert.

2. Die erweiterte Variante bietet einen zusätzlichen Deanonymisierungsmechanismus, der weniger anschaulich und mit Bargeld nicht realisierbar ist. Dieser Deanonymisierungsmechanismus ist sehr effektiv bei der Aufklärung von Anonymitätsmißbrauch. Dennoch bietet die erweiterte Variante eine mit Bargeld vergleichbare Anonymität.

Für beide Varianten spezifizieren wir für jeden der möglichen Vorgänge die Protokolle zwischen den Teilnehmern auf Basis von markierbaren Münzen. Wir zeigen, daß die geforderten Eigenschaften erfüllt sind und wie die anonymitätsbezogenen Probleme Erpressungen, Geldwäsche und Bankraub mit FlexiCash gelöst werden können.

Kapitel 3

In diesem Kapitel beschreiben wir die Realisierung von markierbaren Münzen auf allgemeinen kryptographischen Primitiven. Dazu führen wir ein neues kryptographisches Primitiv ein, die *blinde Chiffre*. Wir definieren die Eigenschaften und die Operationen einer blinden Chiffre und zeigen, wie sich blinde Signaturen und blinde Chiffren zu markierbaren Münzen kombinieren lassen.

Wir geben dann eine konkrete Realisierung für blinde Signaturen und blinde Chiffren an. Wir stellen die blinde Schnorr Signatur und die blinde ElGamal Chiffre vor, welche auf dem allgemeinen diskreten Logarithmusproblem basieren. Wir beschreiben die Protokolle der beiden Primitive und zeigen, daß sich die Eigenschaften durch die Kombination nicht ändern. Auf dieser Basis realisieren wir die Protokolle der erweiterten Variante von FlexiCash noch einmal detailliert. Anschließend diskutieren wir kurz mögliche alternative Realisierungen für FlexiCash.

Kapitel 4

Im letzten Kapitel diskutieren wir die Ergebnisse einer experimentellen Implementierung von FlexiCash in Java. Die Implementierung basiert auf blinden Schnorr Signaturen und blinden ElGamal Chiffren. Wir beschreiben zunächst die Implementierung der beiden Primitive sowie die

ermittelten Laufzeiten und Ausgabegrößen für die Operationen. Anschließend verwenden wir diese Werte, um für jeden der möglichen Vorgänge von FlexiCash die zu erwartende Laufzeit und die Menge der zu kommunizierenden Daten zu berechnen. Die berechneten Werte vergleichen wir anschließend mit den tatsächlich gemessenen Werten unserer Implementierung. Zum Abschluß diskutieren wir die Effizienz und die Skalierbarkeit von FlexiCash und vergleichen den für die Bank notwendigen Aufwand von FlexiCash mit dem vollständig anonymen Zahlungssystem aus Kapitel 1.

Veröffentlichungen

Die Grundideen dieser Arbeit finden sich in folgenden Veröffentlichungen:

- Marking: A Privacy Protecting Approach Against Blackmailing [KV01a]
- Fair Tracing without Trustees [KV01b]
- Auditable Tracing with Unconditional Anonymity [KV01c]
- Enabling Privacy Protection in E-Commerce Applications [Küg01]
- Unsichtbare Markierungen in elektronischem Geld [KV01d]

Kapitel 1

Anonyme elektronische Zahlungssysteme

In diesem Kapitel beschreiben wir anonyme Zahlungssysteme, die auf dem Konzept von *elektronischen Münzen* oder kurz *Münzen* aufbauen. Eine elektronische Münze ist eine von der herausgebenden Bank digital signierte Seriennummer. Elektronische Münzen sind das digitale Äquivalent zu Bargeld.

Bargeld repräsentiert einen Wert an sich, d.h. durch die Weitergabe eines Geldscheins wird auch der Wert an den Empfänger transferiert. Jeder Geldschein ist durch eine eindeutige Seriennummer gekennzeichnet. Dadurch ist eine Identifizierung jedes Geldscheines möglich. Trotzdem ist Bargeld ein weitgehend anonymes Zahlungssystem: Banken könnten zwar theoretisch beim Abheben eines Geldscheines die Seriennummer notieren und überprüfen, wer diesen Geldschein später wieder einzahlt. Praktisch gesehen stellt diese Verfolgung jedoch nicht nur einen erheblichen Aufwand für die Banken dar, sondern das Verfolgen von Geldscheinen funktioniert auch nur sehr schlecht. Es ist nahezu unmöglich herauszufinden, ob bzw. welche weiteren Personen zwischen dem Abheben und Einzahlen diesen Geldschein besessen haben.

Anders verhält es sich, wenn große Mengen von Bargeld transferiert werden. In diesem Fall kann man davon ausgehen, daß sich bei einem Teil der Geldscheine nur sehr wenige Personen zwischen Abheben und Einzahlen befinden. Auf diese Weise kann man den Geldfluß teilweise rekonstruieren.

Gliederung des Kapitels

Wir beginnen in Abschnitt 1.1 mit der Beschreibung der Grundlagen anonymer, münzbasierter Zahlungssysteme. Anschließend stellen wir in Abschnitt 1.2 eine Realisierung eines Zahlungssystems mit vollständiger Anonymität vor. In Abschnitt 1.3 zeigen wir, daß diese Realisierung stärkere Anonymität bietet als Bargeld und beschreiben die daraus resultierenden Bedrohungen. Wir diskutieren dann in Abschnitt 1.4 verschiedene Ansätze, die versuchen, die aus der starken Anonymität resultierenden Bedrohungen zu lösen. Wir zeigen, warum diese Lösungen nicht geeignet sind. Zum Schluß stellen wir in Abschnitt 1.5 verschiedene Zahlungssysteme vor, die

nicht auf dem Paradigma der elektronischen Münzen basieren. Wir demonstrieren, daß auch diese Zahlungssysteme die anonymitätsbezogenen Probleme nicht lösen können.

1.1 Grundlagen anonymer Zahlungssysteme

In diesem Abschnitt beschreiben wir die Teilnehmer, die Vorgänge und die Eigenschaften von anonymen münzbasierten Zahlungssystemen. Anschließend betrachten wir die rechtlichen Grundlagen der durch ein anonymes Zahlungssystem geschlossenen Kaufverträge. Schließlich diskutieren wir die Voraussetzungen, die anonymes Bezahlen ermöglichen.

1.1.1 Teilnehmer

Ein münzbasiertes Zahlungssystem hat mindestens folgende Teilnehmer: Banken, Kunden, Händler, Verzeichnisse und Richter.

Banken: Banken sind Herausgeber von Münzen und verwalten Kunden- und Händlerkonten. Existieren mehrere Banken, müssen sich diese untereinander koordinieren: Jede Bank kann eine Überweisung auf ein von einer anderen Bank geführtes Konto ausführen.

Kunden: Kunden heben Münzen von ihrem Konto bei der Bank ab und bezahlen Händler mit den Münzen.

Händler: Händler liefern Waren oder Dienstleistungen gegen Münzen. Die Händler zahlen die von Kunden erhaltenen Münzen auf ihr Konto bei der Bank ein.

Richter: Richter entscheiden Konflikte zwischen den Teilnehmern. Die Entscheidung des Richters ist bindend. Es existieren immer mehrere voneinander unabhängige Richter. Wir gehen davon aus, daß jeder Richter korrekte Entscheidungen trifft.

Verzeichnisse: Verzeichnisse erstellen und speichern Zertifikate für die öffentlichen Schlüssel aller Teilnehmer. Existieren mehrere Verzeichnisse, muß garantiert sein, daß beim Zugriff auf ein Verzeichnis das geforderte Zertifikat des Teilnehmers verfügbar ist.

1.1.2 Vorgänge

Ein münzbasiertes Zahlungssystem muß mindestens die Vorgänge INITIALISIEREN, ABHEBEN, BEZAHLEN, EINZAHLEN und ZURÜCKGEBEN unterstützen. Die Vorgänge BEZAHLEN und EINZAHLEN sind zwei getrennte Vorgänge, sie werden aber oft zu einem Vorgang zusammengesetzt. Von einem *Online-Zahlungssystem* spricht man, wenn beide Vorgänge gleichzeitig

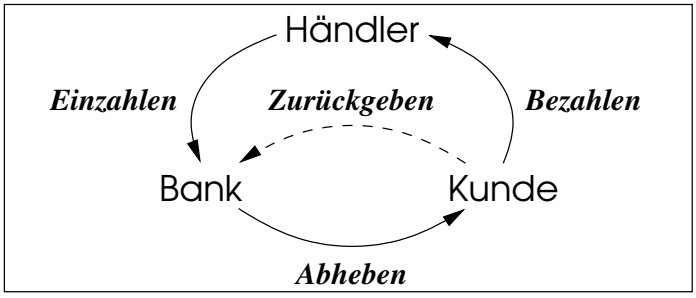

Abbildung 1.1: Zyklus einer Münze: Abheben, Bezahlen, Einzahlen

stattfinden müssen. Kann das Einzahlen nach dem Bezahlen erfolgen, spricht man von einem *Offline-Zahlungssystem*.

Abbildung 1.1 skizziert die Vorgänge ABHEBEN, BEZAHLEN, EINZAHLEN und ZURÜCKGEBEN.

Initialisieren: Die Bank muß das Zahlungssystem initialisieren, bevor andere Vorgänge stattfinden können. Bei der Initialisierung werden die für den weiteren Ablauf notwendigen Daten erzeugt und gegebenenfalls veröffentlicht.

Abheben: Der Kunde erhält Münzen von der Bank. Die Bank belastet das Konto des Kunden mit dem Betrag der abgehobenen Münzen und schreibt den Betrag einem Verrechnungskonto gut.

Bezahlen: Der Kunde gibt seine abgehobenen Münzen an einen Händler und erhält im Gegenzug die bezahlte Leistung vom Händler.

Einzahlen: Der Händler gibt die vom Kunden erhaltenen Münzen an die Bank. Dem Konto des Händlers wird der Betrag der eingezahlten Münzen gutgeschrieben, und das Verrechnungskonto wird mit dem Betrag belastet.

Zurückgeben: Der Kunde gibt die abgehobenen Münzen an die Bank zurück. Dem Konto des Kunden wird der Betrag der zurückgegebenen Münzen gutgeschrieben und das Verrechnungskonto wird mit dem Betrag belastet.

Das Verrechnungskonto gibt den Betrag der sich im Umlauf befindlichen gültigen Münzen an. Das Führen eines Verrechnungskontos ist aus Gründen der doppelten Buchführung notwendig. Das Konto muß sich immer im Haben befinden. Andernfalls wurden von der Bank gefälschte Münzen akzeptiert.

1.1.3 Eigenschaften

Für die Beschreibung der Eigenschaften eines münzbasierten elektronischen Zahlungssystems unterscheiden wir zwischen *gültigen* und *ungültigen Münzen*. Eine Münze ist genau dann gültig, wenn sie vom Kunden legal abgehoben und noch nicht benutzt wurde. In allen anderen Fällen bezeichnen wir eine Münze als ungültig. Ein anonymes münzbasiertes Zahlungssystem muß mindestens folgende Eigenschaften erfüllen:

- *Legal abgehobene Münzen sind immer gültig.*
- *Der Kunde ist beim Bezahlen gegenüber der Bank anonym.*
- *Der Händler muß die vom Kunden erhaltenen Münzen einzahlen.*
- *Zwischen Kunde und Händler kommt ein Kaufvertrag zustande.*
- *Gültige Münzen werden beim Einzahlen von der Bank immer akzeptiert.*
- *Gültige Münzen werden beim Zurückgeben von der Bank genau dann akzeptiert, wenn sie vom gleichen Kunden abgehoben wurden.*
- *Von der Bank einmal akzeptierte Münzen sind anschließend ungültig.*
- *Ungültige Münzen werden von der Bank nie akzeptiert.*

1.1.4 Kaufverträge

Von einem anonymen Zahlungssystem fordern wir, daß Kunde und Händler einen gültigen Kaufvertrag miteinander abschließen können. In diesem Abschnitt betrachten wir die Grundlagen eines Kaufvertrages. Der Kaufvertrag ist geregelt durch §433 BGB:

1. Durch den Kaufvertrag wird der Verkäufer einer Sache verpflichtet, dem Käufer die Sache zu übergeben und das Eigentum an der Sache zu verschaffen (§433 BGB Abs. 1 Satz 1).
2. Der Käufer ist verpflichtet, dem Verkäufer den vereinbarten Kaufpreis zu zahlen und die gekaufte Sache abzunehmen (§433 BGB Abs. 2).

Damit ein Kaufvertrag wirksam ist, müssen zwei übereintreffende Willenserklärungen abgegeben werden. Der Verkäufer bietet dem Käufer eine Sache zu einem bestimmten Preis an. Diese Willenserklärung wird als *Angebot* bezeichnet. Der Käufer kann das Angebot akzeptieren, was als *Annahme* bezeichnet wird.

Für die Abgabe der Willenserklärungen ist die Schriftform nicht notwendig. Liegen beide Willenserklärungen vor, kommt es zur Übergabe: Der Käufer übereignet dem Verkäufer das Geld

und der Verkäufer übereignet dem Käufer die Sache. Die Übergabe ist jeweils durch §929 BGB geregelt.

Der Verkauf von digitalen Waren oder Dienstleistungen gegen anonymes elektronisches Geld wird dadurch erschwert, daß der Austausch von Geld gegen Waren nicht als gleichzeitig stattfindender Vorgang gesehen werden kann:

- Sendet der Käufer dem Verkäufer zuerst das Geld, kann nicht garantiert werden, daß der Verkäufer auch tatsächlich die vereinbarte Sache liefert.
- Sendet der Verkäufer dem Käufer die Sache zuerst, kann nicht garantiert werden, daß der Käufer die gelieferte Sache auch bezahlt.

Dieses Problem wird auch das Problem des *fairen Austausches* genannt (s. z.B. [ASW97]). Daher ist es wichtig, daß der Kaufvertrag gegenüber einem Richter nachvollziehbar ist und die Lieferung der Waren gegebenenfalls eingeklagt werden kann.

Es ist deshalb notwendig, daß Angebot und Annahme nichtabstreitbare Willenserklärungen sind. Da der Kunde gegenüber dem Händler jedoch anonym sein soll, muß die Annahme ohne Erklärung erfolgen (§151 BGB). In diesem Fall akzeptiert der Kunde ein Angebot des Händlers, indem er den Händler bezahlt. Es muß nachvollziehbar sein, daß die Zahlung für ein bestimmtes Angebot erfolgt ist. Nur so hat der Kunde die Garantie, daß er die bezahlte Leistung auch erhalten wird.

1.1.5 Anonymität

Wir haben als Eigenschaft in Abschnitt 1.1.3 gefordert, daß die Anonymität des Kunden beim Bezahlen gegenüber Bank und Händler gegeben ist. Anonymität wird von Pfitzmann und Köhntopp [PK01] wie folgt definiert:

> *Anonymität ist der Zustand der Nichtidentifizierbarkeit innerhalb einer Menge von Subjekten, der Anonymitätsmenge.*

Damit ein Zahlungssystem überhaupt Anonymität bieten kann, müssen die Benutzer eine möglichst große Anonymitätsmenge bilden. Bei einer zu kleinen Anonymitätsmenge kann die Bank über die statistische Verteilung der Vorgänge ABHEBEN und EINZAHLEN Rückschlüsse auf den Kunden treffen: Existiert z.B. nur ein einziger Kunde der eine Münze mit bestimmten Wert abgehoben hat, kann der Kunde eindeutig identifiziert werden, wenn er eine Zahlung mit dieser Münze durchführt.

Es versteht sich von selbst, daß der Kunde möglichst keine personenbezogenen Daten an den Händler gibt. Vorausgesetzt, die vom Kunden angegebenen Daten sind korrekt, dann ist der Kunde natürlich nicht anonym. Ein mögliches Problem ist aber die unbemerkte Identifizierung des

Kunden. Eine derartige Identifizierung ist z.B. über eine dem Kunden fest zugewiesene Netzwerkadresse möglich. Die unbemerkte Identifizierung kann durch die Verwendung öffentlicher Endgeräte, durch Broadcastmechanismen, durch MIXe [Cha81] und eingeschränkt über dynamisch zugewiesene Netzwerkadressen vermieden werden. Grundsätzlich ist diese Art der Deanonymisierung von Kunden abstreitbar. Es kann nicht beweisen werden, daß es sich (aufgrund der festgestellten Netzwerkadresse) tatsächlich um einen bestimmten Kunden handelt.

Zusammenfassend müssen daher zwei zusätzliche Anforderungen erfüllt werden, andernfalls sind Zahlungen nicht anonym:

1. Eine große Anzahl von Kunden verhält sich jeweils annähernd gleich.
2. Der Kunde darf keine personenbezogenen Daten über sich preisgeben.

1.2 Ein vollständig anonymes Zahlungssystem

Im folgenden beschreiben wir eine Realisierung für ein vollständig anonymes Zahlungssystem. Das beschriebene Zahlungssystem basiert auf den Ideen von Chaum's blinder Signatur [Cha83]. Entgegen der ursprünglichen Version des Zahlungssystems, das ausführlich in [Sch97] beschrieben wird, verwendet die hier vorgestellte Realisierung bereits einige Erweiterungen. Anschließend prüfen wir, wie weit diese Realisierung die von uns für ein anonymes Zahlungssystem geforderten Eigenschaften erfüllt.

1.2.1 Blinde Signaturen

Eine blinde Signatur ist ein Protokoll zwischen zwei Teilnehmern, dem *Aussteller* und dem *Empfänger*. Das Ziel des Protokolls ist es, eine Signatur des Austellers zu einer vom Empfänger gewählten Nachricht zu erstellen, so daß folgende Eigenschaften erfüllt sind.

Eigenschaften

Wir fordern zwei Eigenschaften von einer blinden Signatur, damit wir sie als sicher bezeichnen: *Unfälschbarkeit* und *Blindheit*.

Unfälschbarkeit: Der Empfänger darf nicht mehr gültige Signaturen erstellen können, als er vom Aussteller erhalten hat.

Blindheit: Der Aussteller kann nichts aus dem Protokoll lernen, außer daß er eine gültige Signatur ausgestellt hat.

Formale Definitionen zur Sicherheit von blinden Signaturen finden sich in [JLO97]. Für die Unfälschbarkeit von blinden Signaturen wurde von Pointcheval und Stern [PS96] der Begriff *one-more-unforgeability* geprägt.

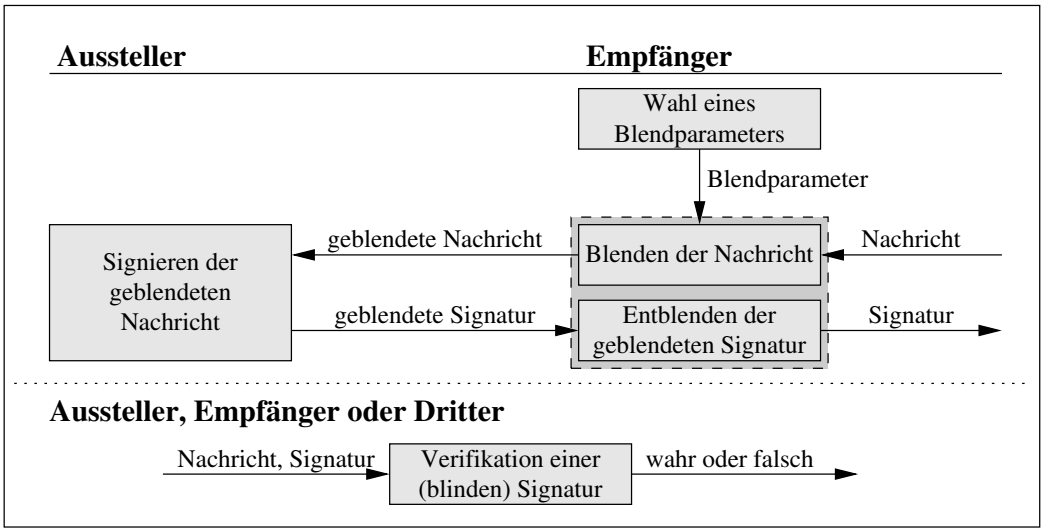

Abbildung 1.2: Ablauf der Operationen einer blinden Signatur

Operationen

Der Aussteller erstellt ein Signaturschlüsselpaar. Der private Signaturschlüssel wird vom Aussteller geheim gehalten, der öffentliche Signaturschlüssel wird publiziert und vom Empfänger verwendet.

Eine deterministische blinde Signatur soll die folgenden Operationen unterstützen. Der Ablauf der Operationen ist in Abbildung 1.2 dargestellt.

1. **Wahl eines Blendparameters.** Ein zufälliger Blendparameter wird gewählt. Die Ausgabe ist der Blendparameter.

2. **Blenden der Nachricht.** Die Eingabe ist die zu signierende Nachricht, der zu verwendende Blendparameter und der öffentliche Signaturschlüssel des Ausstellers. Aus der Nachricht und dem Blendparameter wird die geblendete Nachricht erzeugt. Die Ausgabe ist die geblendete Nachricht.

3. **Signieren der geblendeten Nachricht.** Die Eingabe ist die geblendete Nachricht und der private Signaturschlüssel des Ausstellers. Die geblendete Nachricht wird signiert. Die Ausgabe ist die geblendete Signatur.

4. **Entblenden der geblendeten Signatur.** Die Eingabe ist die geblendete Signatur und der Blendparameter, der zum Blenden der Nachricht verwendet wurde. Die geblendete Signatur wird zu einer Signatur der ursprünglichen Nachricht transformiert. Die Ausgabe ist eine reguläre Signatur.

5. **Verifikation einer (blinden) Signatur.** Die Eingabe ist die Nachricht, eine reguläre Signatur und der öffentliche Signaturschlüssel des Ausstellers. Die Signatur wird überprüft. Die Ausgabe ist `wahr`, wenn die Signatur gültig ist und `falsch` sonst.

1.2.2 Seriennummern der Münzen

In einem auf blinden Signaturen basierenden Zahlungssystem ist eine Münze eine von der Bank blind signierte Seriennummer. Der Wert einer Münze richtet sich nach dem verwendeten Signaturschlüssel der blinden Signatur. Für jeden möglichen Wert gibt es einen eigenen Signaturschlüssel.

Die Seriennummer wird vom Kunden zufällig und gleichverteilt gewählt. Die Seriennummer kann als zufälliger Bitstring mit fester Länge gewählt werden. Der Bitstring selbst hat keine Funktion. Allerdings muß die Wahrscheinlichkeit, daß zwei Kunden in einer Generation (s. nächster Abschnitt) den gleichen Bitstring verwenden, vernachlässigbar sein.

Alternativ kann die Seriennummer als digitales Pseudonym [Cha81] dienen. Für jede Münze wählt der Kunde ein neues, zufälliges Schlüsselpaar und verwendet den öffentlichen Schlüssel als Seriennummer. Der zugehörige private Schlüssel wird vom Kunden geheim gehalten. Wir nennen den öffentlichen Schlüssel den (öffentlichen) Münzschlüssel und den privaten Schlüssel den privaten Münzschlüssel der Münze. Die Seriennummer hat nun eine Funktion: Da der private Münzschlüssel nur dem Kunden bekannt ist, kann mit diesem Schlüssel der Verwendungszweck für die Münze festlegt werden. Auf diese Weise gibt der Kunde eine (signierte) Willenserklärung ab, deren Gültigkeit mit dem öffentlichen Münzschlüssel überprüft werden kann. Dadurch ist es dem Kunden möglich, das Angebot eines Händlers anzunehmen und so einen Kaufvertrag zu schließen (s. Abschnitt 1.1.4). Wir nennen diese mit dem privaten Münzschlüssel signierte Willenserklärung des Kunden die *Annahmeerklärung*.

1.2.3 Phasen einer Münzgeneration

Jede Münze gehört einer Generation an. Jede Generation hat eine Lebensdauer. Außerhalb ihrer Generation sind die Münzen ungültig. Wir teilen eine Generation in verschiedene Phasen ein, in denen die Vorgänge des Zahlungssystems (INITIALISIEREN, ABHEBEN, BEZAHLEN, EINZAHLEN und ZURÜCKGEBEN) jeweils erlaubt sind. Wir nennen die Phasen die *Initialisierungsphase*, die *Abhebephase*, die *Akzeptanzphase* und die *Rückgabephase*. Die Phasen einer Generation sind in Abbildung 1.3 dargestellt.

Initialisierungsphase

In der Initialisierungsphase ist der Vorgang INITIALISIEREN erlaubt. Die Bank erzeugt neue Signaturschlüssel. Das Verzeichnis erstellt und veröffentlicht die Zertifikate dieser öffentlichen Schlüssel. Mit dem Ende der Initialisierungsphase werden die neuen Schlüssel aktiviert. An die

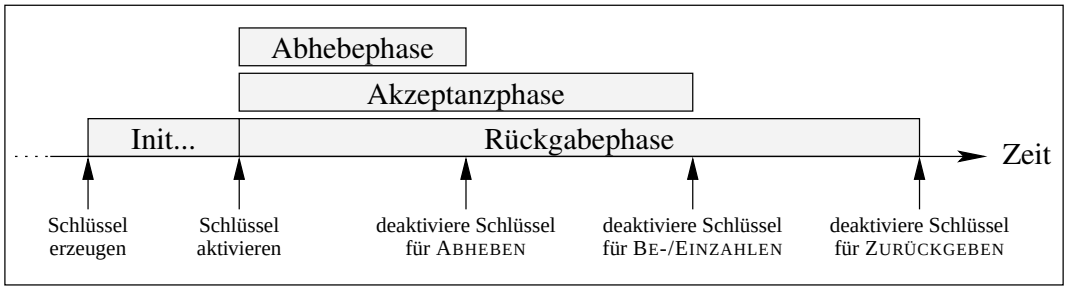

Abbildung 1.3: Phasen einer Münzgeneration

Initialisierungsphase schließen sich die Abhebephase, die Akzeptanzphase und die Rückgabephase an.

Abhebephase

In der Abhebephase ist der Vorgang ABHEBEN erlaubt. Bank und Kunde erzeugen gemeinsam neue Münzen unter Verwendung der Signaturschlüssel aus der zugehörigen Initialisierungsphase. Mit dem Ende der Abhebephase werden die Signaturschlüssel für Abhebevorgänge deaktiviert. Für einen reibungslosen Ablauf ist es notwendig, daß mit dem Ende jeder Abhebephase die Abhebephase der jeweils nächsten Generation beginnt.

Akzeptanzphase

In der Akzeptanzphase sind die Vorgänge BEZAHLEN und EINZAHLEN erlaubt. Die Bank akzeptiert die Münzen der zugehörigen Abhebephase. Mit dem Ende der Akzeptanzphase werden die Schlüssel für Zahlungen deaktiviert. Die Münzen sind jetzt für Zahlungen ungültig.

Rückgabephase

In der Rückgabephase ist der Vorgang ZURÜCKGEBEN erlaubt. Der Kunde kann seine unbenutzten Münzen der zugehörigen Abhebephase an die Bank zurückgegeben. Mit dem Ende der Rückgabephase werden die Schlüssel auch für Rückgaben deaktiviert. Die Münzen sind nun ungültig. Die Rückgabephase muß so lang sein, daß jedem Kunde die Möglichkeit gegeben wird, seine unbenutzten Münzen an die Bank zurückzugeben, bevor diese ungültig werden.

1.2.4 Realisierung der Teilnehmer

Im folgenden beschreiben wir die für die Banken, Kunden, Händler und Verzeichnisse notwendigen Vorbereitungen. Für den Richter sind keine besonderen Vorbereitungen notwendig.

Sitzungstabelle
Sitzungsnummer
Zeitpunkt Authentisierung
Identität Kunde/Händler

Einzahltabelle
Sitzungsnummer
Münze:
– Generation
– Seriennummer
– Münzsignatur

Annahmeerklärungen
Sitzungsnummer
Annahmeerklärung:
– Identität des Händlers
– Angebotsnummer

Abbildung 1.4: Datenstrukturen der Bank

Allen Teilnehmern ist gemein, daß sie zunächst Schlüsselpaare erzeugen müssen, die zur Authentisierung sowie zum Signieren verwendet werden. Die öffentlichen Schlüssel werden vom Verzeichnis zertifiziert und dort publiziert.

Banken

Die Bank benötigt eine Datenbank, in der die Kunden- und Kontodaten verwaltet werden. Für die Realisierung der Vorgänge werden weiterhin folgende Tabellen in der Datenbank angelegt: Eine Sitzungstabelle, eine Einzahltabelle und eine Annahmeerklärungstabelle. Diese Tabellen sind in Abbildung 1.4 dargestellt und werden im folgenden näher erläutert:

`Sitzungstabelle:` In der Sitzungstabelle werden nach einer erfolgreichen Authentisierung eines Kunden bzw. Händlers Informationen über die Authentisierung gespeichert. Es wird mindestens die Identität des Kunden bzw. Händlers und der Zeitpunkt der Authentisierung gespeichert. Beim Speichern wird eine eindeutige *Sitzungsnummer* von der Datenbank vergeben. Mit dieser Sitzungsnummer werden alle weiteren zu speichernden Daten dieser Sitzung verknüpft.

`Einzahltabelle:` In der Einzahltabelle werden die eingezahlten bzw. zurückgegebenen Münzen gespeichert. Zusätzlich wird zu jeder Münze die mit dem öffentlichen Münzschlüssel verifizierbare Münzsignatur abgespeichert.

`Annahmeerklärungstabelle:` In der Annahmeerklärungstabelle werden die Annahmeerklärungen gespeichert. Die Annahmeerklärungen werden unsigniert gespeichert, da die Münzsignaturen zu jeder eingezahlten Münze bereits in der Einzahltabelle gespeichert wurden.

Kunden

Der Kunde besitzt eine elektronische Geldbörse, mit der die von ihm abgehobenen Münzen verwaltet werden. Dazu führt die elektronische Geldbörse folgende Listen: eine Münzliste und eine

Münzliste
Münznummer
Münze:
– Generation
– Seriennummer
– Signatur der Bank
– privater Münzschlüssel
Zahlungsnummer

Zahlungsliste
Zahlungsnummer
Münznummern
Angebot des Händlers

Abbildung 1.5: Datenstrukturen des Kunden

Zahlungsliste. Diese Listen sind in Abbildung 1.5 dargestellt und werden im folgenden näher erläutert:

`Münzliste:` In der Münzliste werden die abgehobenen Münzen gespeichert. Nachdem der Liste eine neue Münze angefügt wurde, kann auf diese Münze über die eindeutige, fortlaufende Münznummer zugegriffen werden. Falls die Münze verwendet wurde, wird zu jeder Münze die Zahlungsnummer (s. Zahlungsliste) gespeichert.

`Zahlungsliste:` In der Zahlungsliste wird gespeichert, welche Münzen in welchem Zahlungsvorgang verwendet wurden. Für jeden Zahlungsvorgang wird eine neue, fortlaufende Zahlungsnummer vergeben. Unter dieser eindeutigen Nummer werden die Münznummern (s. Münzliste) der für die Zahlung verwendeten Münzen und das Angebot des Händlers gespeichert.

Händler

Der Händler verwendet eine Datenbank, mit der die angebotenen Produkte verwaltet werden. Für jede Bestellung wird eine neue, eindeutige Bestellnummer vergeben. Unter der Bestellnummer werden die bestellten Produkte sowie der Status der Bestellung (z.B. Angebot verschickt, Münzen erhalten, Betrag gutgeschrieben, Produkte geliefert) gespeichert.

Da diese Tabellen für die Realisierung der Vorgänge nur wenig relevant sind, verzichten wir hier auf eine detailliertere Beschreibung der Tabellen.

Verzeichnisse

Das Verzeichnis verwendet eine Datenbank, in der die vom Verzeichnis selbst erstellten Zertifikate gespeichert werden. Die Zertifikate der vom Verzeichnis erstellten Signaturschlüssel haben folgende Struktur:

1. Zu signierende Daten:
   (a) Die Identität des Verzeichnisses.
   (b) Die Identität der Bank bzw. des Kunden.
   (c) Der öffentliche Schlüssel.
   (d) Die Gültigkeit des Schlüssels:
      i. Zeitpunkt, ab dem der Schlüssel gültig ist.
      ii. Zeitpunkt, ab dem der Schlüssel nicht mehr gültig ist.
   (e) Die Nutzbarkeit des Schlüssels:
      i. Die Kennzeichnung für digitale Signatur.
2. Die Signatur des Verzeichnisses über die zu signierenden Daten.

Ist ein Zertifikat abgelaufen, muß es durch ein neues Zertifikat ersetzt werden. Wie die Teilnehmer dabei mit dem Verzeichnis interagieren wird nicht weiter betrachtet.

1.2.5 Realisierung der Vorgänge

Wir beschreiben nun die Realisierung der Vorgänge INITIALISIEREN, ABHEBEN, BEZAHLEN, EINZAHLEN und ZURÜCKGEBEN.

Initialisieren

Bei der Initialisierung jeder Münzgeneration wählt die Bank neue Signaturschlüssel für die einzelnen Münzwerte. Das Verzeichnis (s. Abschnitt 1.1.1) erstellt Zertifikate für die öffentlichen Signaturschlüssel und veröffentlicht diese. Die privaten Signaturschlüssel werden von der Bank geheim gehalten. Die Zertifikate für die öffentlichen Signaturschlüssel haben folgende Struktur:

1. Zu signierende Daten:
   (a) Die Identität des Verzeichnisses.
   (b) Die Identität der Bank.
   (c) Der öffentliche Schlüssel.
   (d) Die Gültigkeit des Schlüssels:
      i. Der Zeitpunkt, ab dem der Schlüssel gültig ist.
      ii. Der Zeitpunkt, ab dem der Schlüssel für das ABHEBEN ungültig ist.
      iii. Der Zeitpunkt, ab dem der Schlüssel für das BE-/EINZAHLEN ungültig ist.

iv. Der Zeitpunkt, ab dem der Schlüssel für das ZURÜCKGEBEN ungültig ist.

(e) Die Nutzbarkeit des Schlüssels:

i. Die Kennzeichnung für blinde Signatur.

ii. Der Wert der Münze.

2. Die Signatur des Verzeichnisses über die zu signierenden Daten.

Aus Gründen der Übersichtlichkeit verwenden wir in unserer Realisierung nur einen einzigen Signaturschlüssel bzw. Münzwert.

Abheben

Bevor ein Kunde neue Münzen von seinem Konto abheben kann, muß sich der Kunde gegenüber der Bank authentisieren. Nach der erfolgreichen Authentisierung des Kunden speichert die Bank den Zeitpunkt der Authentisierung sowie die Identität des Kunden in der Sitzungstabelle ab und erhält eine eindeutige Sitzungsnummer von der Datenbank. Anschließend interagieren Kunde und Bank in folgendem Abhebeprotokoll.

Protokoll 1. Abhebeprotokoll für vollständig anonyme Münzen.

1. Der Kunde wählt für jede abzuhebende Münze ein neues, zufälliges Schlüsselpaar und verwendet jeweils den öffentlichen Schlüssel als Seriennummer (Münzschlüssel) der Münze.
2. Auf jede Seriennummer erhält der Kunde eine blinde Signatur von der Bank. Der Kunde überprüft die blinden Signaturen und speichert die Münzen in der Münzliste ab.

   Ist eine Münze nicht korrekt signiert, kann diese später nicht zum Bezahlen oder Zurückgeben verwendet werden!
3. Die Bank führt die Buchung “Kundenkonto an Verrechnungskonto” durch.

Bezahlen und Einzahlen

Bevor eine Zahlung durchgeführt wird, sendet der Händler ein Angebot an den Kunden. Das Angebot hat folgende Struktur:

1. Zu signierende Daten:

   (a) Die Bestellnummer.

   (b) Die Liste der bestellten Waren und Preise.
2. Die Signatur des Händlers über die zu signierenden Daten.
3. Das Zertifikat des Händlers.

Der Kunde überprüft, ob das Angebot mit den von ihm bestellten Waren übereinstimmt und ob die Signatur korrekt ist. Nimmt der Kunde das Angebot an, interagieren Kunde, Händler

und Bank im folgenden Zahlungsprotokoll. Bevor Bank und Händler die Einzahlung starten, authentisiert sich der Händler gegenüber der Bank. Nach der erfolgreichen Authentisierung des Händlers speichert die Bank den Zeitpunkt der Authentisierung sowie die Identität des Händlers in der Sitzungstabelle ab und erhält eine eindeutige Sitzungsnummer von der Datenbank.

Protokoll 2. Zahlungsprotokoll mit vollständiger Anonymität des Kunden.

1. Der Kunde speichert das Angebot in der Zahlungsliste ab und wählt die Münzen aus, die für die Zahlung verwendet werden sollen. Der Kunde erstellt aus dem Angebot und den Münzen eine *Annahmeerklärung* und sendet sie an den Händler. Die Annahmeerklärung hat folgende Struktur:

   (a) Zu signierende Daten:

      i. Die Identität des Händlers.
      ii. Die Bestellnummer aus dem Angebot.

   (b) Die Liste der mit den privaten Münzschlüsseln erstellten Signaturen.

   (c) Die Liste der verwendenden Münzen.

2. Der Händler überprüft die Annahmeerklärung:

   (a) Die Identität des Händlers muß korrekt sein.

   (b) Die Bestellnummer muß beim Händler vorhanden sein.

   (c) Der Betrag der Münzen muß dem zu zahlenden Betrag entsprechen.

   Schlägt einer der Tests fehl, bricht der Händler die Zahlung ab.

   Der Händler startet die Einzahlung und sendet die Annahmeerklärung an die Bank.

3. Die Bank überprüft die Einzahlung:

   - Die Bank überprüft die Annahmeerklärung:
     – Die Identität des Händlers muß mit der Identität des Händlers in der Annahmeerklärung übereinstimmen.
     – Die Münzsignaturen der Annahmeerklärung müssen mit den öffentlichen Münzschlüsseln verifizierbar sein.
   - Die Bank überprüft die Münzen:
     – Der Einzahlzeitpunkt muß jeweils in der Akzeptanzphase der verwendeten Münzen liegen.
     – Die Seriennummern der Münzen dürfen nicht in der Einzahltabelle enthalten sein.
     – Die Signaturen der Münzen müssen gültig sein.

Schlägt einer dieser Tests fehl, wird die Einzahlung abgelehnt.

Die Bank speichert die Münzen und die Münzsignaturen in der Einzahltabelle sowie die Annahmeerklärung in der Annahmeerklärungstabelle jeweils unter der Sitzungsnummer ab. Die Münzen sind jetzt ungültig.

Anschließend führt die Bank die Buchung "Verrechnungskonto an Händlerkonto" durch und informiert den Händler über die erfolgreiche Einzahlung.

4. Der Händler liefert die Waren an den Kunden.

Zurückgeben

Bevor ein Kunde unbenutzte Münzen auf sein Konto zurückgeben kann, muß sich der Kunde gegenüber der Bank authentisieren. Nach der erfolgreichen Authentisierung des Kunden speichert die Bank den Zeitpunkt der Authentisierung sowie die Identität des Kunden in der Sitzungstabelle ab und erhält eine eindeutige Sitzungsnummer von der Datenbank. Anschließend interagieren Kunde und Bank in folgendem Rückgabeprotokoll.

Protokoll 3. Rückgabeprotokoll.

1. Der Kunde wählt die Münzen aus, die für die Rückgabe verwendet werden sollen und erstellt für jede Münze eine Rückgabesignatur, indem er die Seriennummer mit dem privaten Münzschlüssel signiert. Wir nennen diese Signaturen die Rückgabesignaturen. Der Kunde sendet die Münzen und die Rückgabesignaturen an die Bank.

2. Die Bank überprüft die Rückgabe:

   (a) Der Kunde darf maximal soviele Münzen zurückgeben, wie er zuvor abgehoben hat.

   (b) Die Bank überprüft die Münzen:

   - Der Rückgabezeitpunkt muß jeweils in der Rückgabephase der verwendeten Münzen liegen.
   - Die Münzen dürfen nicht in der Einzahltabelle enthalten sein.
   - Die Signaturen der Münzen müssen gültig sein.
   - Die Rückgabesignaturen müssen gültig sein.

   Schlägt einer dieser Tests fehl, wird die Rückgabe abgelehnt.

   Die Bank speichert die Münzen und die Rückgabesignaturen in der Einzahltabelle unter der Sitzungsnummer ab. Die Münzen sind jetzt ungültig.

   Anschließend führt die Bank die Buchung "Verrechnungskonto an Kundenkonto" durch und informiert den Kunden über die erfolgreiche Rückgabe.

1.2.6 Überprüfung der Eigenschaften

Wir überprüfen die in Abschnitt 1.1.3 geforderten Eigenschaften.

Legal abgehobene Münzen sind immer gültig: Der Kunde bemerkt ungültig signierte Münzen beim Verifizieren der blinden Signaturen. Die Münzen können weder zum Bezahlen benutzt noch zurückgegeben werden. Wir werden in Abschnitt 2.2.1 eine allgemeine Lösung für dieses Problem vorstellen. Diese Lösung ist auf jedes münzbasierte Zahlungssystem übertragbar und erlaubt, ungültig signierte Münzen an die Bank zurückzugeben.

Der Kunde ist beim Bezahlen gegenüber der Bank anonym: Um einen Kunden beim Bezahlen zu identifizieren, muß die Bank aus den vom Händler eingezahlten Münzen auf die abgehobenen Münzen (blinde Münzen) eines Kunden schließen können (s. Abschnitt 1.1.5). Die Verknüpfung zwischen einer eingezahlten und einer abgehobenen Münze ist aufgrund der *Blindheit* der blinden Signatur nicht möglich (s. Abschnitt 1.2.1).

Der Händler muß die vom Kunden erhaltenen Münzen einzahlen: Der Händler kann die vom Kunden erhaltenen Münzen weder zum Bezahlen verwenden, noch als seine eigenen Münzen an die Bank zurückgeben:

- Zum Bezahlen (eines anderen Händlers) muß der Händler eine neue Annahmeerklärung mit den vom Kunden erhaltenen Münzen signieren.
- Zum Zurückgeben muß der Händler für jede Münze eine Rückgabesignatur erstellen.

In beiden Fällen benötigt der Händler die privaten Münzschlüssel, die er jedoch nicht vom Kunden erhalten hat. Er kann die Münzen also nur zum Einzahlen verwenden. Erhält der Händler den privaten Münzschlüssel vom Kunden, handelt es sich um Geldwäsche. Geldwäsche werden wir in Abschnitt 1.3.2 betrachten.

Zwischen Kunde und Händler kommt ein Kaufvertrag zustande: Der Händler hat dem Kunden ein Angebot unterbreitet. Durch die Annahmeerklärung des Kunden wird das Angebot in dem Moment angenommen, in dem die Bank die Münzen akzeptiert hat (s. Abschnitt 1.1.4). Der daraus resultierende Kaufvertrag kann von einem Richter überprüft werden:

- Der Kunde präsentiert das Angebot des Händlers. Da das Angebot vom Händler signiert wurde, kann das Angebot nicht abgestritten werden.
- Die Bank präsentiert die Annahmeerklärung und die Münzen des Kunden. Da die Annahmeerklärung des Kunden mit den privaten Münzschlüsseln signiert wurde, kann die Annahme des Angebots nicht abgestritten werden.

Da die Bank den Betrag der Münzen dem Konto des Händlers gutgeschrieben hat, muß der Händler die im Angebot vereinbarten Leistungen erfüllen.

Gültige Münzen werden beim Einzahlen von der Bank immer akzeptiert: Bestreitet die Bank die Gültigkeit einer eingezahlten Münze, kann die Zahlung wiederholt werden. An der wiederholten Zahlung kann ein Richter an der Stelle des Händlers teilnehmen. Die Bank darf eine eingezahlte Münze aus folgenden Gründen ablehnen:

- Die Münze ist ungültig signiert.
- Die Annahmeerklärung ist ungültig signiert.
- Die Akzeptanzphase der Münzgeneration ist abgelaufen.
- Die Münze wurde bereits verwendet und die Bank muß eine gültige Annahmeerklärung bzw. Rückgabesignatur für diese Münze präsentieren können.

In allen Fällen kann der Richter leicht überprüfen, ob eine Münze zurecht von der Bank abgelehnt wurden.

Gültige Münzen werden beim Zurückgeben von der Bank genau dann akzeptiert, wenn sie vom gleichen Kunden abgehoben wurden: Bestreitet die Bank die Gültigkeit einer zurückgegebenen Münze, muß ein Richter die Rückgabe im Auftrag des Kunden durchführen. Die Bank kann eine zurückgegebene Münze aus folgenden Gründen ablehnen:

- Die Münze ist ungültig signiert.
- Die Rückgabesignatur ist ungültig.
- Die Rückgabephase der Münzgeneration ist abgelaufen.
- Die Münze wurde bereits verwendet und die Bank muß eine gültige Annahmeerklärung bzw. Rückgabesignatur für diese Münze präsentieren können.
- Der Kunde hat mehr Münzen zurückgegeben, als er abgehoben hat.

Mit Ausnahme des letzten Falls kann der Richter in allen Fällen leicht überprüfen, ob die Münze zurecht von der Bank abgelehnt wurden. Versucht der Kunde mehr Münzen zurückzugeben, als er abgehoben hat, muß die Bank anhand der abgehobenen und zurückgegebenen Münzen den Richter davon überzeugen, daß der Kunden zuviele Münzen zurückgeben will.

Es ist aber prinzipiell möglich, daß ein Kunde “fremde” Münzen eines anderen Kunden zurückgibt. In diesem Fall muß der ursprüngliche Besitzer den privaten Münzschlüssel weitergegeben haben. Damit handelt es sich um Geldwäsche. Geldwäsche werden wir in Abschnitt 1.3.2 betrachten.

Von der Bank einmal akzeptierte Münzen sind anschließend ungültig: Nachdem die Bank eine gültige Münze akzeptiert hat, wird die Seriennummer der Münze in die Einzahltabelle eingetragen und ist damit ungültig. Die Bank kann beweisen, daß sie eine Münze bereits akzeptiert hat, indem sie eine Annahmeerklärung bzw. Rückgabesignatur präsentiert, die mit dem öffentlichen Münzschlüssel verifiziert werden kann.

Ungültige Münzen werden von der Bank nie akzeptiert: Eine formal korrekte Münze ist in folgenden Fällen ungültig:

- Die Münze wurde bereits benutzt. In diesem Fall ist die Seriennummer der Münze bereits in der Einzahltabelle vorhanden und die Bank wird diese Münze nicht akzeptieren.
- Die Münze wurde nicht legal abgehoben:
  - Die Münze wurde gefälscht. Um ein Münze zu fälschen, muß die Signatur der Seriennummer gefälscht werden. Das Fälschen einer (blinden) Signatur ist aufgrund der *Unfälschbarkeit* der blinden Signatur nicht möglich (s. Abschnitt 1.2.1).
  - Im folgenden werden wir zwei weitere Fälle für illegal abgehobene Münzen beschreiben: Münzen, die aus einer Erpressung (s. Abschnitt 1.3.1) oder aus einem Bankraub (s. Abschnitt 1.3.3) resultieren.

Es ist der Bank nicht möglich, legal abgehobene Münzen von illegal abgehobenen Münzen zu unterscheiden. Daher muß die Bank jede Münze akzeptieren, die korrekt signiert ist, während der Akzeptanzphase eingezahlt bzw. während der Rückgabephase zurückgegeben wird und noch nicht in der Einzahltabelle vorhanden ist.

1.3 Anonymitätsbezogene Probleme

Im Vergleich zu Bargeld ist es mit vollständig anonymen Zahlungssystemen möglich, große Geldmengen schnell, anonym und unbeobachtbar zu transferieren. Die daraus resultierenden Bedrohungen werden wir in diesem Abschnitt betrachten.

1.3.1 Erpressung

Für einen Erpresser ist die Übergabe des physischen Lösegeldes gewöhnlich der gefährlichste Teil einer Erpressung. In der Regel gelingt es dabei, den Erpresser festzunehmen oder zu identifizieren. Ein anonymes elektronisches Zahlungssystem kann für die Lösegeldübergabe bei einer Erpressung wie folgt verwendet werden:

> Ein Erpresser entführt ein Baby und fordert von den Eltern ein Lösegeld in Form von anonymen Münzen. Der Erpresser wählt Seriennummern, blendet sie und sendet diese auf anonymen Weg (z.B. per Briefpost) an die Eltern, welche sie an die Bank weiterreichen. Die Bank bucht den entsprechenden Betrag vom Konto der Eltern ab, signiert die geblendeten Seriennummern und sendet sie an die Eltern zurück. Die geblendeten Signaturen werden nun in einem Broadcast-Medium publiziert (z.B. in einer Zeitung, per Videotext oder über das Usenet), so daß der Erpresser sie lesen kann, ohne Spuren auf seine Identität zu hinterlassen. Für andere Personen sind die Signaturen wertlos, da nur der Erpresser in der Lage ist, sie zu entblenden.

Da die Lösegeldübergabe kontaktlos erfolgt und die erpreßten Münzen unerkannt ausgegeben werden können, hinterläßt der Erpresser für die Strafverfolgungsbehörden keine verwertbaren Spuren. Aus diesem Grund wird diese Form der Erpressung das *perfekte Verbrechen* [vSN92] genannt.

1.3.2 Geldwäsche

Eine weitere Bedrohung, die direkt aus der starken Anonymität abgeleitet wird, ist die Erleichterung von Geldwäsche. Bei anonymen Zahlungssystemen wird nur dem Kunden Anonymität gewährt, der Händler ist der Bank grundsätzlich bekannt. Geldwäsche ist nur möglich, wenn Kunde und Händler kooperieren. Dazu transferiert der Kunde die Münzen entweder zusammen mit dem privaten Münzschlüssel an den Händler, oder er führt eine andere Transaktion im Auftrag des Händlers durch.

Folgendes Szenario veranschaulicht, wie Geldwäsche möglich ist:

> Ein Dealer verkauft Drogen gegen elektronische Münzen. Gleichzeitig bietet der Dealer eine legale elektronische Dienstleistung an, z.B. könnte der Dealer digitale Bilder verkaufen. Die aus dem Drogenverkauf erhaltenen Münzen benutzt der Dealer, um seine eigene Dienstleistung anonym zu kaufen. Das Geld wird somit auf das Konto des Dealers gutgeschrieben. Es ist nicht nachweisbar, daß das Geld eigentlich aus dem Verkauf von Drogen resultiert.

Geldwäsche ist notwendig, um die illegale Herkunft des Geldes zu verschleiern. Das läßt sich über den Verkauf von digitalen Daten sehr leicht durchführen:

- Die Herkunft des Geldes läßt sich aufgrund der Anonymität des Zahlenden nicht feststellen.
- Das Bereitstellen von digitalen Waren ist mit geringem Aufwand möglich, da sie leicht reproduziert werden können.
- Selbst der Kunde ist nicht in der Lage, die Geldwäsche nachzuvollziehen, wenn die Münzen blind abgehoben wurden (s. Erpressungszenario).

1.3.3 Bankraub

Die Fähigkeit Banknoten zu fälschen, ist selbstverständlich eine starke Bedrohung: Eine große Menge von gefälschtem Geld würde das Finanzsystem eines Staates zusammenbrechen lassen. Bei der Verwendung von elektronischen Münzen ist diese Bedrohung noch wesentlich stärker, da gefälschte Münzen nicht von regulär ausgestellten Münzen unterschieden werden können. Dieses grundsätzliche Problem wurde zuerst von Jakobsson und Yung [JY96] beschrieben und *Bankraub* genannt. Man unterscheidet zwei Arten von Bankraub:

- Ein privater Signaturschlüssel der Bank wird erpreßt (oder gestohlen), so daß der Erpresser beliebig viele Münzen selbst herstellen kann.
- Analog zu der Erpressung eines Kunden wird die Bank erpreßt, eine bestimmte Anzahl an Münzen herauszugeben.

Wir erweitern den Begriff des Bankraub hier um das Szenario des kryptographischen Zusammenbruchs. Wird das zur Münzerstellung verwendete Kryptosystem unsicher, kann prinzipiell jeder Münzen selbst erstellen.

1.4 Zahlungssysteme mit eingeschränkter Anonymität

Ein anonymes Zahlungssystem muß die anonymitätsbezogenen Probleme Erpressung, Geldwäsche und Bankraub lösen können. Sander und Ta-Shma [STS99b] behaupten, daß ein Zahlungssystem mit folgenden zwei Eigenschaften unattraktiv für kriminellen Mißbrauch ist:

- Münzen dürfen nicht zwischen Kunden transferierbar sein.
- Der von einem Kunden monatlich abhebbare Betrag muß begrenzt sein.

Durch diese Eigenschaften soll verhindert werden, daß große Geldmengen anonym akkumuliert werden können. Wir meinen, daß diese Forderungen nicht ausreichend sind. Erpressungen, Geldwäsche und Bankraub sind weiterhin möglich:

- Bei einer Erpressung wird der Erpresser fordern, die Betragsbegrenzung zu umgehen.
- Die Transferierbarkeit von Münzen ist für Geldwäsche nicht unbedingt notwendig. Anstelle dessen kann der Kunde die Transaktionen im Auftrag des Händlers durchführen.
- Bankraub kann auf diese Weise grundsätzlich nicht verhindert werden, da die Bank unter der Kontrolle des Bankräubers ist.

Wir meinen, daß ein *Deanonymisierungsmechanismus* unumgänglich ist. Mit einem Deanonymisierungsmechanismus kann die Anonymität von Zahlungen gezielt aufgehoben werden. Wir nennen Zahlungssysteme, die Deanonymisierungsmechanismen bieten, Zahlungssysteme mit eingeschränkter Anonymität.

Im folgenden betrachten wir Zahlungssysteme mit eingeschränkter Anonymität, in denen die Deanonymisierungsmechanismen über einen bzw. mehrere *Treuhänder* (Trusted Third Parties oder auch Trustees) realisiert werden.

1.4.1 Deanonymisierungsmechanismen

Stadler, Piveteau und Camenisch [SPC95] führen das Konzept der *fairen blinden Signatur* ein. Zahlungssysteme auf Basis von fairen blinden Signaturen haben zwei Deanonymisierungsmechanismen, um die Herkunft und den Empfänger von elektronischen Münzen zu bestimmen. Wir nennen diese Mechanismen *Münzverfolgung* (coin tracing) und *Kundenverfolgung* (owner tracing):

Münzverfolgung: Die abgehobenen Münzen eines verdächtigen oder erpreßten Kunden werden deanonymisiert, so daß die Bank diese Münzen beim Einzahlen erkennen kann.

Kundenverfolgung: Die von einem verdächtigen Händler eingezahlten Münzen werden deanonymisiert, so daß die Bank die Kunden ermitteln kann, die die Münzen abgehoben haben.

Ein wesentlicher Unterschied zwischen Münz- und Kundenverfolgung ist die Anzahl der möglicherweise betroffenen Kunden. Münzverfolgung richtet sich gegen einen Kunden. Kundenverfolgung richtet sich gegen einen Händler. Dadurch werden aber alle Kunden dieses Händlers deanonymisiert. Kundenverfolgung ist ein wesentlich mächtigerer Deanonymisierungsmechanismus als Münzverfolgung, aber auch ungleich gefährlicher in der Anwendung, wie folgendes Beispiel demonstriert:

> Angenommen, ein Händler verkauft legale Waren oder Dienstleistungen. Ein Kunde kauft bei diesem Händler ein. Da dieser Händler verdächtigt wird, an Geldwäsche beteiligt zu sein, wird der Kunde deanonymisiert. Nun wird ebenfalls gegen den Kunden ermittelt, obwohl dieser an der Geldwäsche völlig unbeteiligt ist. Unter Umständen wird der Kunde jedoch niemals darüber informiert, daß dieser Verdacht gegen ihn vorlag und seine Zahlung deanonymisiert wurde.

Mit Bargeld ist Münzverfolgung möglich, aber sehr ineffizient (siehe Einleitung dieses Kapitels). Kundenverfolgung ist mit Bargeld nicht realisierbar. Obwohl Bargeld einen hohen Grad an Anonymität bietet, ist es dennoch relativ unattraktiv für kriminellen Mißbrauch, da die Handhabung

Zahlungssystem	Treuhänder	Deanonymisierung
Brickell, Gemmell, Kravitz [BGK95]	passiv, verteilt	Münzen
Jakobsson, Yung [JY96]	aktiv	Münzen, Kunden
Petersen, Poupard [PP97]	passiv	Münzen, Kunden
Camenisch, Maurer, Stadler [CMS96]	passiv,verteilt	Münzen, Kunden
(Davida,) Frankel, Tsiounis, Yung [FTY96, DFTY97]	passiv, verteilt möglich	Münzen, Kunden
Jakobsson, Yung [JY97, JM99]	aktiv, verteilt	Münzen, Kunden

Tabelle 1.1: Übersicht treuhänderbasierter Zahlungssysteme

und Umverteilung von großen Mengen an Bargeld kompliziert und riskant ist. Im Gegensatz dazu wird durch elektronische Zahlungssysteme die Handhabung von großen Geldmengen signifikant erleichtert, so daß die im Vergleich zu Bargeld stärkeren Deanonymisierungsmechanismen Münz- und Kundenverfolgung unserer Meinung nach gerechtfertigt sind.

1.4.2 Treuhänderbasierte Zahlungssysteme

Wir unterscheiden zwischen *aktiven* und *passiven* Treuhändern. Ein aktiver Treuhänder ist an jedem Abhebevorgang direkt beteiligt, während ein passiver Treuhänder an den Abhebevorgängen unbeteiligt ist. Zusätzlich kann die Funktion des Treuhänders auf mehrere Institutionen *verteilt* sein, so daß die Mehrheit der verteilten Treuhänder einer Deanonymisierung zustimmen muß. Das grundsätzliche Problem der treuhänderbasierten Zahlungssysteme ist, daß die Deanonymisierungsmechanismen mißbraucht werden können. Der Treuhänder ist nicht kontrollierbar. Je nach Zahlungssystem kann der Treuhänder entweder alleine oder mit der Bank zusammen jede Zahlung rückwirkend deanonymisieren. Aber nicht nur der Kunde, sondern auch die Bank bzw. die Strafverfolgungsbehörden müssen dem Treuhänder Vertrauen entgegenbringen. Weigert sich ein Treuhänder an einer Deanonymisierung teilzunehmen, kann diese nicht durchgeführt werden.

Eine Übersicht über eine Auswahl an verschiedenen treuhänderbasierten Zahlungssystemen findet sich in Tabelle 1.1. Im folgenden zeigen wir, in wie weit treuhänderbasierte Zahlungssysteme geeignet sind, um anonymitätsbezogene Probleme zu lösen.

Erpressung

Erpressung von Kunden kann durch Münzverfolgung gelöst werden. Erpreßte blinde Münzen werden vom Treuhänder deanonymisiert. Die deanonymisierten Münzen werden auf eine schwarze Liste gesetzt, so daß die Bank diese Münzen beim Einzahlen nicht akzeptieren wird.

Geldwäsche

Geldwäsche kann theoretisch allein über Münzverfolgung bekämpft werden. Die Münzen von verdächtigen Kunden werden deanonymisiert, so daß herausgefunden werden kann, bei welchem Händler die Münzen eingelöst werden. Der Einsatz von Kundenverfolgung ist jedoch effektiver. Mit Kundenverfolgung kann ein verdächtiger Händler überprüft werden. Dadurch kann herausgefunden werden, welche Kunden bei diesem Händler einkaufen.

Bankraub

Bankraub kann durch Münz- und Kundenverfolgung nicht verhindert werden. Die Erstellung von Münzen ist unter Kontrolle der Bankräuber, so daß die Deanonymisierungsmechanismen umgangen werden können. Um regulär abgehobene Münzen von gestohlenen Münzen unterscheiden zu können, muß ein zusätzlicher Mechanismus gefunden werden.

Einen derartigen Mechanismus bieten die Zahlungssysteme von Jakobsson und Yung [JY96] und von Petersen und Poupard [PP97].

1.4.3 Freiwillige Deanonymisierung

Eine interessante Variante von treuhänderbasierten Zahlungssystemen wird von Pfitzmann und Sadeghi [PS01] vorgeschlagen: Der Kunde soll die Rolle des Treuhänders übernehmen. Auf diese Weise kann das Problem der Erpressung elegant gelöst werden. Nach einer Erpressung ist der Kunde selbst in der Lage, die erpreßten Münzen zu deanonymisieren. Dadurch können die erpreßten Münzen invalidiert werden, so daß die Bank diese Münzen beim Einzahlen nicht akzeptieren wird. Dieser Ansatz hat aber zwei gravierende Nachteile:

- Es ist nicht möglich, die Probleme Geldwäsche und Bankraub zu lösen.
- Der Kunde kann gezwungen werden, sich *unfreiwillig* zu deanonymisieren.

Diese beiden Probleme sind miteinander verbunden. Es gibt keine Möglichkeit, das Problem der Geldwäsche über einen Deanonymisierungsmechanismus zu lösen. Daher könnte es zu folgender Konstruktion kommen: Die Strafverfolgungsbehörden fordern von einem verdächtigen Kunden, seine abgehobenen Münzen zu deanonymisieren. In diesem Fall liegt eine Umkehr der Beweislast zugrunde. Normalerweise gilt ein Angeklagter solange als unschuldig, bis seine Schuld erwiesen ist. Hier gilt ein Angeklagter als schuldig, es sei denn, er kann seine Unschuld beweisen. Wir halten diese Konstruktion daher für äußerst bedenklich. Man betrachte dazu den Fall, daß ein Kunde die zum Deanonymisieren seiner Münzen notwendigen geheimen Daten tatsächlich nicht mehr besitzt (z.B. durch Verlust einer SmartCard oder Datenverlust einer Festplatte). In diesem Fall wird der Kunde verurteilt, da es ihm unmöglich ist, seine Unschuld zu beweisen.

Dennoch existiert diese Konstruktion bereits in der Praxis: Das britische Gesetz zur “Regulation of Investigationary Powers” [RIP] besagt, daß der Empfänger verschlüsselter Daten in der Lage sein muß, diese den Strafverfolgungsbehörden auf Verlangen zu entschlüsseln.

1.5 Alternative Konzepte

Nicht alle münzbasierten Zahlungssysteme beruhen auf dem Konzept der blinden Signatur. Wir stellen drei Ansätze vor, die auf anderen Verfahren basieren.

1.5.1 Blind Auditable Membership Proofs

Sander und Ta-Shma [STS99a] präsentieren ein Zahlungssystem, in dem Münzen die Blätter eines öffentlich bekannten Hash-Baumes sind. Zum Bezahlen muß der Kunde beweisen, daß er einen Weg von der Wurzel des Baumes zu einer Münze kennt. Diesen Beweis erbringt der Kunde, ohne dabei bekannt zu geben, für welche Münze er diesen Beweis erbringt. Dieser Beweis wird *Blind Auditable Membership Proof* genannt. Eine effizientere Implementierung dieses Zahlungssystems ist in [STSY01] zu finden.

Da die Bank weiß, welchem Kunde welche Blätter bzw. Münzen in diesem Baum gehören, können diese bei Bedarf leicht entfernt werden. Dadurch lassen sich Erpressung und Bankraub leicht lösen, Geldwäsche aber nicht.

Während dieses Zahlungssystem aus theoretischer Sicht sehr interessant ist, ist die praktische Umsetzung schwer zu realisieren: Der Hash-Baum muß bei jedem Abhebevorgang aktualisiert werden. Ein effizientes Broadcast-Medium ist daher notwendig, um die Änderungen ständig an alle Händler und Kunden zu übertragen.

1.5.2 Self-Scrambling Anonymizers

Pointcheval [Poi01] stellt ein Zahlungssystem vor, in dem eine Münze erst nach dem Abheben anonymisiert wird.

Abheben: Die Bank verschlüsselt den öffentlichen Schlüssel des Kunden für einen Treuhänder mit einer randomisierten Chiffre. Anschließend signiert sie den Chiffretext. Die Münze ist der signierte Chiffretext.

Anonymisieren: Der Kunde rerandomisiert den Chiffretext einer Münze und sendet die Münze und den rerandomisierten Chiffretext an einen Anonymisierer (praktisch gesehen ist das eine andere Bank). Der Anonymisierer signiert den rerandomiserten Chiffretext, nachdem der Kunde zwei Eigenschaften bewiesen hat:

1. Beide Chiffretexte (die Münze und der rerandomisierte Chiffretext) haben den gleichen Klartext.
2. Der Kunde kennt den privaten Schlüssel zum verschlüsselten öffentlichen Schlüssel.

Der Anonymitätsprovider gibt die neue Münze an den Kunden und die alte Münze an die (vorhergehende) Bank. Das Anonymisieren der Münze kann beliebig oft wiederholt werden.

Bankraub und Erpressungen lassen sich mit diesem Zahlungssystem leicht vermeiden, da die Bank die legalen (nicht-anonymen) Münzen kennt. Versucht ein Anonymitätsprovider eine Münze zu anonymisieren, muß er die nicht-anonyme Münze an die vorhergehende Bank zurückgeben. Die herausgebende Bank lehnt schließlich die illegale Münze einfach ab.

Bei Verdacht auf Geldwäsche kann Kundenverfolgung angewendet werden. In diesem Fall entschlüsselt der Treuhänder den Chiffretext der Münze und erhält den öffentlichen Schlüssel des Kunden. Wie bei allen treuhänderbasierten Zahlungssystemen ist der Deanonymisierungsmechanismus nicht kontrollierbar.

1.5.3 Mini-Cash

Ein weiteres Zahlungssystem von Jakobsson [Jak99] verwendet einen Hybridansatz aus Münzen und Konten: Eine Münze ist ein Konto (bei einer Bank) mit festem Wert. Wird eine Münze benutzt, erhält ausschließlich der neue Besitzer Zugriff auf das Konto. Eine Kette von Transaktionen entsteht, von der die Bank nur den ersten Kunden (der ein Münzkonto anlegt) und den letzten Kunden (der ein Münzkonto auflöst) kennt. Zusätzlich weiß die Bank, wie oft der Besitzer des Kontos gewechselt hat. Mit der Länge der Kette steigt die Anonymität der Zahlungen. Hat die Kette nur zwei Kunden, ist die Transaktion nicht anonym. Jakobsson behauptet, das Zahlungssystem ist nicht deanonymisierbar, aber gegen Erpressung und Bankraub geschützt. Wir zeigen das Gegenteil von beiden Aussagen:

- Das Zahlungssystem ist deanonymisierbar: Unter der Annahme, daß jeder Kunde bei einer Übereignung eines Münzkontos die Identität des Händlers speichern muß, kann Münzverfolgung realisiert werden. Dieser Mechanismus erfordert die Kooperation der betroffenen Kunden. Daher sind Deanonymisierungen kontrollierbar und gegen Mißbrauch geschützt. Weigert sich ein Kunde zu kooperieren, liegt der Verdacht nahe, daß dieser Kunde an kriminellen Aktivitäten beteiligt ist. Das Problem der freiwilligen Deanonymisierung haben wir bereits in Abschnitt 1.4.3 behandelt.

- Erpressung und Bankraub sind möglich: Im Falle von Erpressung und Bankraub weiß die Bank, welche Münzkonten davon betroffen sind. Diese Münzkonten können daher gelöscht und das Geld kann auf das ursprüngliche Konto zurücktransferiert werden. Hat der Besitzer des Münzkontos jedoch bereits gewechselt (tatsächlich oder nur scheinbar, indem der Täter sich selbst bezahlt), ist es nicht mehr möglich, dieses Münzkonto zu löschen. Es ist also leicht, erpreßtes Geld zu waschen. Wie sich hier zeigt, ist Münzverfolgung nicht ausreichend, um diese Art von Geldwäsche zu verhindern.

Kapitel 2

FlexiCash: Ein Zahlungssystem mit markierbaren Münzen

Im vorhergehenden Kapitel haben wir gezeigt, daß die vollständige Anonymität des Zahlenden zu Problemen führen kann. Der treuhänderbasierte Ansatz ist bisher am besten geeignet, um diese Probleme zu lösen. Dieser Ansatz hat jedoch einen gravierenden Nachteil: Deanonymisierungen sind nicht kontrollierbar. Es immer möglich, die Identität des Zahlenden aufzudecken. Es kann aber nicht garantiert werden, daß Deanonymisierungen ausschließlich auf die wirklich notwendigen Fälle beschränkt werden. Es fehlt ein Kontrollmechanismus für Deanonymisierungen.

In diesem Kaptitel entwerfen wir das erste Zahlungssystem, das die Probleme Erpressung, Geldwäsche und Bankraub lösen kann, ohne die Anonymität des Kunden stark einzuschränken. Unser Zahlungssystem ist zum Deanonymisieren einer Zahlung nicht auf dritte Parteien angewiesen. Deanonymisierungen sind immer nachträglich kontrollierbar. Dadurch kann der Zahlende selbst überprüfen, ob seine Zahlungen deanonymisert wurden. Da Deanonymisierungen überprüfbar sind, ist ein Mißbrauch von Deanonymisierungen praktisch ausgeschlossen.

Gliederung des Kapitels

In Abschnitt 2.1 beschreiben wir unsere Idee für ein Zahlungssystem mit überprüfbarem Deanonymisierungsmechanismus. Dieses Zahlungssystem realisieren wir in Abschnitt 2.2 auf Basis von markierbaren Münzen. In Abschnitt 2.3 zeigen wir, daß unser Zahlungssystem die Grundeigenschaften für ein anonymes Zahlungssystem erfüllt. Schließlich beschreiben wir in Abschnitt 2.4, wie Erpressung, Geldwäsche und Bankraub in unserem Zahlungssystem behandelt werden.

2.1 Überprüfbare Deanonymisierung

Wir betrachten zunächst die Unterschiede zwischen Bargeld und elektronischen Münzen. Bargeld hat zwei wesentliche Eigenschaften, die elektronische Münzen so nicht besitzen:

1. Die Seriennummern der Geldscheine können beim Abheben notiert werden.
2. Geldscheine können beim Abheben “unsichtbar” markiert werden.

Beide Methoden dienen dazu, die herausgegebenen Geldscheine später dem Abhebevorgang wieder zuordnen zu können. Dadurch kann der Fluß einer großen Menge von Bargeld zumindest teilweise wieder rekonstruiert werden. Bei Erpressungen geht man beispielsweise so vor:

> Die Seriennummern der abgehobenen Geldscheine werden notiert und anschließend dem Erpresser übergeben. Der Erpresser benutzt die Geldscheine zum Bezahlen (oder zahlt sie sogar selbst auf sein eigenes Konto ein). Werden die ausgegebenen Geldscheine eingezahlt, entdeckt die Bank, daß diese Geldscheine aus einer Erpressung stammen. Die Bank meldet den Strafverfolgungsbehörden, wann und von wem diese Geldscheine eingezahlt wurden. Dadurch kann ein Profil des Täters erstellt werden, das zu dessen Ergreifung führen kann.
>
> Nach dem Ergreifen des Täters können alle gefundenen und unbenutzten Geldscheine dem Opfer gutgeschrieben werden. Es ist allerdings nicht möglich, dem Opfer bereits ausgegebene Geldscheine zu erstatten, wenn der neue Besitzer diese in gutem Glauben erhalten hat (§932 BGB).

Das Notieren von Seriennummern ist in etwa vergleichbar mit Münzverfolgung durch Treuhänder. Der Unterschied ist, daß Treuhänder die Seriennummern grundsätzlich notieren müssen und nicht nur ausschließlich bei Bedarf. Erst dadurch wird auch Kundenverfolgung möglich.

Wir verfolgen hier erstmals den Ansatz der *unsichtbaren Markierungen*. Das Ziel der unsichtbaren Markierungen ist es, Deanonymisierungen auf kontrollierbare Weise zu ermöglichen. Im Unterschied zum Notieren von Seriennummern wird eine Markierung nur dann in eine Münze eingebettet, wenn die Münze verfolgt werden soll. Dadurch realisieren wir Münzverfolgung. Später werden wir zeigen, wie wir den Deanonymisierungsmechanismus auf Kundenverfolgung erweitern, ohne die Kontrollierbarkeit der Deanonymisierungen einzuschränken.

2.1.1 Zusätzliche Teilnehmer

In Abschnitt 1.1.1 haben wir die Teilnehmer eines münzbasierten Zahlungssystems beschrieben: Banken, Kunden, Händler, Richter und Verzeichnisse. Wir erweitern die Aufgaben des Richters und führen einen zusätzlichen Teilnehmer ein: die *Strafverfolgung*.

Strafverfolgung

Die Aufgabe der Strafverfolgung ist es Straftaten aufzuklären. Für die Aufklärung kann es notwendig sein, Zahlungen zu deanonymisieren. Ob eine Zahlung deanonymisiert werden soll, ist von zwei Faktoren abhängig:

1. Welcher Kunde die Münzen abhebt.

2. Welcher Händler die Münzen einzahlt.

Die Strafverfolgung kann die Bank anweisen, Deanonymisierungen durchzuführen. Damit die Durchführung von Deanonymisierungen legal ist, benötigt die Bank ein Deanonymisierungszertifikat. Deanonymisierungszertifikate beantragt die Strafverfolgung bei einem Richter.

Die Strafverfolgung ist immer über mehrere Instanzen verteilt. Die einzelnen Instanzen müssen sich nicht koordinieren.

Richter

Der Richter stellt Deanonymisierungszertifikate aus und verurteilt illegale Deanonymisierungen. Damit der Richter ein Deanonymisierungszertifikat für einen bestimmten Kunden bzw. Händler ausstellt, muß die Strafverfolgung ausreichende Verdachtsmomente gegen den Kunden bzw. Händler vorbringen können. Ein gültiges Deanonymisierungszertifikat hat folgende Struktur:

1. Zu signierende Daten:

   (a) Die Identität des zu deanonymisierenden Kunden bzw. Händlers.

   (b) Die Münzgeneration, in der die Deanonymisierung durchgeführt werden soll.

2. Die Signatur des Richters über die zu signierenden Daten.

3. Das Zertifikat des Richters.

Es existieren immer mehrere voneinander unabhängige Richter. Die einzelnen Richter müssen sich koordinieren: Jeder Richter kann in Erfahrung bringen, ob ein anderer Richter ein Deanonymisierungszertifikat für einen bestimmten Kunden oder Händler ausgestellt hat.

2.1.2 Zusätzliche Vorgänge

Für ein münzbasiertes Zahlungssystem haben wir die Vorgänge INITIALISIEREN, ABHEBEN, BEZAHLEN, EINZAHLEN und ZURÜCKGEBEN in Abschnitt 1.1.2 beschrieben. Wir führen zwei zusätzliche Vorgänge ein: VERFOLGEN (mit markierten Münzen) und ÜBERPRÜFEN (von markierten Münzen). Das Verfolgen von Münzen ist zum einen an das Abheben, und zum anderen an das Bezahlen und Einzahlen gekoppelt. Das Überprüfen von Münzen ist von allen anderen Vorgängen getrennt.

Abheben mit Verfolgen

Der Kunde erhält Münzen von der Bank. Ist die Bank im Besitz eines Deanonymisierungszertifikates für diesen Kunden, werden die Münzen unsichtbar markiert. Die Bank belastet das Konto des Kunden mit dem Betrag der abgehobenen Münzen und schreibt den Betrag einem Verrechnungskonto gut.

Bezahlen und Einzahlen mit Verfolgen

Der Kunde gibt seine abgehobenen Münzen an einen Händler. Der Händler gibt die vom Kunden erhaltenen Münzen der Bank. Entdeckt die Bank unsichtbar markierte Münzen, kann die Bank die Münzen zum Abhebevorgang zurückverfolgen. Wenn die Münzen nicht aus einer Erpressung stammen, schreibt die Bank den Betrag der eingezahlten Münzen dem Konto des Händlers gut und belastet das Verrechnungskonto mit dem Betrag. Der Kunde erhält die bezahlte Leistung vom Händler.

Überprüfen

Die Bank publiziert, wie Markierungen entdeckt werden können. Die Kunden überprüfen ihre abgehobenen Münzen auf Markierungen. Für entdeckte markierte Münzen muß die Bank ein Deanonymisierungszertifikat vorweisen können, andernfalls wird die Bank von einem Richter wegen illegaler Deanonymisierung verurteilt.

2.1.3 Zusätzliche Eigenschaften

In Abschnitt 1.1.3 haben wir die Eigenschaften eines münzbasierten Zahlungssystems definiert. Wir haben jetzt zwei zusätzliche Vorgänge, für die wir zusätzliche Eigenschaften fordern. Wir fassen diese Eigenschaften unter dem Begriff *überprüfbare Deanonymisierungen* zusammen:

- *Die Anonymität eines Kunden kann grundsätzlich aufgehoben werden.*
- *Ein Kunde kann nur beim Abheben oder beim Bezahlen deanonymisiert werden.*
- *Der Kunde kann Deanonymisierungen seiner Zahlungen kontrollieren.*

2.2 Realisierung von FlexiCash

In diesem Abschnitt beschreiben wir die Realisierung der Vorgänge unseres Zahlungssystems mit markierbaren Münzen. Wir beschreiben zwei Varianten des Zahlungssystems. Die Basisvariante, die ausschließlich Münzverfolgung unterstützt, sowie die erweiterte Variante, die zusätzlich Kundenverfolgung ermöglicht.

2.2.1 Seriennummern der Münzen

Als Seriennummern von Münzen verwenden wir Münzschlüssel (s. Abschnitt 1.2.2) und fügen zusätzlich einen *Authentisierungscode* hinzu. Eine Münze hat demnach folgende Struktur:

1. Die Seriennummer der Münze:
   (a) Ein zufällig gewählter Münzschlüssel.
   (b) Ein Authentisierungscode.
2. Die blinde Signatur der Bank über die Seriennummer.

Der Authentisierungscode wird mit einem geheimen Rückgabeschlüssel über den zu verwendenden Blendparameter berechnet. Dadurch wird der Kunde gezwungen, zuerst ein Münzschlüsselpaar, einen Rückgabeschlüssel und einen Blendparameter zu wählen. Anschließend kann der Kunde die Seriennummer berechnen.

2.2.2 Münzen mit Tags

Die Seriennummer einer Münze wird allein vom Kunden gewählt. Daher ist eine für den Kunden unsichtbare Einbettung von Markierungen in die Seriennummer schwer möglich. Ebenso schwer ist es, diese Markierung in die Signatur der Bank einzubetten, da jede Manipulation der Signatur öffentlich erkennbar ist.

Um dieses Problem zu lösen, führen wir den Begriff des *Tags* ein. Ein Tag ist ein von der Bank ausgestellter “Anhänger” an einer Münze, in den eine Markierung unsichtbar eingebettet wird. Tags enthalten immer eine Markierung. Soll eine Münze nicht markiert werden, wird eine spezielle Markierung in das Tag eingebettet. Wir nennen diese spezielle Markierung die *Standardmarkierung*. Die Standardmarkierung ist für alle anonymen Münzen gleich. Münzen mit Standardmarkierung heißen *unmarkiert*.

Zum Einbetten und Extrahieren einer Markierung in ein Tag wird ein geheimer *Markierungsschlüssel* verwendet. Wie die Tags implementiert werden, beschreiben wir ausführlich im nächsten Kapitel. Zunächst gehen wir von folgenden Eigenschaften der Tags aus:

- *Ohne Kenntnis des Markierungsschlüssels kann nichts über die in einem Tag eingebettete Markierung ausgesagt werden.*
- *Tags werden von der Bank blind ausgestellt, so daß die Bank ein Tag später nicht an seiner äußeren Form erkennen kann.*
- *Der Versuch, eine in einem Tag eingebettete Markierung zu manipulieren, resultiert mit hoher Wahrscheinlichkeit in einer ungültigen Markierung.*
- *Der Versuch, eine Münze mit dem Tag einer anderen Münze zu benutzen, resultiert mit hoher Wahrscheinlichkeit in einer ungültigen Markierung.*

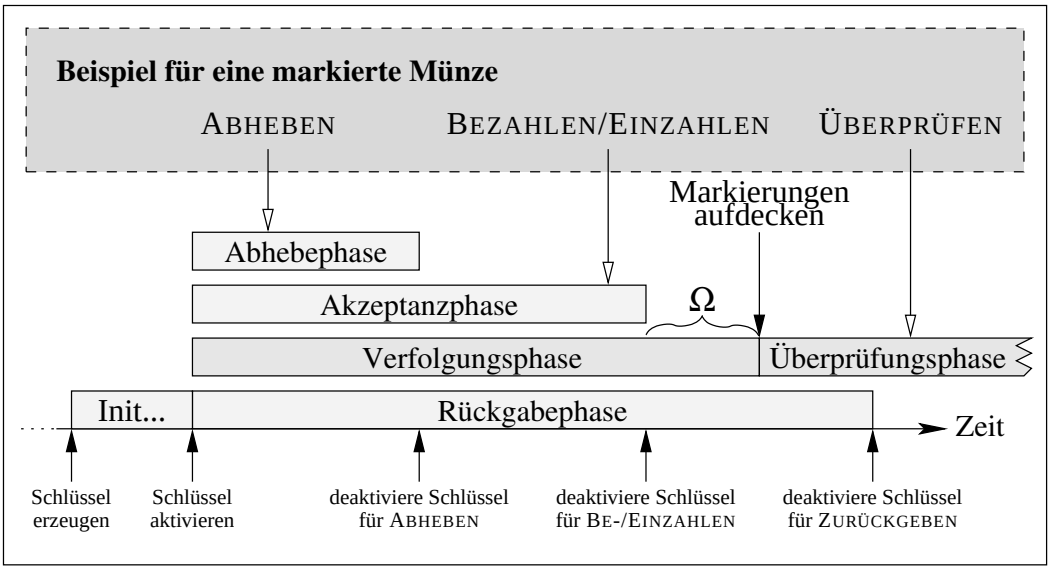

Abbildung 2.1: Zusätzliche Phasen einer Münzgeneration

2.2.3 Zusätzliche Phasen einer Generation

In Abschnitt 1.2.3 haben wir Münzgenerationen eingeführt und jede Generation in folgende Phasen unterteilt: die Initialisierungsphase, die Abhebephase, die Akzeptanzphase und die Rückgabephase. Die beiden neuen Vorgänge VERFOLGEN und ÜBERPRÜFEN erzwingen zwei zusätzliche Phasen in der Münzgeneration. Wir nennen diese Phasen die *Verfolgungsphase* und die *Überprüfungsphase*. Abbildung 2.1 zeigt, wie sich die beiden neuen Phasen zu der Abhebe-, der Akzeptanz- und der Rückgabephase verhalten.

Verfolgungsphase

In der Verfolgungsphase sind Markierungen für den Kunden unsichtbar. Alle Zahlungen mit markierten Münzen werden von der Bank deanonymisiert. Deanonymisierungen sind während der Verfolgungsphase für den Kunden nicht entdeckbar. Die Verfolgungsphase beginnt gleichzeitig mit der Akzeptanzphase, ist aber um einen festgelegten Zeitraum Ω länger. Damit ist garantiert, daß der Strafverfolgung mindestens die Zeit Ω für Ermittlungen in Verdachtsfällen zur Verfügung steht.

Überprüfungsphase

In der Überprüfungsphase publiziert die Bank die Standardmarkierung und den geheimen Markierungsschlüssel. Dadurch kann jeder Kunde seine abgehobenen Münzen selbst auf Markierungen überprüfen. Stellt der Kunde eine Deanonymisierung fest, muß die Bank ein Deanonymisie-

Sitzungstabelle
Sitzungsnummer Zeitpunkt Authentisierung Identität Kunde/Händler

Abhebetabelle
Sitzungsnummer blinde Münze

Einzahltabelle
Sitzungsnummer Münze: – Generation – Seriennummer – Münzsignatur – Markierung

Markierungstabelle
Sitzungsnummer Markierung Deanonymisierungszert.

Abhebeauthorisierungen
Sitzungsnummer Abhebeautorisierung: – Signatur des Kunden

Annahmeerklärungen
Sitzungsnummer Annahmeerklärung: – Identität des Händlers – Angebotsnummer

Abbildung 2.2: Datenstrukturen der Bank

rungszertifikat vorlegen. Besitzt die Bank kein Deanonymisierungszertifikat, ist die Deanonymisierung illegal.

Aus Gründen des Datenschutzes sollten die Deanonymisierungszertifikate auch in der Überprüfungsphase nicht öffentlich zugänglich sein, sondern nur an die jeweils betroffenen Personen gegeben werden dürfen.

2.2.4 Realisierung der Teilnehmer

Wie schon in Abschnitt 1.2.4 beschrieben, müssen für die Banken, Kunden, Händler, Verzeichnisse, Richter einige Vorbereitungen getroffen werden. Für die Strafverfolgung als zusätzlichen Teilnehmer sind keine besonderen Vorbereitungen notwendig.

Allen Teilnehmern ist gemein, daß sie zunächst Schlüsselpaare erzeugen müssen, die zur Authentisierung sowie zum Signieren verwendet werden. Die öffentlichen Schlüssel werdem von Verzeichnis zertifiziert und dort publiziert.

Banken

Die Bank benötigt eine Datenbank, in der die Kunden- und Kontodaten verwaltet werden. Für die Realisierung der Vorgänge werden weiterhin folgende Tabellen in der Datenbank angelegt: Eine Sitzungstabelle, eine Markierungstabelle, eine Abhebetabelle, eine Abhebeauthorisierungstabelle, eine Einzahltabelle und Annahmeerklärungstabelle. Diese Tabellen sind in Abbildung 2.2 dargestellt und werden im folgenden näher erläutert:

`Sitzungstabelle:` In der Sitzungstabelle werden nach einer erfolgreichen Authentisierung eines Kunden bzw. Händlers Informationen über die Authentisierung gespeichert. Es wird mindestens die Identität des Kunden bzw. Händlers und der Zeitpunkt der Authentisierung gespeichert. Beim Speichern wird eine eindeutige *Sitzungsnummer* von der Datenbank vergeben. Mit dieser Sitzungsnummer werden alle weiteren zu speichernden Daten dieser Sitzung verknüpft.

`Markierungstabelle:` In der Markierungstabelle wird jeweils die Sitzungsmarkierung gespeichert, die zum Markieren der Münzen verwendet wird, die in dieser Sitzung abgehoben werden. Über die Sitzungsnummer kann eine entdeckte Markierung der Kundenidentität zugeordnet werden.

`Abhebetabelle:` In der Abhebetabelle werden die blinden Münzen gespeichert, d.h. die Sicht der Bank auf die abgehobenen Münzen. Über die Sitzungsnummer kann bei einer Rückgabe überprüft werden, welcher Kunde die blinde Münze abgehoben hat.

`Abhebeauthorisierungstabelle:` In der Abhebeauthorisierungstabelle werden die Abhebeauthorisierungen (s. Abhebeprotokoll) des Kunden gespeichert. Die Abhebeautorisierung ist eine Signatur des Kunden über die blinden Münzen. Da die blinden Münzen bereits in der Abhebetabelle gespeichert wurden, muß hier nur die Signatur des Kunden gespeichert werden.

`Einzahltabelle:` In der Einzahltabelle werden die eingezahlten bzw. zurückgegebenen Münzen gespeichert. Zusätzlich wird zu jeder Münze die mit dem öffentlichen Münzschlüssel verifizierbare Münzsignatur sowie die in der Münze enthaltene Markierung abgespeichert.

`Annahmeerklärungstabelle:` In der Annahmeerklärungstabelle werden die Annahmeerklärungen gespeichert. Die Annahmeerklärungen werden unsigniert gespeichert, da die Münzsignaturen zu jeder eingezahlten Münze bereits in der Einzahltabelle gespeichert wurden.

Kunden

Der Kunde besitzt eine elektronische Geldbörse, mit der die von ihm abgehobenen Münzen verwaltet werden. Dazu führt die elektronische Geldbörse folgende Listen: eine Münzliste, eine Abhebeliste und eine Zahlungsliste. Diese Listen sind in Abbildung 2.3 dargestellt und werden im folgenden näher erläutert:

`Münzliste:` In der Münzliste werden die abgehobenen Münzen gespeichert. Nachdem der Liste eine neue Münze angefügt wurde, kann auf diese Münze über die eindeutige, fortlaufende Münznummer zugegriffen werden. Zu jeder Münze wird die Abhebenummer und, falls die Münze verwendet wurde, die Zahlungsnummer (s. Zahlungsliste) gespeichert.

Abbildung 2.3: Datenstrukturen des Kunden

`Abhebeliste:` In der Abhebeliste wird gespeichert, welche Münzen in welchem Abhebevorgang erzeugt wurden. Für jeden Abhebevorgang wird eine neue, fortlaufende Abhebenummer vergeben. Unter dieser eindeutigen Nummer wird das Abhebezertifikat (s. Abhebeprotokoll) und die Münznummern (s. Münzliste) der abgehobenen Münzen gespeichert. Da die Münzen bereits in der Münzliste vorhanden sind, wird für das Abhebezertifikat nur die Signatur der Bank gespeichert.

`Zahlungsliste:` In der Zahlungsliste wird gespeichert, welche Münzen in welchem Zahlungsvorgang verwendet wurden. Für jeden Zahlungsvorgang wird eine neue, fortlaufende Zahlungsnummer vergeben. Unter dieser eindeutigen Nummer werden die Münznummern (s. Münzliste) der für die Zahlung verwendeten Münzen und das Angebot des Händlers gespeichert. In der erweiterten Variante wird zusätzlich ein von der Bank erstelltes Einzahlzertifikat (s. Zahlungsprotokoll) gespeichert.

Händler

Der Händler verwendet eine Datenbank, mit der die angebotenen Produkte verwaltet werden. Für jede Bestellung wird eine neue, eindeutige Bestellnummer vergeben. Unter der Bestellnummer werden die bestellten Produkte sowie der Status der Bestellung (z.B. Angebot verschickt, Münzen erhalten, Betrag gutgeschrieben, Produkte geliefert) gespeichert.

Da diese Tabellen für die Realisierung der Vorgänge nur wenig relevant sind, verzichten wir hier auf eine detailliertere Beschreibung der Tabellen.

Verzeichnisse

Das Verzeichnis verwendet eine Datenbank, in der die vom Verzeichnis selbst erstellten Zertifikate gespeichert werden. Die Zertifikate der vom Verzeichnis erstellten Signaturschlüssel haben folgende Struktur:

1. Zu signierende Daten:
   (a) Die Identität des Verzeichnisses.
   (b) Die Identität der Bank bzw. des Kunden.
   (c) Der öffentliche Schlüssel.
   (d) Die Gültigkeit des Schlüssels:
      i. Zeitpunkt, ab dem der Schlüssel gültig ist.
      ii. Zeitpunkt, ab dem der Schlüssel nicht mehr gültig ist.
   (e) Die Nutzbarkeit des Schlüssels:
      i. Die Kennzeichnung für digitale Signatur.
2. Die Signatur des Verzeichnisses über die zu signierenden Daten.

Ist ein Zertifikat abgelaufen, muß es durch ein neues Zertifikat ersetzt werden. Wie die Teilnehmer dabei mit dem Verzeichnis interagieren, wird nicht weiter betrachtet.

Richter

Der Richter führt eine Liste, in der die ausgestellten Deanonymisierungszertifikate gespeichert werden.

2.2.5 Basisvariante mit Münzverfolgung

Wir beschreiben in diesem Abschnitt die Realisierung der Vorgänge INITIALISIEREN, ABHEBEN, BEZAHLEN, EINZAHLEN, VERFOLGEN, ÜBERPRÜFEN und ZURÜCKGEBEN für die Basisvariante unseres Zahlungssystems.

Initialisieren

Bei der Initialisierung jeder Münzgeneration wählt die Bank neue Signaturschlüssel für die einzelnen Münzwerte, einen Markierungsschlüssel und die Standardmarkierung. Das Verzeichnis (s. Abschnitt 1.1.1) erstellt Zertifikate für die öffentlichen Signaturschlüssel und veröffentlicht diese. Die privaten Signaturschlüssel, der Markierungsschlüssel und die Standardmarkierung werden von der Bank geheim gehalten. Die Zertifikate für die öffentlichen Signaturschlüssel haben folgende Struktur:

1. Zu signierende Daten:
   (a) Die Identität des Verzeichnisses.
   (b) Die Identität der Bank.
   (c) Der öffentliche Schlüssel.
   (d) Die Gültigkeit des Schlüssels:
      i. Der Zeitpunkt, ab dem der Schlüssel gültig ist.
      ii. Der Zeitpunkt, ab dem der Schlüssel für das ABHEBEN ungültig ist.
      iii. Der Zeitpunkt, ab dem der Schlüssel für das BE-/EINZAHLEN ungültig ist.
      iv. Der Zeitpunkt, ab dem der Schlüssel für das ZURÜCKGEBEN ungültig ist.
      v. Der Zeitpunkt, ab dem der Schlüssel für ÜBERPRÜFEN gültig ist.
   (e) Die Nutzbarkeit des Schlüssels:
      i. Die Kennzeichnung für blinde Signatur.
      ii. Der Wert der Münze.
2. Die Signatur des Verzeichnisses über die zu signierenden Daten.

Aus Gründen der Übersichtlichkeit verwenden wir in unserer Realisierung nur einen einzigen Signaturschlüssel bzw. Münzwert.

Abheben mit Münzverfolgung

Bevor ein Kunde neue Münzen von seinem Konto abheben kann, muß sich der Kunde gegenüber der Bank authentisieren. Nach der erfolgreichen Authentisierung des Kunden speichert die Bank den Zeitpunk der Authentisierung sowie die Identität des Kunden in der Sitzungstabelle ab und erhält eine eindeutige Sitzungsnummer von der Datenbank. Ist die Bank im Besitz eines Deanonymisierungszertifikates für den Kunden, wählt die Bank eine neue, eindeutige Sitzungsmarkierung und speichert die Sitzungsmarkierung unter der Sitzungsnummer in der Markierungstabelle ab. Anschließend interagieren Kunde und Bank in folgendem Abhebeprotokoll.

Protokoll 4. Abhebeprotokoll für Münzen mit möglicher Münzverfolgung.

1. Die Bank und der Kunde erzeugen Münzen. Für jede abzuhebende Münze wählt der Kunde ein neues, zufälliges Schlüsselpaar (Münzschlüssel) und einen geheimen Rückgabeschlüssel. Der Kunde berechnet die Seriennummer (s. Abschnitt 2.2.1). Die Münzen sind die von der Bank blind signierten Seriennummern. Wir nennen die Sicht der Bank auf die Münzen *blinde Münzen*.

   Ist eine Münze nicht korrekt signiert, wird das Abheben abgebrochen.

2. Der Kunde speichert die Münzen in der Münzliste ab und erstellt eine *Abhebeauthorisierung* mit folgender Struktur:

   (a) Die Liste der blinden Münzen.

   (b) Die Signatur des Kunden über die Liste der blinden Münzen.

   (c) Das Zertifikat des Kunden.

   Der Kunde sendet die Signatur der Abhebeauthorisierung an die Bank.

3. Die Bank rekonstruiert die Abhebeauthorisierung und überprüft die Signatur des Kunden.

   Schlägt der Test fehl, wird das Abheben abgebrochen.

   Die Bank speichert alle blinden Münzen in der Abhebetabelle sowie die Abhebeauthorisierung in der Abhebeauthorisierungstabelle jeweils unter der Sitzungsnummer ab.

   Anschließend führt die Bank die Buchung “Kundenkonto an Verrechnungskonto” durch.

4. Die Bank und der Kunde erzeugen für jede ausgestellte Münze ein Tag. Wir nennen die Sicht der Bank auf die Tags *blinde Tags*. Die Bank bettet in alle Tags folgende Markierung ein:

   - Die Standardmarkierung, wenn die Münzen des Kunden *nicht* markiert werden sollen.
   - Die Sitzungsmarkierung, wenn die Münzen des Kunden markiert werden sollen.

5. Die Bank erstellt ein *Abhebezertifikat* mit folgender Struktur:

   (a) Zu signierende Daten:

      i. Die Identität des Kunden.

      ii. Die Münzgeneration der ausgestellten Münzen.

      iii. Die blinden Münzen und die blinden Tags.

   (b) Die Signatur der Bank über die zu signierenden Daten.

   (c) Das Zertifikat der Bank.

   Die Bank sendet die Signatur des Abhebezertifikates an den Kunden.

6. Der Kunde rekonstruiert das Abhebezertifikat und überprüft die Signatur der Bank.

   Schlägt der Test fehl, gibt der Kunde alle abgehobenen Münzen an die Bank zurück (s. Zurückgeben).

   Der Kunde trägt die Tags in der Münzliste nach und speichert das Abhebezertifikat in der Abhebeliste ab.

Bezahlen und Einzahlen mit Münzverfolgung

Bevor eine Zahlung durchgeführt wird, sendet der Händler ein Angebot an den Kunden. Das Angebot hat folgende Struktur:

1. Zu signierende Daten:
   (a) Die Bestellnummer.
   (b) Die Liste der bestellten Waren und Preise.
2. Die Signatur des Händlers über die zu signierenden Daten.
3. Das Zertifikat des Händlers.

Der Kunde überprüft, ob das Angebot mit den von ihm bestellten Waren übereinstimmt und ob die Signatur korrekt ist. Nimmt der Kunde das Angebot an, interagieren Kunde, Händler und Bank im folgenden Zahlungsprotokoll. Bevor Bank und Händler die Einzahlung starten, authentisiert sich der Händler gegenüber der Bank. Nach der erfolgreichen Authentisierung des Händlers speichert die Bank den Zeitpunkt der Authentisierung sowie die Identität des Händlers in der Sitzungstabelle ab und erhält eine eindeutige Sitzungsnummer von der Datenbank.

Protokoll 5. Zahlungsprotokoll mit möglicher Münzverfolgung.

1. Der Kunde speichert das Angebot in der Zahlungsliste ab und wählt die Münzen aus, die für die Zahlung verwendet werden sollen. Der Kunde erstellt aus dem Angebot und den Münzen eine Annahmeerklärung und sendet sie an den Händler. Die Annahmeerklärung hat folgende Struktur:
   (a) Zu signierende Daten:
      i. Die Identität des Händlers.
      ii. Die Bestellnummer aus dem Angebot.
   (b) Die Liste der mit den privaten Münzschlüsseln erstellten Signaturen.
   (c) Die Liste der zu verwendenden Münzen und Tags.
2. Der Händler überprüft die Annahmeerklärung:
   (a) Die Identität des Händlers muß korrekt sein.
   (b) Die Bestellnummer beim Händler muß vorhanden sein.
   (c) Der Betrag der Münzen muß dem zu zahlenden Betrag entsprechen.

   Schlägt einer der Tests fehl, bricht der Händler die Zahlung ab.

   Der Händler startet die Einzahlung und sendet die Annahmeerklärung an die Bank.

3. Die Bank überprüft die Einzahlung:

   - Die Bank überprüft die Annahmeerklärung:
     - Die Identität des Händlers muß mit der Identität des Händlers in der Annahmeerklärung übereinstimmen.
     - Die Münzsignaturen der Annahmeerklärung müssen mit den öffentlichen Münzschlüsseln verifizierbar sein.
   - Die Bank überprüft die Münzen:
     - Der Einzahlzeitpunkt muß jeweils in der Akzeptanzphase der verwendeten Münzen liegen.
     - Die Seriennummern der Münzen dürfen nicht in der Einzahltabelle enthalten sein.
     - Die Signaturen der Münzen müssen gültig sein.

   Schlägt einer dieser Tests fehl, wird die Einzahlung abgelehnt.

   Die Bank extrahiert die Markierungen aus den Tags:

   - Enthält ein Tag die Standardmarkierung, ist die Münze anonym.
   - Andernfalls muß die Markierung eine Sitzungsmarkierung sein und die Bank ermittelt die zugehörige Kundenidentität aus der Markierungs- und Sitzungstabelle.

   Enthält ein Tag eine ungültige Markierung, oder eine Sitzungsmarkierung aus einer Erpressung, wird die Einzahlung abgelehnt.

   Die Bank speichert die Münzen, die Markierungen und die Münzsignaturen in der Einzahltabelle sowie die Annahmeerklärung in der Annahmeerklärungstabelle jeweils unter der Sitzungsnummer ab. Die Münzen sind jetzt ungültig.

   Anschließend führt die Bank die Buchung “Verrechnungskonto an Händlerkonto” durch und informiert den Händler über die erfolgreiche Einzahlung.

4. Der Händler liefert die Waren an den Kunden.

Anmerkungen:

1. Wird eine Zahlung von der Bank unberechtigt (aus Sicht des Kunden) abgebrochen, kann die Zahlung erneut gestartet werden. In diesem Fall nimmt ein Richter anstelle des Händlers am Zahlungsprotokoll teil. Führt die Bank dennoch das Zahlungsprotokoll falsch durch, kann die Bank dafür belangt werden.

2. Lehnt die Bank eine Zahlung aufgrund einer Markierung ab, muß die Bank ein *Ablehnungszertifikat* ausstellen. Das Ablehnungszertifikat hat folgende Struktur:

(a) Zu signierende Daten:

i. Die Liste der abgelehnten Münzen.

ii. Die Liste der abgelehnten Tags.

(b) Die Signatur der Bank über die zu signierenden Daten.

(c) Das Zertifikat der Bank.

Zurückgeben

Bevor ein Kunde unbenutzte Münzen auf sein Konto zurückgeben kann, muß sich der Kunde gegenüber der Bank authentisieren. Nach der erfolgreichen Authentisierung des Kunden speichert die Bank den Zeitpunkt der Authentisierung sowie die Identität des Kunden in der Sitzungstabelle ab und erhält eine eindeutige Sitzungsnummer von der Datenbank. Anschließend interagieren Kunde und Bank in folgendem Rückgabeprotokoll.

Protokoll 6. Rückgabeprotokoll.

1. Der Kunde wählt die Münzen aus, die für die Rückgabe verwendet werden sollen und erstellt für jede Münze eine Rückgabesignatur, indem er die Seriennummer mit dem privaten Münzschlüssel signiert. Der Kunde sendet die Seriennummern, die Rückgabesignaturen, die blinden Münzen, die Blendparameter und die geheimen Rückgabeschlüssel an die Bank.

2. Die Bank überprüft die Rückgabe:

   (a) Die blinden Münzen müssen in der Abhebetabelle vorhanden und vom gleichen Kunden abgehoben worden sein.

   (b) Die Seriennummern dürfen nicht in der Einzahltabelle vorhanden sein.

   (c) Die Authentisierungscodes der Seriennummern (s. Abschnitt 2.2.1) müssen korrekt sein.

   (d) Die blinden Münzen müssen sich aus der Seriennummer berechnen lassen.

   (e) Die Rückgabesignaturen müssen mit dem öffentlichen Münzschlüssel verifizierbar sein.

   Schlägt einer der Tests fehl, wird die Rückgabe abgelehnt.

   Die Bank speichert die Münzen und die Rückgabesignaturen in der Einzahltabelle unter der Sitzungsnummer ab. Die Münzen sind jetzt ungültig.

   Anschließend führt die Bank die Buchung “Verrechnungskonto an Kundenkonto” durch und informiert den Kunden über die erfolgreiche Rückgabe.

Überprüfen

Mit Beginn der Überprüfungsphase publiziert die Bank die Standardmarkierung und den geheimen Markierungsschlüssel, mit dem die Markierung aus den Tags extrahiert werden kann. Für jede Münze der entsprechenden Münzgeneration führt der Kunde folgende Schritte durch:

1. Der Kunde überprüft Münzverfolgung:

   Für jede abgehobene Münze extrahiert der Kunde die Markierung aus dem Tag. Entdeckt der Kunde ein Tag, das nicht die Standardmarkierung enthält, wurde die zugehörige Münze beim Abheben deanonymisiert und die Bank muß ein Deanonymisierungszertifikat vorlegen können.

   Besitzt die Bank kein Deanonymisierungszertifikat für den Kunden, dann interagiert der Kunde mit einem Richter in folgendem Überprüfungsprotokoll für illegale Münzverfolgung.

2. Der Kunde überprüft abgelehnte Münzen:

   Für jeden Zahlungsvorgang, den die Bank aufgrund einer entdeckten Markierung abgelehnt hat, überprüft der Kunde das Ablehnungszertifikat. Entdeckt der Kunde keine markierte Münze, interagiert der Kunde mit einem Richter in folgendem Überprüfungsprotokoll für illegal abgelehnte Münzen.

Protokoll 7. Überprüfungsprotokoll für illegale Münzverfolgung.

1. Der Kunde sendet das Abhebezertifikat (s. Abhebeprotokoll) des Abhebevorgangs, in dem Münzen illegal deanonymisiert wurden, an den Richter.

2. Der Richter überprüft das Abhebezertifikat:

   - Es darf kein Deanonymisierungszertifikat für den Kunden in der Münzgeneration vorliegen.
   - Die Signatur des Abhebezertifikates muß gültig sein.
   - Mindestens ein blindes Tag enthält *nicht* die Standardmarkierung.

   Schlägt einer der Tests fehl, wird die Überprüfung abgebrochen. Liegt ein Deanonymisierungszertifikat für den Kunden vor, sendet der Richter das Zertifikat an den Kunden.

   Der Richter informiert den Kunden über die erfolgreiche Anzeige einer illegalen Deanonymisierung.

Protokoll 8. Überprüfungsprotokoll für illegal abgelehnte Münzen.

1. Der Kunde sendet das Ablehnungszertifikat (s. Abhebeprotokoll) des Abhebevorgangs, in dem Münzen illegal abgelehnt wurden, an den Richter.

2. Der Richter überprüft das Ablehnungszertifikat:

   - Die Signatur des Ablehnungszertifikates muß gültig sein.
   - Alle Tags müssen die Standardmarkierung enthalten.

   Schlägt einer der Tests fehl, wird die Überprüfung abgebrochen.

   Der Richter informiert den Kunden über die erfolgreiche Anzeige einer illegalen Ablehnung.

2.2.6 Erweiterte Variante mit Münz- und Kundenverfolgung

Treuhänderbasierte Zahlungssysteme unterstützen in der Regel Kundenverfolgung, unser Zahlungssystem bisher nicht. Wir zeigen, wie wir unsere Basisvariante so erweitern, daß auch Kundenverfolgung möglich wird.

- Die Realisierung von Münzverfolgung über Markierungen ist ein anschaulicher Prozeß: Münzen werden beim Abheben von der Bank markiert. Diese Markierungen werden beim Einzahlen der Münze von der Bank wiedererkannt.
- Bei Kundenverfolgung wird die Entscheidung, eine Münze zu deanonymisieren jedoch nicht beim Abheben getroffen. Die Entscheidung wird erst beim Einzahlen getroffen und richtet sich nach dem Händler, der die Münzen einzahlt.

Markierungen können nur beim Abheben einer Münze angebracht werden. Für Kundenverfolgung müßten wir Münzen grundsätzlich markieren. Grundsätzliches Markieren von Münzen widerspricht der Anonymität (vgl. treuhänderbasierte Zahlungssysteme in Abschnitt 1.4.2).

Wir ermöglichen Kundenverfolgung, indem wir die Ideen der unsichtbaren Markierungen mit denen des *Oblivious Transfers* kombinieren. Beim Abheben einer Münze erzeugt die Bank zwei Tags, ein *Markierungstag* und ein *Identitätstag*. Das Markierungstag wird genauso verwendet wie in der Basisvariante. Das Identitätstag enthält **immer** die Sitzungsmarkierung. Diese beiden Tags werden von der Bank zufällig permutiert. Die Permutation selbst wird in einem dritten Tag gespeichert, dem *Indextag*. Die drei Tags werden an den Kunden gegeben. Aus Sicht des Kunden nennen wir die drei Tags das Index-, das linke und das rechte Tag. Damit der Kunde das Index-, Markierungs- und Identitätstag einer Münze nicht vertauschen kann, wird jedes Tag mit einem eigenen Markierungsschlüssel erstellt.

Beim Bezahlen sendet der Kunde zunächst die Münze und das Indextag an den Händler, der sie bei der Bank einzahlt. Über das Indextag lernt die Bank die Zuordnung von Markierungs- und Identitätstag zum linken und rechten Tag. Nun kann die Bank wählen, welches der beiden anderen Tags sie sehen will. Je nachdem, ob die Bank Kundenverfolgung bei dem Händler durchführen will, wählt sie zwischen dem Markierungs- oder dem Identitätstag. Der Kunde weiß jedoch nicht, welchen Inhalt das von der Bank gewünschte Tag hat.

- Soll Kundenverfolgung durchgeführt werden, erhält die Bank die Identitätstags der Münzen.
- Andernfalls erhält die Bank die Markierungstags der Münzen und prüft auf mögliche Münzverfolgung.

Münzverfolgung und Kundenverfolgung schließen sich nicht gegenseitig aus: Wurde beim Abheben einer Münze Münzverfolgung durchgeführt, enthalten beide Tags die gleiche Markierung. Wird beim Bezahlen Kundenverfolgung durchgeführt, erhält die Bank das Identitätstag, das immer die Deanonymisierung des Kunden erlaubt. Wird beim Bezahlen keine Münzverfolgung durchgeführt, erhält die Bank das Markierungstag, das die Deanonymisierung des Kunden genau dann erlaubt, wenn beim Abheben Münzverfolgung durchgeführt wurde.

Die folgenden Protokolle sind die Erweiterungen der Protokolle aus dem vorherigen Abschnitt. Zum besseren Verständnis beschreiben wir die Vorgänge INITIALISIEREN, ABHEBEN, BEZAHLEN, EINZAHLEN, VERFOLGEN und ÜBERPRÜFEN jeweils noch einmal vollständig. Der Vorgang ZURÜCKGEBEN bleibt gegenüber der Basisvariante unverändert und wird nicht mehr beschrieben.

Initialisieren

Bei der Initialisierung jeder Münzgeneration wählt die Bank neue Signaturschlüssel für die einzelnen Münzwerte, drei Markierungsschlüssel und folgende Markierungen: Die Standardmarkierung, die *Nullmarkierung* und die *Einsmarkierung*. Die Nullmarkierung bzw. die Einsmarkierung sind wie die Standardmarkierung spezielle, zufällig gewählte Markierungen. Die Nullmarkierung repräsentiert den Wert Null und die Einsmarkierung repräsentiert den Wert Eins.

Das Verzeichnis (s. Abschnitt 1.1.1) erstellt Zertifikate für die öffentlichen Signaturschlüssel und veröffentlicht diese. Die privaten Signaturschlüssel, die Markierungsschlüssel, die Standardmarkierung, die Nullmarkierung und die Einsmarkierung werden von der Bank geheim gehalten. Die Zertifikate für die öffentlichen Signaturschlüssel haben folgende Struktur:

1. Zu signierende Daten:
   (a) Die Identität des Verzeichnisses.
   (b) Die Identität der Bank.
   (c) Der öffentliche Schlüssel.
   (d) Die Gültigkeit des Schlüssels:
      i. Der Zeitpunkt, ab dem der Schlüssel gültig ist.
      ii. Der Zeitpunkt, ab dem der Schlüssel für das ABHEBEN ungültig ist.
      iii. Der Zeitpunkt, ab dem der Schlüssel für das BE-/EINZAHLEN ungültig ist.

iv. Der Zeitpunkt, ab dem der Schlüssel für das ZURÜCKGEBEN ungültig ist.
v. Der Zeitpunkt, ab dem der Schlüssel für das ÜBERPRÜFEN ungültig ist.

(e) Die Nutzbarkeit des Schlüssels:
i. Die Kennzeichnung für blinde Signatur.
ii. Der Wert der Münze.

2. Die Signatur des Verzeichnisses über die zu signierenden Daten.

Aus Gründen der Übersichtlichkeit verwenden wir in unserer Realisierung nur einen einzigen Signaturschlüssel bzw. Münzwert.

Abheben mit Münz- und Kundenverfolgung

Bevor ein Kunde neue Münzen von seinem Konto abheben kann, muß sich der Kunde gegenüber der Bank authentisieren. Nach der erfolgreichen Authentisierung des Kunden speichert die Bank den Zeitpunkt der Authentisierung sowie die Identität des Kunden in der Sitzungstabelle ab und erhält eine eindeutige Sitzungsnummer von der Datenbank. Die Bank wählt eine neue, eindeutige Sitzungsmarkierung und speichert die Sitzungsmarkierung in der Markierungstabelle unter der Sitzungsnummer ab. Anschließend interagieren Kunde und Bank in folgendem Abhebeprotokoll.

Protokoll 9. Abhebeprotokoll für Münzen mit möglicher Münz- und Kundenverfolgung.

1. Die Bank und der Kunde erzeugen Münzen. Für jede abzuhebende Münze wählt der Kunde ein neues, zufälliges Schlüsselpaar (Münzschlüssel) und einen geheimen Rückgabeschlüssel. Der Kunde berechnet die Seriennummer (s. Abschnitt 2.2.1). Die Münzen sind die von der Bank blind signierten Seriennummern. Wir nennen die Sicht der Bank auf die Münzen *blinde Münzen*.

   Ist eine Münze nicht korrekt signiert, wird das Abheben abgebrochen.

2. Der Kunde speichert die Münzen in der Münzliste ab und erstellt eine *Abhebeauthorisierung* mit folgender Struktur:

   (a) Die Liste der blinden Münzen.
   (b) Die Signatur des Kunden über die Liste der blinden Münzen.
   (c) Das Zertifikat des Kunden.

   Der Kunde sendet die Signatur der Abhebeauthorisierung an die Bank.

3. Die Bank rekonstruiert die Abhebeauthorisierung und überprüft Signatur des Kunden.

   Schlägt der Test fehl, wird das Abheben abgebrochen.

   Die Bank speichert alle Münzen in der Abhebetabelle sowie die Abhebeauthorisierung in der Abhebeauthorisierungstabelle jeweils unter der Sitzungsnummer ab.

   Anschließend führt die Bank die Buchung “Kundenkonto an Verrechnungskonto” durch.

4. Die Bank und der Kunde erzeugen für jede ausgestellte Münze **drei** Tags. Die drei Tags enthalten jeweils folgende Markierungen:

   - In das *Indextag* wird die Null- oder Einsmarkierung eingebettet, entsprechend einem von der Bank zufällig und gleichverteilt gewählten Bit i.
   - Das *Markierungstag* wird für Münzverfolgung verwendet:
     - Sollen die Münzen des Kunden *nicht* markiert werden, bettet die Bank die Standardmarkierung in das Markierungstag ein.
     - Sollen die Münzen des Kunden markiert werden, bettet die Bank die Sitzungsmarkierung in das Markierungstag ein.
   - Das *Identitätstag* wird für Kundenverfolgung verwendet. In das Identitätstag wird immer die Sitzungsmarkierung eingebettet.

   Die Bank sendet die drei Tags an den Kunden, wobei sie deren Reihenfolge permutiert:

   - Für $i = 0$ sendet die Bank die Permutation (Indextag, Markierungstag, Identitätstag).
   - Für $i = 1$ sendet die Bank die Permutation (Indextag, Identitätstag, Markierungstag).

   Wir nennen die Sicht der Bank auf die Tags *blinde Tags*. Da der Kunde zwischen Markierungs- und Identitätstag nicht unterscheiden kann, nennen wir die Tags aus Sicht des Kunden *linkes Tag* und *rechtes Tag*.

5. Die Bank erstellt ein *Abhebezertifikat* mit folgender Struktur:

   (a) Zu signierende Daten:
      i. Die Identität des Kunden.
      ii. Die Münzgeneration der ausgestellten Münzen.
      iii. Die blinden Münzen und die blinden Tags.

   (b) Die Signatur der Bank über die zu signierenden Daten.

   (c) Das Zertifikat der Bank.

   Die Bank sendet die Signatur des Abhebezertifikates an den Kunden.

6. Der Kunde rekonstruiert das Abhebezertifikat und überprüft die Signatur der Bank.

   Schlägt der Test fehl, gibt der Kunde alle abgehobenen Münzen an die Bank zurück (s. Zurückgeben im vorhergehenden Abschnitt).

   Der Kunde trägt die Tags in der Münzliste nach und speichert das Abhebezertifikat in der Abhebeliste ab.

Bezahlen und Einzahlen mit Münz- und Kundenverfolgung

Bevor eine Zahlung durchgeführt wird, sendet der Händler ein Angebot an den Kunden. Das Angebot hat folgende Struktur:

1. Zu signierende Daten:
   (a) Die Bestellnummer.
   (b) Die Liste der bestellten Waren und Preise.
2. Die Signatur des Händlers über die zu signierenden Daten.
3. Das Zertifikat des Händlers.

Der Kunde überprüft, ob das Angebot mit den von ihm bestellten Waren übereinstimmt und ob die Signatur korrekt ist. Nimmt der Kunde das Angebot an, interagieren Kunde, Händler und Bank im folgenden Zahlungsprotokoll. Bevor Bank und Händler die Einzahlung starten, authentisiert sich der Händler gegenüber der Bank. Nach der erfolgreichen Authentisierung des Händlers speichert die Bank den Zeitpunkt der Authentisierung sowie die Identität des Händlers in der Sitzungstabelle ab und erhält eine eindeutige Sitzungsnummer von der Datenbank.

Protokoll 10. Zahlungsprotokoll mit möglicher Münz- und Kundenverfolgung.

1. Der Kunde speichert das Angebot in der Zahlungsliste ab und wählt die Münzen aus, die für die Zahlung verwendet werden sollen. Der Kunde erstellt aus dem Angebot und den Münzen eine Annahmeerklärung und sendet sie an den Händler. Die Annahmeerklärung hat folgende Struktur:
   (a) Zu signierende Daten:
      i. Die Identität des Händlers.
      ii. Die Bestellnummer aus dem Angebot.
   (b) Die Liste der mit den privaten Münzschlüsseln erstellten Signaturen.
   (c) Die Liste der zu verwendenden Münzen und Indextags.
2. Der Händler überprüft die Annahmeerklärung:
   (a) Die Identität des Händlers muß korrekt sein.
   (b) Die Bestellnummer beim Händler muß vorhanden sein.
   (c) Der Betrag der Münzen muß dem zu zahlenden Betrag entsprechen.

   Schlägt einer der Tests fehl, bricht der Händler die Zahlung ab.

   Der Händler startet die Einzahlung und sendet die Annahmeerklärung an die Bank.

3. Die Bank überprüft die Einzahlung:

   - Die Bank überprüft die Annahmeerklärung:
     - Die Identität des Händlers muß mit der Identität des Händlers in der Annahmeerklärung übereinstimmen.
     - Die Münzsignaturen der Annahmeerklärung müssen mit den öffentlichen Münzschlüsseln verifizierbar sein.
   - Die Bank überprüft die Münzen:
     - Der Einzahlzeitpunkt muß jeweils in der Akzeptanzphase der verwendeten Münzen liegen.
     - Die Seriennummern der Münzen dürfen nicht in der Einzahltabelle enthalten sein.
     - Die Signaturen der Münzen müssen gültig sein.

   Schlägt einer der Tests fehl, wird die Einzahlung abgelehnt.

   Die Bank speichert alle Münzen und Münzsignaturen in der Einzahltabelle sowie die Annahmeerklärung in der Annahmeerklärungstabelle jeweils unter der Sitzungsnummer ab. Die Markierung ist noch nicht bekannt und wird später in der Einzahltabelle für jede Münze nachgetragen (s.u.).

   Die Münzen sind jetzt ungültig. Wird die Einzahlung im folgenden abgebrochen, muß sie später fortgesetzt werden!

   Anschließend extrahiert die Bank die Markierung aus den Indextags:

   - Das Indextag enthält die Nullmarkierung: Das linke Tag ist das Markierungstag, das rechte Tag ist das Identitätstag.
     - Soll Kundenverfolgung bei diesem Händler in der jeweiligen Generation der Münzen durchgeführt werden, verlangt die Bank das rechte Tag vom Kunden.
     - Andernfalls verlangt die Bank das linke Tag vom Kunden.
   - Das Indextag enthält die Einsmarkierung: Das linke Tag ist das Identitätstag, das rechte Tag ist das Markierungstag.
     - Soll Kundenverfolgung bei diesem Händler in der jeweiligen Generation der Münzen durchgeführt werden, verlangt die Bank das linke Tag vom Kunden.
     - Andernfalls verlangt die Bank das rechte Tag vom Kunden.

     Enthält das Indextag einen ungültigen Wert, wird die Einzahlung abgelehnt.

   Die Bank bildet einen Auswahl-Bitstring, wobei jedes Bit angibt, ob die Bank das linke (0) oder rechte (1) Tag der Münze sehen möchte.

   Die Bank erstellt ein *Einzahlzertifikat* mit folgender Struktur:

(a) Zu signierende Daten:

i. Die Identität des Händlers.
ii. Der Auswahl-Bitstring.
iii. Die eingezahlten Münzen und Indextags.

(b) Die Signatur der Bank über die zu signierenden Daten.

(c) Das Zertifikat der Bank.

Die Bank sendet die Signatur des Einzahlzertifikates und den Auswahl-Bitstring an den Händler.

4. Der Händler sendet die Signatur des Einzahlzertifikates und den Auswahl-Bitstring an den Kunden.

5. Der Kunde rekonstruiert das Einzahlzertifikat und überprüft die Signatur der Bank.

Schlägt der Test fehl, wird die Zahlung abgebrochen.

Der Kunde speichert zu jeder Münze das Auswahlbit in der Münzliste sowie das Einzahlzertifikat in der Zahlungsliste ab. Anschließend *sendet* der Kunde je nach Wahl der Bank jeweils das linke oder rechte Tag an den Händler.

6. Der Händler sendet die Tags an die Bank.

7. Die Bank extrahiert die Markierungen aus den erhaltenen Tags:

- Hat die Bank Kundenverfolgung beim Bezahlen durchgeführt, müssen **alle** extrahierten Markierungen der verfolgten Münzgenerationen gültige Sitzungsmarkierungen sein. Die Bank ermittelt die zugehörigen Kundenidentitäten aus der Markierungs- und der Sitzungstabelle.
- Andernfalls kann die Bank Münzverfolgung beim Abheben durchgeführt haben:
  - Enthält ein Tag die Standardmarkierung, ist die Münze anonym.
  - Andernfalls muß die Markierung eine Sitzungsmarkierung sein und die Bank ermittelt die zugehörige Kundenidentität aus Markierungs- und Sitzungstabelle.

Enthält ein Tag eine ungültige Markierung oder eine Markierung aus einer Erpressung, wird die Einzahlung abgelehnt.

Die extrahierten Markierungen werden in der Einzahltabelle nachgetragen (s.o.).

Die Bank führt die Buchung “Verrechnungskonto an Händlerkonto” durch und informiert den Händler über die erfolgreiche Einzahlung.

8. Der Händler liefert die Waren an den Kunden.

Anmerkungen:

1. Wird eine Zahlung abgebrochen, nachdem die Bank die Seriennummern in der Einzahltabelle gespeichert hat, muß die Zahlung später fortgesetzt werden. An der Fortsetzung der Zahlung muß der Händler nicht beteiligt sein.

2. Eine Zahlung muß in der Akzeptanzphase der verwendeten Münzen beginnen. Der Zahlungsvorgang muß innerhalb der Verfolgungsphase abgeschlossen werden. Dies gilt auch für abgebrochene und später fortgesetzte Zahlungen.

3. Wird eine Zahlung von der Bank unberechtigt (aus Sicht des Kunden) abgebrochen, kann die Zahlung erneut gestartet werden. In diesem Fall nimmt ein Richter anstelle des Händlers am Zahlungsprotokoll teil. Führt die Bank dennoch das Zahlungsprotokoll falsch durch, kann die Bank dafür belangt werden.

4. Lehnt die Bank eine Zahlung aufgrund einer Markierung ab, muß die Bank ein *Ablehnungszertifikat* ausstellen. Das Ablehnungszertifikat hat folgende Struktur:

   (a) Zu signierende Daten:

      i. Die Liste der abgelehnten Münzen.
      ii. Die Liste der abgelehnten Tags.

   (b) Die Signatur der Bank über die zu signierenden Daten.

   (c) Das Zertifikat der Bank.

Überprüfen

Mit Beginn der Überprüfungsphase publiziert die Bank die Standardmarkierung, die Nullmarkierung, die Einsmarkierung und die drei geheimen Markierungsschlüssel, mit denen die Markierungen aus den Tags extrahiert werden können. Für jede Münze der entsprechenden Münzgeneration führt der Kunde folgende Schritte durch:

1. Der Kunde lernt die Zuordnung von Markierungs- und Identitätstag zum linken und rechten Tag:

   Der Kunde extrahiert die Markierung aus dem Indextag.

   Enthält das Indextag nicht die Null- oder die Einsmarkierung, liegt eine unübliche Form von Münzverfolgung vor. Dieses hat die gleichen Konsequenzen wie die reguläre Münzverfolgung.

2. Der Kunde überprüft Münzverfolgung:

   - Enthält das Markierungstag die Standardmarkierung, liegt keine Münzverfolgung vor.

- Andernfalls wurde die Münze deanonymisiert und die Bank muß ein Deanonymisierungszertifikat vorweisen können.

Wenn Münzverfolgung durchgeführt wurde und die Bank kein Deanonymisierungszertifikat für den Kunden vorlegen kann, dann interagiert der Kunde mit einem Richter in folgendem Überprüfungsprotokoll für illegale Münzverfolgung.

3. Der Kunde überprüft Kundenverfolgung:

   - Entspricht das bei der Zahlung erhaltene Auswahlbit dem Inhalt des Indextags, liegt keine Kundenverfolgung vor.
   - Andernfalls wurde die Zahlung deanonymisiert und die Bank muß ein Deanonymisierungszertifikat vorweisen können.

   Wenn Kundenverfolgung durchgeführt wurde und die Bank kein Deanonymisierungszertifikat für den Händler vorlegen kann, dann interagiert der Kunde mit einem Richter in folgendem Überprüfungsprotokoll für illegale Kundenverfolgung.

4. Der Kunde überprüft abgelehnte Münzen:

   Für jeden Zahlungsvorgang, den die Bank aufgrund einer entdeckten Markierung abgelehnt hat, überprüft der Kunde das Ablehnungszertifikat. Hat der Kunde keine markierte Münze entdeckt, interagiert der Kunde mit einem Richter in folgendem Überprüfungsprotokoll für illegal abgelehnte Münzen.

Protokoll 11. Überprüfungsprotokoll bei illegaler Münzverfolgung.

1. Der Kunde sendet das Abhebezertifikat (s. Abhebeprotokoll) des Abhebevorgangs, in dem Münzen illegal deanonymisiert wurden, an den Richter.

2. Der Richter überprüft das Abhebezertifikat:

   - Es darf kein Deanonymisierungszertifikat für den Kunden in der Münzgeneration vorliegen.
   - Die Signatur des Abhebezertifikates muß gültig sein.
   - Mindestens ein blindes Markierungstag enthält *nicht* die Standardmarkierung oder mindestens ein blindes Indextag enthält nicht die Null- oder Einsmarkierung.

   Schlägt einer der Tests fehl, wird die Überprüfung abgebrochen.

   Der Richter stellt eine illegale Deanonymisierung fest und informiert den Kunden über die erfolgreiche Anzeige.

Protokoll 12. Überprüfungsprotokoll bei illegaler Kundenverfolgung.

1. Der Kunde sendet das Einzahlzertifikat (s. Zahlungsprotokoll) des Zahlungsvorgangs, in dem der Kunde illegal deanonymisiert wurden, an den Richter.
2. Der Richter überprüft das Einzahlzertifikat:
   - Es darf kein Deanonymisierungszertifikat für den Händler in der Münzgeneration vorliegen.
   - Die Signatur des Einzahlzertifikates muß gültig sein.
   - Mindestens ein Bit des Auswahl-Bitstrings entspricht nicht der extrahierten Null- bzw. Einsmarkierung eines Indextags.

   Schlägt einer der Tests fehl, wird die Überprüfung abgebrochen.

   Der Richter stellt eine illegale Deanonymisierung fest und informiert den Kunden über die erfolgreiche Anzeige.

Analog zur Überprüfung von Kundenverfolgung durch den Kunden kann der Händler ebenso Kundenverfolgung überprüfen. Der Händler kann jedoch nicht bei einer festgestellten Kundenverfolgung auf die Identität des Kunden schließen. Der Händler kann lediglich eine Sitzungsmarkierung aus einem Tag extrahieren. Nur in Verbindung mit der Bank und ihrer Sitzungstabelle kann aus einer Sitzungsmarkierung die Kundenidentität ermittelt werden.

Protokoll 13. Überprüfungsprotokoll für illegal abgelehnte Münzen.

1. Der Kunde sendet das Ablehnungszertifikat (s. Abhebeprotokoll) des Abhebevorgangs, in dem Münzen illegal abgelehnt wurden, an den Richter.
2. Der Richter überprüft das Ablehnungszertifikat:
   - Die Signatur des Ablehnungszertifikates muß gültig sein.
   - Alle Indextags müssen die Null- oder Einsmarkierung enthalten.
   - Alle Markierungstags müssen die Standardmarkierung enthalten.

   Schlägt einer der Tests fehl, wird die Überprüfung abgebrochen.

   Der Richter informiert den Kunden über die erfolgreiche Anzeige einer illegalen Ablehnung.

2.3 Überprüfung der Eigenschaften

Wir überprüfen die Eigenschaften unseres Zahlungssystems mit den geforderten Eigenschaften aus Abschnitt 1.1.3 und den geforderten Eigenschaften für faire Deanonymisierung aus Abschnitt 2.1.3.

Legal abgehobene Münzen sind immer gültig: Der Abhebevorgang gliedert sich in vier Teile:

1. Kunde und Bank erstellen die Münzen.
2. Der Kunde autorisiert die Abbuchung von seinem Konto.
3. Kunde und Bank erstellen die zugehörigen Tags.
4. Die Bank erstellt ein Abhebezertifikat.

Durch diese Teilung kann der Kunde zuerst die Münzen überprüfen. Bemerkt der Kunde ungültig signierte Münzen beim Verifizieren der blinden Signaturen, bricht er den Abhebevorgang ab. In diesem Fall darf die Bank das Konto des Kunden nicht belasten. Diese Trennung ist möglich, da die Münze alleine für den Kunden wertlos ist. Er kann die Münze weder zum Bezahlen noch zum Zurückgeben verwenden:

- Zum Bezahlen und Einzahlen benötigt der Kunde das bzw. die zugehörigen Tags. Der Kunde kann alleine keine gültigen Tags erzeugen.
- Zum Zurückgeben muß die Münze in der Abhebetabelle der Bank gespeichert sein. Die Bank speichert keine Münzen in der Abhebetabelle, für die sie keine Abhebeauthorisierung erhalten hat.

Schließlich kann die Bank eine legal abgehobene Münze aufgrund einer möglichen Markierung ablehnen. In diesem Fall muß die Bank ein Ablehnungszertifikat ausstellen. In der Überprüfungsphase stellt der Kunde fest, daß die Münze entweder unmarkiert ist oder eine Markierung beim Abheben illegal angebracht wurde. Lehnt die Bank eine unmarkierte Münze ab, kann das über das Ablehnungszertifikat nachgewiesen werden. Hat die Bank eine illegale Münzverfolgung durchgeführt, ist sie nicht im Besitz eines Deanonymisierungszertifikates. In beiden Fällen kann die Bank dafür belangt werden.

Der Kunde ist beim Bezahlen gegenüber der Bank anonym: Um einen Kunden beim Bezahlen zu identifizieren, muß die Bank aus den vom Händler eingezahlten Münzen auf die abgehobenen Münzen (blinde Münzen) eines Kunden schließen können. Das ist aufgrund der *Blindheit* der blinden Signatur (s. Abschnitt 1.2.1) nicht möglich. Allerdings kann die Anonymität eines Kunden beim Abheben durch Münzverfolgung oder beim Bezahlen durch Kundenverfolgung aufgehoben werden. Diese Möglichkeit ist nur in Verbindung mit einem Deanonymisierungszertifikat erlaubt und kann vom Kunden nachträglich kontrolliert werden.

Zusätzlich müssen wir zeigen, daß die Bank aus dem Authentisierungscode der Seriennummer nichts über die blinde Münze lernen kann. Der Authentisierungscode erlaubt eine Zuordnung von Münze und blinder Münze genau dann, wenn der Rückgabeschlüssel bekannt ist. Angenommen, es existiert ein Algorithmus zur Berechnung des Rückgabeschlüssels aus einem gegebenen

Authentisierungscode und dem Blendparameter. Dann kann die Bank überprüfen, ob ein Rückgabeschlüssel zu einem berechneten Blendparameter existiert. Unter der Voraussetzung, daß die Menge der möglichen Blendparameter größer ist, als die Menge der möglichen Authentisierungscodes, kann mit hoher Wahrscheinlichkeit für jedes Paar aus einer Münze und einer blinder Münze ein gültiger Rückgabeschlüssel berechnet werden. Folglich kann in keinem Fall eine Aussage über die Zusammengehörigkeit von Münze und blinder Münze getroffen werden.

Der Händler muß die vom Kunden erhaltenen Münzen einzahlen: Der Händler kann die vom Kunden erhaltenen Münzen weder zum Bezahlen verwenden noch als seine eigenen Münzen an die Bank zurückgeben:

- Zum Bezahlen (eines anderen Händlers) muß der Händler eine neue Annahmeerklärung mit den vom Kunden erhaltenen Münzen signieren.
- Zum Zurückgeben muß der Händler für jede Münze eine Rückgabesignatur erstellen.

In beiden Fällen benötigt der Händler die privaten Münzschlüssel, die er jedoch nicht vom Kunden erhalten hat. Er kann die Münzen also nur zum Einzahlen verwenden. Erhält der Händler den privaten Münzschlüssel vom Kunden, handelt es sich um Geldwäsche (s. Abschnitt 1.3.2). Die Behandlung von Geldwäsche vertiefen wir in Abschnitt 2.4.

Zwischen Kunde und Händler kommt ein Kaufvertrag zustande: Der Händler hat dem Kunden ein Angebot unterbreitet. Durch die Annahmeerklärung des Kunden wird das Angebot in dem Moment angenommen, in dem die Bank die Münzen akzeptiert hat (s. Abschnitt 1.1.4). Der daraus resultierende Kaufvertrag kann von einem Richter überprüft werden:

- Der Kunde präsentiert das Angebot des Händlers. Da das Angebot vom Händler signiert wurde, kann das Angebot nicht abgestritten werden.
- Die Bank präsentiert die Annahmeerklärung und die Münzen des Kunden. Da die Annahmeerklärung des Kunden mit den privaten Münzschlüsseln signiert wurde, kann die Annahme des Angebots nicht abgestritten werden.

Da die Bank den Betrag der Münzen dem Konto des Händlers gutgeschrieben hat, muß der Händler die im Angebot vereinbarten Leistungen erfüllen.

Gültige Münzen werden beim Einzahlen von der Bank immer akzeptiert: Bestreitet die Bank die Gültigkeit einer eingezahlten Münze, kann die Zahlung wiederholt werden. An der wiederholten Zahlung kann ein Richter an der Stelle des Händlers teilnehmen. Die Bank darf eine eingezahlte Münze aus folgenden Gründen ablehnen:

- Die Münze ist ungültig signiert.
- Die Annahmeerklärung ist ungültig signiert.
- Die Akzeptanzphase der Münzgeneration ist vorbei.
- Die Münze wurde bereits verwendet, und die Bank muß eine gültige Annahmeerklärung bzw. Rückgabesignatur für diese Münze präsentieren können.
- Die Münze ist markiert.

Mit Ausnahme von markierten Münzen kann der Richter in allen Fällen leicht überprüfen, ob eine Münze zurecht von der Bank abgelehnt wurde. Stellt die Bank eine markierte Münze fest, kann der Richter zunächst nur überprüfen, ob das von der Bank ausgestellte Ablehnungszertifikat gültig ist. Erst in der Überprüfungsphase lernt der Kunde, ob die Münze tatsächlich markiert ist. Bei einer abgelehnten unmarkierten Münze kann der Kunde dann über das Ablehnungszertifikat nachweisen, daß die Bank eine unmarkierte Münze zurückgewiesen hat.

Gültige Münzen werden beim Zurückgeben von der Bank genau dann akzeptiert, wenn sie vom gleichen Kunden abgehoben wurden: Bestreitet die Bank die Gültigkeit einer zurückgegebenen Münze, muß ein Richter die Rückgabe im Auftrag des Kunden durchführen. Die Bank kann eine zurückgegebene Münze aus folgenden Gründen ablehnen:

- Der Authentisierungscode der blinden Münze ist ungültig.
- Die Rückgabesignatur ist ungültig.
- Die Rückgabephase der Münzgeneration ist vorbei.
- Die Münze wurde bereits verwendet, und die Bank muß eine gültige Annahmeerklärung bzw. Rückgabesignatur für diese Münze präsentieren können.
- Die blinde Münze wurde von einem anderen Kunden abgehoben, und die Bank muß eine gültige Abhebeauthorisierung eines anderen Kunden für diese blinde Münze präsentieren können.

In allen Fällen kann der Richter leicht überprüfen, ob die Münze zurecht von der Bank abgelehnt wurden.

Es ist nicht möglich, daß ein Kunde die von einem anderen Kunden abgehobenen “fremden” Münzen zurückgibt. In diesem Fall muß der Kunde die fremden Münzen auf eigene blinde Münzen abbilden. Wir zeigen, daß diese Abbildung unmöglich ist:

Gegeben sei eine Münze, eine blinde Münze und der *berechnete* Blendparameter, der die blinde Münze auf die Münze abbildet. Um die fremde Münze zurückzugeben, muß der Kunde einen

Rückgabeschlüssel finden, der den Blendparameter auf den Authentisierungscode der Seriennummer der Münze abbildet. Die Unfälschbarkeit des Authentisierungscodes impliziert, daß sich aus einem Paar von Blendparameter und Authentisierungscode der Rückgabeschlüssel nicht algorithmisch berechnen läßt.

Von der Bank einmal akzeptierte Münzen sind anschließend ungültig: Nachdem die Bank eine gültige Münze akzeptiert hat, wird die Seriennummer der Münze in die Einzahltabelle eingetragen und ist damit ungültig. Die Bank kann beweisen, daß sie eine Münze akzeptiert hat, indem sie eine Annahmeerklärung bzw. Rückgabesignatur präsentiert, die mit dem öffentlichen Münzschlüssel verifiziert werden kann.

Ungültige Münzen werden von der Bank nie akzeptiert: Eine formal gültige Münze ist in folgenden Fällen ungültig:

- Die Münze wurde bereits benutzt. In diesem Fall ist die Seriennummer der Münze bereits in der Einzahltabelle vorhanden und die Bank wird diese Münze nicht akzeptieren.
- Die Münze wurde nicht legal abgehoben.
  - Die Münze wurde gefälscht. Um ein Münze zu fälschen, muß die Signatur der Seriennummer gefälscht werden. Das Fälschen einer (blinden) Signatur ist aufgrund der *Unfälschbarkeit* der blinden Signatur nicht möglich (s. Abschnitt 1.2.1).
  - Die Münze resultiert aus einer Erpressung oder aus einem Bankraub. Diese Fälle behandeln wir ausführlich in Abschnitt 2.4. Die Münzen werden nicht akzeptiert.

Die Anonymität eines Kunden kann grundsätzlich aufgehoben werden: Um Münz- oder Kundenverfolgung zu umgehen, müssen die Tags der Münzen ausgetauscht oder manipuliert werden. Beides resultiert mit überwältigender Wahrscheinlichkeit in Tags mit ungültiger Markierung (Eigenschaften des Tags, s. Abschnitt 2.2.2). Ein Tag mit ungültiger Markierung wird von der Bank nicht angenommen. Daher wird die Zahlung nicht ausgeführt und die Deanonymisierung kann nicht umgangen werden.

Ein Kunde kann nur beim Abheben oder beim Bezahlen deanonymisiert werden: Deanonymisierungen erfolgen ausschließlich durch Markierungen. Markierungen können nur beim Abheben angebracht werden. Erhält die Bank beim Bezahlen ein Tag mit Standardmarkierung, ist die Münze garantiert anonym. Nachträgliche Deanonymisierungen sind nicht möglich. Es existiert jedoch eine Ausnahme: Der Kunde kann gezwungen werden, sich “freiwillig” durch Ausführen des Rückgabeprotokolls zu deanonymisieren (s. Abschnitt 1.4.3*)*. Der Kunde kann unfreiwillige Deanonymisierungen umgehen, indem er die Authentisierungscodes falsch berechnet. In diesem Fall kann er die Münzen jedoch nicht zurückgeben. Alternativ kann der Kunde auch einfach den Rückgabeschlüssel und den verwendeten Blendparameter nach erfolgter Zahlung löschen. Dann gibt es keine Möglichkeit mehr, eine unfreiwillige Deanonymisierung durchzuführen.

Der Kunde kann Deanonymisierungen seiner Zahlungen kontrollieren: Es gibt zwei Möglichkeiten, wie die Bank entdeckbare Deanonymisierungen umgehen könnte:

- Die Bank publiziert einen falschen Markierungsschlüssel: Die Bank publiziert am Anfang der Überprüfungsphase für das Index-, Markierungs- und Identitätstag jeweils genau einen Markierungsschlüssel. Publiziert die Bank einen falschen Markierungsschlüssel, werden (fast) alle Kunden markierte Münzen feststellen.
- Die Bank verbirgt Informationen in der Permutation von Markierungs- und Identitätstag: Ein Kunde erhält z.B. bei 90% der abgehobenen Münzen immer die Zuordnung "linkes Tag ist Markierungstag". Werden beim Bezahlen Münzen verwendet, bei denen zu etwa 90% diese Zuordnung verwendet wird, schließt die Bank daraus, daß es sich um genau diesen Kunden handelt. Dieser Deanonymisierungsmechanismus ist zwar sehr ungenau und kann nur für eine sehr geringe Anzahl an Kunden verwendet werden. Trotzdem müssen wir diesen illegalen Deanonymisierungsmechanismus verhindern. Daher zwingen wir die Bank, die Permutation nicht zufällig zu wählen, sondern *überprüfbar zufällig*. In der Überprüfungsphase muß die Bank dann publizieren, wie die Berechnung der Permutation erfolgt ist.

Da Deanonymisierungen immer entdeckt werden können, kann die Legalität einer Deanonymisierung überprüft werden. Aufgrund des Abhebezertifikates kann die Bank eine entdeckte illegale Deanonymisierung nicht abstreiten. Unter der Annahme, daß die Strafe für illegale Deanonymisierungen hoch genug ist, z.B. Entzug der Banklizenz, wird keine Bank daran interessiert sein, illegale Deanonymisierungen durchzuführen.

2.4 Lösung anonymitätsbezogener Probleme

In diesem Abschnitt möchten wir zeigen, wie die anonymitätsbezogenen Probleme aus Abschnitt 1.3 mit unserem Zahlungssystem gelöst werden können.

2.4.1 Erpressung

Im Falle einer Erpressung soll die Bank die Münzen des Kunden markieren, so daß diese beim Einzahlen wiedererkannt werden. Sobald der Kunde der Bank mitteilt, daß der Grund der Erpressung hinfällig ist, können alle unbenutzten, markierten Münzen des Kunden invalidiert werden. Diese Münzen werden dann nicht mehr von der Bank akzeptiert und der Betrag der invalidierten Münzen kann dem Kunden unmittelbar gutgeschrieben werden. Damit die Bank die Münzen des erpreßten Kunden markieren darf, benötigt sie ein Deanonymisierungszertifikat. Der Kunde kann dieses Zertifikat entweder selbst ausstellen, oder aber der Kunde schaltet die Strafverfolgung ein, die ein solches Zertifikat von einem Richter ausstellen läßt.

Wenn sich der Kunde selbst in der Gewalt des Erpressers befindet, nennen wir das *Entführung*. Entführungen sind ein stärkerer Angriff als Erpressungen, da der Kunde die Bank nun nicht mehr direkt über eine Erpressung informieren bzw. ein Deanonymisierungszertifikat ausstellen kann. Um den Kunden vor solchen Angriffen zu schützen, schlagen wir folgende Vorgehensweisen vor:

- Der Kunde kann eine *Deanonymisierungsvereinbarung* bei der Bank hinterlegen. Diese Vereinbarung gibt an, wann die Bank Münzen beim Abheben markieren soll. Die Vereinbarung muß vom Kunden signiert sein und darf nur sehr restriktiv geändert werden. Änderungen sind z.B. nur möglich, wenn der Kunde persönlich bei der Bank erscheint oder Änderungen treten erst nach längerer Zeit in Kraft. Dadurch soll verhindert werden, daß ein Entführer den Kunden zwingt, die Deanonymisierungsvereinbarung außer Kraft zu setzten.

  Wir geben ein Beispiel für eine Deanonymisierungsvereinbarung:

  > *Wenn mehr als €1.000 pro Tag oder mehr als €5.000 pro Woche abgehoben werden, sollen weitere abgehobene Münzen dieser Generation markiert werden.*

- Der Kunde kann über einen verdeckten Kanal zur Bank unbemerkt ein Deanonymisierungszertifikat ausstellen. Eine Möglichkeit dafür bietet die Authentisierung des Kunden.

  Wir geben ein Beispiel für einen verdeckten Kanal:

  > Der Kunde besitzt mindestens zwei Schlüsselpaare, mit denen er sich gegenüber der Bank authentisieren kann: Das Authentisierungs-Schlüsselpaar und das Notfall-Schlüsselpaar. Der Unterschied zwischen beiden Schlüsselpaaren ist, daß die Bank für den öffentlichen Notfallschlüssel ein von einem Richter ausgestelltes Zertifikat besitzt. Die Authentisierung des Kunden erfolgt dann über ein Challenge-Response Verfahren.
  >
  > 1. Die Bank wählt eine Challenge mit ausreichender Bitlänge und sendet die Challenge an den Kunden.
  > 2. Der Kunde authentisiert sich, indem er die Challenge der Bank, seine Identität und die Generation der aktuellen Abhebephase mit dem Authentisierungs- oder Notfallschlüssel signiert.
  > 3. Die Bank überprüft die Antwort des Kunden: Der Inhalt muß korrekt und die Signatur gültig sein. Hat der Kunden den privaten Notfallschlüssel zur Authentisierung benutzt, besitzt die Bank ein gültiges Deanonymisierungszertifikat und markiert alle folgenden Münzen dieser Münzgeneration.

2.4.2 Geldwäsche

Wird ein Kunde verdächtigt, z.B. illegale Waren zu erwerben, kann die Strafverfolgung ein Deanonymisierungszertifikat beantragen. Ein Richter stellt ein Deanonymisierungszertifikat aus,

das die Bank anweist, die Münzen dieses Kunden zu markieren. Die Bank kann dadurch feststellen, wer die Münzen dieses Kunden einzahlt. Die Bank leitet diese Daten an die Strafverfolgung weiter. Dadurch wird der einzahlende Händler verdächtigt, Geldwäsche durchzuführen. Die Strafverfolgung bewirkt ein weiteres Deanonymisierungszertifikat von einem Richter. Dieses Deanonymisierungszertifikat weist die Bank an, alle weiteren von diesem Händler eingezahlten Münzen ebenfalls zu deanonymisieren. Die Bank leitet die gewonnenen Daten über die Kunden des Händlers an die Strafverfolgung weiter, so daß Ermittlungen gegen die deanonymisierten Kunden eingeleitet werden können. Erhärtet sich schließlich der Verdacht gegen diesen Händler, können weitere Schritte gegen ihn eingeleitet werden.

Es bleibt noch die Frage, ob der für alle Kunden gleichzeitige und zuvor festgelegte Beginn der Überprüfungsphase ein Problem darstellt. Unserer Meinung nach ist das bei geeigneter Wahl der Laufzeiten der einzelnen Phasen einer Generation kein Problem:

- Normalerweise sollte die Zeit der Verfolgungsphase ausreichen, um Mißbrauch von Anonymität aufzuklären. Andernfalls kann der Beginn der Überprüfungsphase auf richterliche Anordnung kurzfristig verschoben werden.
- Mit dem Beginn der Überprüfungsphase können Deanonymisierungen bemerkt werden. Ein Kunde bzw. Händler, der eine (versuchte) Deanonymisierung entdeckt, weiß nicht, ob er in Zukunft wieder deanonymisiert wird.

2.4.3 Bankraub

Beim Bankraub ist der Prozeß der Münzerstellung unter der Kontrolle eines Bankräubers. Daher werden die resultierenden Münzen nicht markiert sein. Markierungen sind daher nicht geeignet, um Bankraub zu verhindern.

Wir verhindern Bankraub, indem wir die geraubten Münzen unbrauchbar machen: Nach einem Bankraub wird die Abhebe- und Akzeptanzphase der betroffenen Münzgeneration unmittelbar beendet. Dies hat zur Folge, daß die Vorgänge ABHEBEN, BEZAHLEN und EINZAHLEN nicht mehr ausgeführt werden dürfen. Für unbenutzte Münzen ist nur noch der Vorgang ZURÜCKGEBEN möglich. Es ist somit nur noch möglich, die unbenutzten Münzen nicht-anonym an die Bank zurückzugeben und neue Münzen der nächsten Generation abzuheben.

Der Bankräuber muß versuchen die gestohlenen Münzen an die Bank zurückzugeben, andernfalls sind sie wertlos. Münzen, die in einem Bankraub “erbeutet” wurden, werden von der Bank nicht in die Abhebetabelle eingefügt. Der Bankräuber muß versuchen, die erbeuteten Münzen auf legal abgehobene und bereits ausgegebene Münzen von anderen Kunden abzubilden. Im Prinzip handelt es sich dabei um Geldwäsche. Für diese Abbildung benötigt der Bankräuber einen passenden Rückgabeschlüssel. Die Unfälschbarkeit des Authentisierungscodes impliziert, daß der Bankräuber keinen passenden Schlüssel berechnen kann. Dadurch sind die erbeuteten Münzen für den Bankräuber wertlos.

Im Falle eines kryptographischen Zusammenbruchs des Signaturverfahrens der Münzen kann ebenso verfahren werden. Für die betroffenen Münzgenerationen wird nur noch der Vorgang ZURÜCKGEBEN erlaubt, neue Münzen werden mit einer anderen, sicheren blinden Signatur ausgestellt. Das ist dadurch möglich, daß für das Zurückgeben einer Münze keine gültige Signatur benötigt wird. Die Voraussetzung für das Zurückgeben ist jedoch, daß der Authentisierungscode nicht *gleichzeitig* mit dem Signaturverfahren gebrochen wurde.

Kapitel 3

Realisierung von FlexiCash

Die Protokolle von FlexiCash aus dem vorhergehenden Kapitel basieren auf Münzen mit Tags. Für Tags haben wir bisher nur die Eigenschaften spezifiziert, aber noch keine Realisierung angegeben. In diesem Kapitel beschreiben wir die Realisierung von FlexiCash auf allgemeinen kryptographischen Primitiven.

Gliederung des Kapitels

In Abschnitt 3.1 beschreiben wir, wie die Realisierung von Münzen mit Tags prinzipiell möglich ist. Anschließend stellen wir in Abschnitt 3.2 eine konkrete Realisierung auf Basis des allgemeinen diskreten Logarithmusproblems vor. Auf der Grundlage dieser Realisierung beschreiben wir die Protokolle der erweiterten Variante von FlexiCash detailliert. Schließlich diskutieren wir in Abschnitt 3.3 mögliche alternative Realisierungen.

3.1 Realisierung von Münzen mit Tags

In diesem Abschnitt zeigen wir, wie Münzen mit Tags realisiert werden können. Dazu führen wir zunächst die blinde Chiffre als neues kryptographisches Primitiv ein und zeigen, wie randomisierte blinden Signaturen mit blinden Chiffren kombiniert werden können.

3.1.1 Blinde Chiffren

Eine blinde Chiffre ist ein Protokoll zwischen zwei Teilnehmern, dem *Aussteller* und dem *Empfänger*. Das Ziel des Protokolls ist es, eine vom Aussteller gewählte, geheime Markierung zu einem Tag zu verschlüsseln, so daß folgende Eigenschaften erfüllt sind.

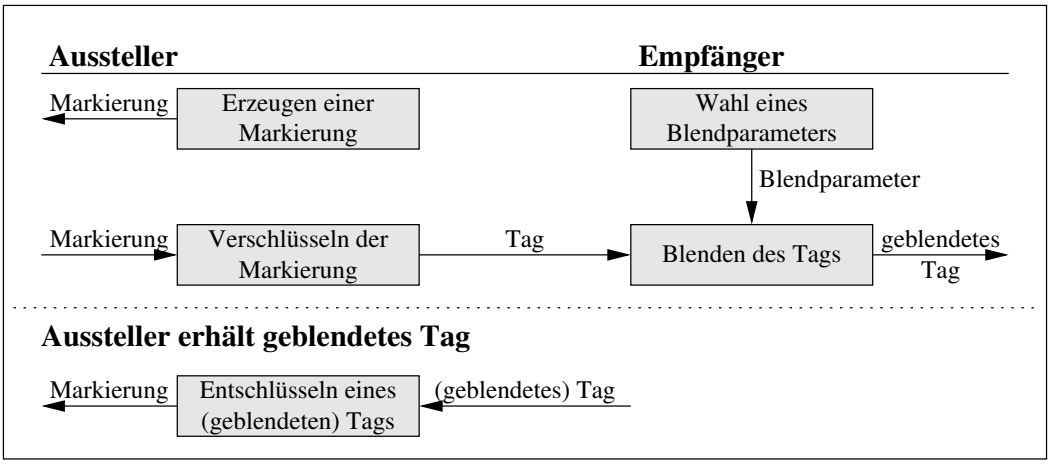

Abbildung 3.1: Ablauf der Operationen einer blinden Chiffre

Eigenschaften

Wir fordern zwei Eigenschaften von einer blinden Chiffre, damit wir sie als sicher bezeichnen: *Ununterscheidbarkeit* und *Blindheit*.

Ununterscheidbarkeit: Der Empfänger kann zwei Tags nicht unterscheiden, auch wenn die verschlüsselten Markierungen von ihm selbst gewählt wurden.

Blindheit: Der Aussteller kann zwei geblendete Tags genau dann unterscheiden, wenn die verschlüsselten Markierungen unterschiedlich sind.

Die formale Definitionen zur Ununterscheidbarkeit von blinden Chiffren ist äquivalent zur Ununterscheidbarkeit von Chiffren [GM82, GM84]. Die formale Definition der Blindheit entspricht der Blindheit von blinden Signaturen [JLO97].

Operationen

Der Aussteller erstellt ein Chiffrierschlüsselpaar. Der private Chiffrierschlüssel wird vom Aussteller geheim gehalten, der öffentliche Chiffrierschlüssel wird publiziert und vom Empfänger verwendet.

Eine blinde Chiffre soll folgende Operationen unterstützen. Der Ablauf der Operationen ist in Abbildung 3.1 dargestellt.

1. **Erzeugen einer Markierung.** Eine zufällige Markierung wird erzeugt. Die Ausgabe ist die erzeugte Markierung.

2. **Verschlüsseln der Markierung.** Die Eingabe ist eine Markierung und der private Chiffrierschlüssel des Ausstellers. Die Markierung wird durch Verschlüsselung in ein Tag transformiert. Die Ausgabe ist das erzeugte Tag.

3. **Wahl eines Blendparameters.** Ein zufälliger Blendparameter wird gewählt. Die Ausgabe ist der Blendparameter.

4. **Blenden des Tags.** Die Eingabe ist ein Tag, der zu verwendende Blendparameter und der öffentliche Chiffrierschlüssel des Ausstellers. Das Tag wird geblendet, so daß die verschlüsselte Markierung nicht verändert wird. Die Ausgabe ist das geblendete Tag.

5. **Entschlüsseln eines (geblendeten) Tags.** Die Eingabe ist ein (geblendetes) Tag und der private Chiffrierschlüssel des Ausstellers. Das Tag wird durch Entschlüsselung in eine Markierung zurücktransformiert. Die Ausgabe ist die entschlüsselte Markierung.

Randomisierung

Blinde Chiffren sind immer randomisiert, da für jede Markierung immer viele Tags existieren müssen, andernfalls könnte ein Tag nicht geblendet werden. Wir unterteilen blinde Chiffren in zwei Gruppen: *öffentlich randomisierte blinde Chiffren* und *geheim randomisierte blinde Chiffren*.

- **Öffentlich randomisierte blinde Chiffren:** Die zum Verschlüsseln einer Markierung verwendete Randomisierung ist auch zum Entschlüsseln notwendig. Die verwendete Randomisierung ist ein Teil des Tags.

- **Geheim randomisierte blinde Chiffren:** Die zum Verschlüsseln einer Markierung verwendete Randomisierung ist geheim und wird zum Entschlüsseln des resultierenden Tags nicht benötigt.

3.1.2 Randomisierte blinde Signaturen

Wir teilen (blinde) Signaturen in zwei Kategorien ein: deterministische und randomisierte Signaturen. Deterministische blinde Signaturen haben wir bereits in Abschnitt 1.2.1 als allgemeine blinde Signaturen beschrieben. Im folgenden beschreiben wir randomisierte blinde Signaturen.

Eigenschaften

Zusätzlich zu den ursprünglichen Eigenschaften *Unfälschbarkeit* und *Blindheit* fordern wir die *Randomisierung*.

Unfälschbarkeit: Der Empfänger darf nicht mehr gültige Signaturen erstellen können, als er vom Aussteller erhalten hat.

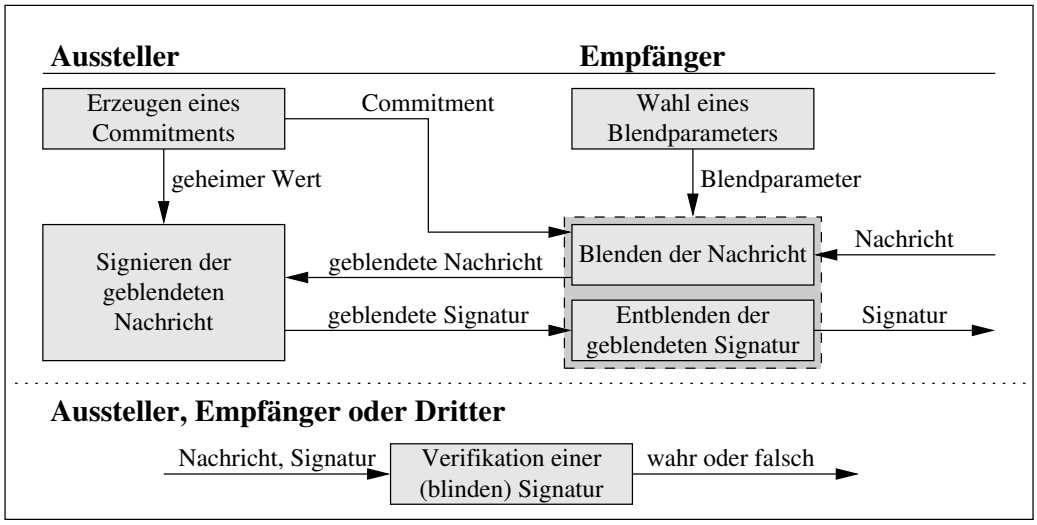

Abbildung 3.2: Ablauf der Operationen einer randomisierten blinden Signatur

Blindheit: Der Aussteller kann nichts aus dem Protokoll lernen, außer daß er eine gültige Signatur ausgestellt hat.

Randomisierung: Zwei blinde Signaturen der gleichen Nachricht resultieren in unterschiedlichen Signaturen.

Operationen

Der Aussteller erstellt ein Schlüsselpaar zum Signieren. Der private Signaturschlüssel wird vom Aussteller geheim gehalten, der öffentliche Signaturschlüssel wird publiziert und vom Empfänger verwendet.

Eine randomisierte blinde Signatur soll folgende Operationen unterstützen. Der Ablauf der Operationen ist in Abbildung 3.2 dargestellt.

1. **Erzeugen eines Commitments.** Ein zufälliger, geheimer Wert wird gewählt und ein Commitment für diesen Wert erzeugt. Die Ausgabe ist das erzeugte Commitment.

2. **Wahl eines Blendparameters.** Ein zufälliger Blendparameter wird gewählt. Die Ausgabe ist der Blendparameter.

3. **Blenden der Nachricht.** Die Eingabe ist die zu signierende Nachricht, der zu verwendende Blendparameter, das Commitment und der öffentliche Signaturschlüssel des Ausstellers. Das Commitment wird geblendet. Aus der Nachricht, dem geblendeten Commitment und dem Blendparameter wird eine geblendete Nachricht erzeugt. Die Ausgabe ist die geblendete Nachricht.

4. **Signieren der geblendeten Nachricht.** Die Eingabe ist die geblendete Nachricht, der geheime Wert des Commitments und der private Signaturschlüssel des Aussteller. Die geblendete Nachricht wird signiert. Die Ausgabe ist die geblendete Signatur.

5. **Entblenden der geblendeten Signatur.** Die Eingabe ist die geblendete Signatur und der Blendparameter, der zum Blenden der Nachricht verwendet wurde. Die geblendete Signatur wird zu einer Signatur der ursprünglichen Nachricht transformiert. Die Ausgabe ist eine reguläre Signatur.

6. **Verifikation einer (blinden) Signatur.** Die Eingabe ist die Nachricht, eine reguläre Signatur und der öffentliche Signaturschlüssel des Ausstellers. Die Signatur wird überprüft. Die Ausgabe ist `wahr`, wenn die Signatur gültig ist und `falsch` sonst.

Randomisierung

Die Randomisierung der blinden Signatur erfolgt durch die Wahl des Commitments des Ausstellers. Benutzt der Aussteller für zwei blinde Signaturen das gleiche Commitment, kann der Empfänger unter Umständen den privaten Schlüssel des Ausstellers berechnen. Der Aussteller muß für jede neue blinde Signatur ein neues Commitment erzeugen. Daher ist es nicht möglich, eine deterministische Signatur zu erzeugen.

Das Commitment ist immer öffentlich. Es ist entweder Teil der Signatur oder kann aus der Signatur erneut berechnet werden.

3.1.3 Kombination von blinder Signatur und blinder Chiffre

Münzen und Tags müssen miteinander verbunden werden können. Münzen erstellen wir durch randomisierte blinde Signaturen, Tags durch blinde Chiffren. Wir benötigen einen Mechanismus, der eine blinde Signatur und eine oder mehrere blinde Chiffren untrennbar miteinander verbindet. Wir realisieren dies, indem wir beide Primitive mit dem gleichen Wert randomisieren.

Wir unterscheiden zwei Vorgehensweisen, je nachdem, ob die blinde Chiffre öffentlich oder geheim randomisiert ist. Bei beiden Varianten erfolgt zunächst das Ausstellen einer Münze über eine blinde Signatur:

Aussteller und Empfänger führen das Protokoll für eine randomisierte blinde Signatur durch:

1. **Erzeugen eines Commitments.** Der Aussteller berechnet ein Commitment und sendet es an den Empfänger.

2. **Wahl eines Blendparameters.** Der Empfänger wählt einen zufälligen Blendparameter.

3. **Blenden der Nachricht.** Der Empfänger blendet das Commitment, erstellt die geblendete Nachricht und sendet sie an den Aussteller.

4. **Signieren der geblendeten Nachricht.** Der Aussteller signiert die geblendete Nachricht und sendet die geblendete Signatur an den Empfänger.

5. **Entblenden der geblendeten Signatur.** Der Empfänger entblendet die Signatur.

Öffentliche Randomisierung

Der Aussteller verwendet das Commitment der blinden Signatur als öffentliche Randomisierung der blinden Chiffre.

Aussteller und Empfänger führen das Protokoll für eine öffentlich randomisierte blinde Chiffre durch:

1. **[Erzeugen einer Markierung.]** Wenn gewünscht, erzeugt der Aussteller eine neue Markierung.

2. **Verschlüsseln der Markierung.** Der Aussteller verwendet das Commitment der blinden Signatur als öffentliche Randomisierung und verschlüsselt die Markierung zu einem Tag. Der Aussteller sendet das Tag an den Empfänger.

3. **Blenden des Tags.** Der Empfänger verwendet den Blendparameter der blinden Signatur, um das Tag zu blenden.

Der Aussteller erhält später eine Münze und ein Tag vom Empfänger:

1. **Verifikation einer (blinden) Signatur.** Der Aussteller überprüft die Signatur und erhält dabei das geblendete Commitment.

2. **Entschlüsseln eines (geblendeten) Tags.** Der Aussteller benutzt das geblendete Commitment als öffentliche Randomisierung und entschlüsselt die Markierung aus dem Tag.

Da als öffentliche Randomisierung das Commitment der blinden Signatur verwendet wird, können wir die öffentliche Randomisierung aus dem Tag entfernen. Damit das Tag korrekt entschlüsselt werden kann, muß das Tag auf die gleiche Weise geblendet werden wie die Signatur.

Geheime Randomisierung

Aus dem Commitment der blinden Signatur leitet der Aussteller einen geheimen Wert ab, den er als geheime Randomisierung der blinden Chiffre verwendet.

Aussteller und Empfänger führen das Protokoll für eine geheim randomisierte blinde Chiffre durch:

1. **[Erzeugen einer Markierung.]** Wenn gewünscht, erzeugt der Aussteller eine neue Markierung.
2. **Verschlüsseln der Markierung.** Der Aussteller verwendet einen aus dem Commitment der blinden Signatur abgeleiteten Wert als geheime Randomisierung und verschlüsselt die Markierung zu einem Tag. Der Aussteller sendet das Tag an den Empfänger.
3. **Blenden des Tags.** Der Empfänger verwendet den Blendparameter der blinden Signatur, um das Tag zu blenden.

Der Aussteller erhält später eine Münze und ein Tag vom Empfänger:

1. **Verifikation einer (blinden) Signatur.** Der Aussteller überprüft die Signatur und erhält dabei das geblendete Commitment.
2. **Entschlüsseln eines (geblendeten) Tags.** Der Aussteller entschlüsselt die Markierung aus dem Tag und extrahiert anschließend die geheime Randomisierung. Der Aussteller leitet aus dem geblendeten Commitment den geheimen Wert ab und vergleicht ihn mit der geheimen Randomisierung des Tags.

3.1.4 Überprüfung der Eigenschaften

Wir überprüfen unsere Realisierung von Tags mit den in Abschnitt 2.2.2 geforderten Eigenschaften:

Ohne Kenntnis des Markierungsschlüssels kann nichts über die in einem Tag eingebettete Markierung ausgesagt werden: Aufgrund der von der blinden Chiffre geforderten *Ununterscheidbarkeit* können Tags ohne Kenntnis des Markierungsschlüssels nicht unterschieden werden.

Tags werden von der Bank blind ausgestellt, so daß die Bank ein Tag später nicht an seiner äußeren Form erkennen kann: Aufgrund der von der blinden Chiffre geforderten *Blindheit* können geblendete Tags nicht an der äußeren Form wiedererkannt werden.

Der Versuch, eine in einem Tag eingebettete Markierung zu manipulieren, resultiert mit hoher Wahrscheinlichkeit in einer ungültigen Markierung: Eine Chiffre, bei der es unmöglich ist, einen verschlüsselten Klartext gezielt zu verändern, nennt man *nichtformbar* [DDN91]. Nichtformbarkeit und Blindheit schließen sich gegenseitig aus. Daher ist es für einen Kunden prinzipiell möglich, die in einem Tag verschlüsselte Markierung zu verändern. Da die von der Bank verwendeten Markierungen dem Kunden nicht bekannt sind, resultiert eine Manipulation einer verschlüsselten Markierung in einer für den Kunden zufälligen Markierung. Die Bank akzeptiert

nur die Markierungen, die in der Markierungstabelle gespeichert sind. Die manipulierte Markierung ist mit hoher Wahrscheinlichkeit nicht in der Markierungstabelle gespeichert und wird nicht von der Bank akzeptiert.

Der Versuch, eine Münze mit dem Tag einer anderen Münze zu benutzen, resultiert mit hoher Wahrscheinlichkeit in einer ungültigen Markierung: Jede Münze verwendet ein anderes, zufällig gewähltes Commitment. Dieses Resultiert aus der *Randomisierung* der blinden Signatur. Wir unterscheiden zwei Fälle, je nachdem, ob die blinde Chiffre öffentlich oder geheim randomisiert ist.

- Das Commitment wird als öffentliche Randomisierung der blinden Chiffre verwendet. Bei einer öffentlich randomisierten Chiffre ist es notwendig, daß die gleiche (geblendete) Randomisierung zum Entschlüsseln verwendet wird, die auch zum Verschlüsseln verwendet wurde. Andernfalls kann das Tag nicht korrekt entschlüsselt werden. Wird eine Markierung mit der falschen Randomisierung entschlüsselt, ist die resultierende Markierung eine für den Kunden zufällige Markierung. Die Bank akzeptiert nur die Markierungen, die in der Markierungstabelle gespeichert sind. Die entschlüsselte Markierung ist mit hoher Wahrscheinlichkeit nicht in der Markierungstabelle gespeichert und wird nicht von der Bank akzeptiert.
- Aus dem Commitment wird ein geheimer Wert abgeleitet, der als geheime Randomisierung der blinden Chiffre verwendet wird. Die Markierung kann immer entschlüsselt werden. Jedoch akzeptiert die Bank eine Markierung nur dann, wenn die geheime Randomisierung der blinden Chiffre aus dem geblendeten Commitment der blinden Signatur abgeleitet werden kann. Da für jede Münze eine andere Randomisierung verwendet wird, lehnt die Bank die Markierung als ungültig ab.

3.2 Realisierung auf dem allgemeinen DL-Problem

Basierend auf einer Kombination aus blinder Schnorr Signatur und blinder ElGamal Chiffre beschreiben wir eine Realisierung der erweiterten Variante von FlexiCash auf dem allgemeinen diskreten Logarithmusproblem.

Als Systemparameter verwenden wir im folgenden eine Gruppe G mit Primzahlordnung q und einen Erzeuger g von G.

3.2.1 Die blinde Schnorr Signatur

Die blinde Variante der Schnorr Signatur [Sch91] verwendet als privaten Signaturschlüssel $x \in_R \mathbb{Z}_q$ und als öffentlichen Signaturschlüssel $y = g^x$. Sei $H()$ eine Hashfunktion, die Nachrichten beliebiger Länge auf Elemente vom $\mathbb{Z}_q$ abbildet.

Protokoll

Das Ausstellen einer blinden Schnorr Signatur erfolgt mit folgendem Protokoll:

1. **Erzeugen eines Commitments.** Der Aussteller wählt einen zufälligen Wert $r \in_R \mathbb{Z}_q$ und berechnet das Commitment $a = g^r$. Das Commitment wird an den Empfänger gesendet.
2. **Wahl eines Blendparameters.** Der Empfänger wählt einen zufälligen Blendparameter $(\alpha, \beta) \in_R \mathbb{Z}_q^2$.
3. **Blenden der Nachricht.** Der Empfänger berechnet das geblendete Commitment $a' = ag^\alpha y^\beta$. Anschließend wählt er die zu signierende Nachricht m und berechnet die Challenge $c' = H(m, a')$. Die geblendete Challenge $c = c' - \beta \bmod q$ wird an den Aussteller gesendet.
4. **Signieren der geblendeten Nachricht.** Der Aussteller antwortet dem Empfänger mit der Response $s = r - cx \bmod q$.
5. **Entblenden der geblendeten Signatur.** Der Empfänger entblendet die Response zu $s' = s + \alpha \bmod q$. Die Signatur ist das Paar (c', s').
6. **Verifikation einer (blinden) Signatur.** Um die Signatur (c', s') einer Nachricht m zu überprüfen, muß zunächst das Commitment $a' = g^{s'} y^{c'}$ aus der blinden Signatur rekonstruiert werden. Die Signatur ist gültig, wenn $c' \stackrel{?}{=} H(m, a')$ ist.

Überprüfung der Eigenschaften

Wir weisen nun die von uns für eine blinde Signatur geforderten Eigenschaften Unfälschbarkeit und Blindheit nach.

Unfälschbarkeit: Die Unfälschbarkeit von blinden Signaturen nachzuweisen ist schwierig. Pointcheval und Stern [PS00] gelingt der Nachweis unter folgenden Bedingungen:

1. Die blinde Signatur muß *Witness Indistinguishable* [FS90] sein. Okamoto [Oka92] beschreibt eine Schnorr-Variante, die Witness Indistinguishable ist.
2. Die Anzahl der ausgestellten blinden Signaturen muß stark eingeschränkt werden. Es dürfen nicht mehr als polylogarithmisch viele blinde Signaturen ausgestellt werden. Pointcheval [Poi98] beschreibt eine Methode auf *Cut-And-Choose* Basis, um diese Beschränkung zu umgehen.

Schnorr [Sch01a,Sch01b] zeigt, daß diese Einschränkungen unnötig sind und beweist, daß blinde Schnorr Signaturen im Random Oracle und generischem Modell unfälschbar sind.

Blindheit: Der Empfänger wählt Nachrichten m_i und erhält vom Aussteller die Signaturen (c'_i, s'_i) mit $0 \leq i \leq 1$. Sei (a_i, c_i, s_i) die geblendete Sicht des Ausstellers auf die Protokolldurchläufe. Der Empfänger wählt $k \in_R \{0,1\}$ und sendet (m_k, c'_k, s'_k) an den Aussteller. Der Aussteller soll nun $l \in \{0,1\}$ so wählen, daß $l = k$ ist.

Die signierte Nachricht (m_k, c'_k, s'_k) kann auf jede geblendete Sicht (a_i, c_i, s_i) abgebildet werden. Der dazu jeweils notwendige Blendparameter ist (α, β) mit $\alpha = s'_k - s_i \bmod q$, $\beta = c'_k - c_i \bmod q$ und es gilt folgende Abbildung:

$$\begin{aligned} a'_k &= a_i g^\alpha y^\beta = g^{s'_k} y^{c'_k} \\ c'_k &= H(m'_k, a'_k) \\ c_i &= c'_k - \beta \bmod q \\ s'_k &= s_i + \alpha \bmod q \end{aligned}$$

Da jede Signatur auf jede geblendete Signatur abgebildet werden kann, ist eine Unterscheidung der Signaturen für den Aussteller nicht möglich.

3.2.2 Die blinde ElGamal Chiffre

Die blinde Variante der ElGamal Chiffre [ElG85] verwendet als privaten Chiffrierschlüssel $x_0 \in_R \mathbb{Z}_q$ und als öffentlichen Chiffrierschlüssel $y_0 = g^{x_0}$.

Protokoll

Das Ausstellen eines blinden ElGamal Chiffretextes erfolgt mit folgendem Protokoll:

1. **Erzeugen einer Markierung.** Der Aussteller wählt eine zufällige Nachricht $m \in_R G$
2. **Verschlüsseln der Markierung.** Der Aussteller wählt einen Wert $r \in_R \mathbb{Z}_q$ und berechnet $a = g^r$. Das Tag ist $(a, t) = (a, a^{x_0} m)$. Das Tag wird an den Empfänger gesendet.
3. **Wahl eines Blendparameters.** Der Empfänger wählt einen zufälligen Blendparameter $\delta \in_R \mathbb{Z}_q$.
4. **Blenden des Tags.** Der Empfänger berechnet das geblendete Tag $(a', t') = (a g^\delta, t y_0^\delta)$.
5. **Entschlüsseln eines (geblendeten) Tags.** Zum Entschlüsseln eines Tags (a', t') berechnet der Aussteller $m = t' a'^{-x_0}$.

Überprüfung der Eigenschaften

Wir weisen die von uns für eine blinde Chiffre geforderten Eigenschaften Ununterscheidbarkeit und Blindheit nach.

Ununterscheidbarkeit: Der Aussteller erhält vom Empfänger die Markierungen m_i und berechnet die Tags (a_i, t_i) mit $0 \leq i \leq 1$. Der Aussteller wählt $k \in_R \{0,1\}$ und sendet (a_k, t_k) an den Empfänger. Der Empfänger soll nun $l \in \{0,1\}$ so wählen, daß $k = l$ ist.

Der Empfänger wählt $l \in_R \{0,1\}$ und berechnet $h = t_k m_l^{-1}$. Wenn $k = l$ ist, dann gilt $h = a^{x_0}$. Kann der Empfänger entscheiden, ob der Wert l richtig geraten ist, kann er das Decision-Diffie-Hellman Problem (DDH) in G lösen:

Gegeben: $a = g^r, y = g^{x0}$ und h

DDH: Entscheide $h \stackrel{?}{=} g^{rx_0}$

Es wird eine Gruppe G benötigt, in der das Decision-Diffie-Hellman-Problem schwer ist. Beispiele für geeignete Gruppen können in [Bon98] gefunden werden.

Blindheit: Der Aussteller wählt die Markierungen m_i und berechnet die Tags (a_i, t_i) mit $0 \leq i \leq 1$. Der Empfänger blendet die Tags zu (a'_i, t'_i), wählt $k \in_R \{0,1\}$ und sendet (a'_k, t'_k) an den Aussteller. Der Aussteller soll nun $l \in \{0,1\}$ so wählen, daß $k = l$ ist. Wir unterscheiden zwei Fälle:

1. **Die verschlüsselten Markierungen sind gleich.** Das geblendete Tag (a'_k, t'_k) läßt sich auf jedes ungeblendete Tag (a_i, t_i) abbilden. Der dazu jeweils notwendige Blendparameter ist $\lambda = a'_k a_i^{-1}$ und es gilt folgende Abbildung:

$$\begin{aligned} a'_d &= a_i \lambda \\ t'_d &= t_i \lambda^{x_0} \end{aligned}$$

   Der Empfänger benutzt jedoch g^δ mit $\delta \in \mathbb{Z}_q$ zum Blenden. Da für jedes λ ein δ existiert, so daß $\lambda = g^\delta$, kann jedes Tag auf jedes geblendete Tag abgebildet werden. Eine Unterscheidung der Tags ist für den Aussteller nicht möglich.

2. **Die verschlüsselten Markierungen sind ungleich.** Der Aussteller entschlüsselt das Tag und erhält $m_k = t'_k a'^{-x_0}_k$. Der Aussteller wählt l so, daß $m_l = m_k$ ist.

3.2.3 Kombination von Schnorr und ElGamal

Die blinde ElGamal Chiffre ist öffentlich randomisiert. Daher wird das Commitment $a = g^r$ der blinden Schnorr Signatur als öffentliche Randomisierung der blinden Chiffre verwendet. Da das Commitment a vom Empfänger zu $a' = a g^\alpha y^\beta$ geblendet wird, muß das Tag t_0 zu $t'_0 = t_0 (g^\alpha y^\beta)^{x_0} = t_0 y_0^\alpha (y^{x_0})^\beta$ geblendet werden. Andernfalls kann das geblendete Tag vom Aussteller nicht korrekt entschlüsselt werden. Da der Empfänger den privaten Chiffrierschlüssel

des Ausstellers nicht kennt, muß der Aussteller zusätzlich den Wert $z_0 = y^{x_0}$ veröffentlichen. Der Empfänger blendet das Tag dann zu $t'_0 = t_0 y_0^{\alpha} z_0^{\beta}$.

Wir nennen den öffentlichen ElGamal Schlüssel *abhängigen Schlüssel*, da er in Abhängigkeit von dem Schlüssel der blinden Schnorr Signatur erzeugt werden muß.

- **Erzeugung abhängiger ElGamal Schlüssel.** Die Eingabe ist der Systemparameter (g, q, G) und ein öffentlicher Schnorr Schlüssel y. Der private Schlüssel $x_0 \in_R \mathbb{Z}_q$ wird zufällig gewählt. Der öffentliche Schlüssel ist $(y_0 = g^{x_0}, z_0 = y^{x_0})$. Die Ausgabe ist das abhängige Schlüsselpaar $(x_0, (y_0, z_0))$.
- **Entschlüsseln eines (geblendeten) Tags.** Der Aussteller erhält die Signatur (c', s') sowie das Tag t'_0. Anschließend berechnet er das Commitment $a' = g^{s'} y^{c'}$ aus der Signatur und entschlüsselt die Markierung $m = t'_0 a'^{-x_0}$. Der Aussteller akzeptiert das Tag genau dann, wenn m eine gültige Markierung ist.

Überprüfung der Eigenschaften

Die blinde Signatur und die blinde Chiffre sind mit Ausnahme des gemeinsamen Commitments voneinander unabhängig. Wir überprüfen, ob das Verwenden des gleichen Commitments die Sicherheit beeinträchtigt.

Unfälschbarkeit: Die Unfälschbarkeit von Signaturen kann nicht beeinträchtigt werden, da das Commitment öffentlich ist und nichts über die Konstruktion des Commitments verraten wird.

Blindheit: Die Blindheit von Signaturen und Tags kann nicht beeinträchtigt werden, da das Blenden der Signaturen und Tags nicht eingeschränkt wird.

Ununterscheidbarkeit: Tags können unterschieden werden, wenn folgende Bedingungen gleichzeitig erfüllt sind:

1. Es werden mehrere Tags mit dem gleichen Commitment randomisiert.
2. Es wird der gleiche Schlüssel zum Verschlüsseln verwendet.

Der Aussteller erhält vom Empfänger die Markierungen m_i und berechnet die Tags $t_i = a^{x_i} m_i$ mit gemeinsamer Randomisierung a und $0 \leq i \leq 1$. Der Aussteller wählt $k \in_R \{0, 1\}$ und sendet (t_k, t_{1-k}) an den Empfänger. Der Empfänger soll nun $l \in \{0, 1\}$ so wählen, daß $l = k$ ist. Wir unterscheiden zwei Fälle:

1. **Die verwendeten Schlüssel sind gleich.** Der Empfänger berechnet $t = t_k t_{1-k}^{-1} = m_k m_{1-k}^{-1}$ und wählt l so, daß $t = m_l m_{1-l}^{-1}$ ist.

2. **Die verwendeten Schlüssel sind ungleich.** Die Tags können nicht unterschieden werden, da jeder Wert a^{x_i} nur einmal verwendet wird.

3.2.4 Realisierung der Vorgänge

Wir realisieren die Vorgänge INITIALISIEREN, ABHEBEN, BEZAHLEN, EINZAHLEN, VERFOLGEN, ÜBERPRÜFEN und ZURÜCKGEBEN für die erweiterte Variante unseres Zahlungssystems mit Münz- und Kundenverfolgung auf Basis von blinden Schnorr Signaturen und blinden ElGamal Chiffren. Bei der Basisvariante, die nur Münzverfolgung unterstützt, reduzieren sich die drei Tags einer Münze (Index-, Markierungs- und Identitätstag) zum Markierungstag. Entsprechend wird nur ein einziges ElGamal Schlüsselpaar benötigt.

Wir beschreiben alle Vorgänge zur besseren Übersichtlichkeit und Verständlichkeit in einer vereinfachten Form:

1. Die Vorgänge werden mit nur einer einzigen Münze durchgeführt.
2. Die Münzgeneration wird in den Protokollen nicht aufgeführt.

Wir verwenden folgende Notation für die Abhebeauthorisierung und das Abhebe- und Einzahlzertifikat: $\mathcal{C} = \{\text{zu signierende Daten}\}_{\text{Aussteller}}$ enthält die zu signierenden Daten und eine Signatur des Ausstellers.

Initialisieren

Für jede Münzgeneration wählt die Bank neue Schlüsselpaare für die blinde Schnorr Signatur und die blinde ElGamal Chiffre:

- **Erzeugung eines Systemparameters:** Die Bank wählt als Systemparameter eine Gruppe G mit Erzeuger g. In dieser Gruppe muß das Decision-Diffie-Hellman Problem schwer sein.
- **Erzeugung von Schnorr Schlüsselpaaren:** Die Bank wählt $x \in_R \mathbb{Z}_q$ als privaten Schlüssel und berechnet $y = g^x$ als zugehörigen öffentliche Schlüssel
- **Erzeugung von abhängigen ElGamal Schlüsselpaaren:** Die Bank wählt drei private Schlüssel $x_j \in_R \mathbb{Z}_q$ und berechnet den vom öffentlichen Schnorr Signaturschlüssel y abhängigen öffentliche Schlüssel $(y_j = g^{x_j}, z_j = y^{x_j})$ mit $(0 \leq j \leq 2)$.

Sollen mehrere Münzwerte zur Verfügung gestellt werden, muß für jeden Münzwert ein eigenes Schnorr Schlüsselpaar gewählt werden. Entsprechend multipliziert sich damit auch die Anzahl der abhängigen ElGamal Schlüssel. Die Bank publiziert die öffentlichen Schlüssel y und (y_j, z_j).

Alle Markierungen werden zufällig und gleichverteilt aus G gewählt. Für jede Münzgeneration wählt die Bank eine Standardmarkierung $M_{default} \in_R G$ und die Null- bzw. Einsmarkierung $(P_0, P_1) \in_R G^2$.

Abheben mit Verfolgung

Sei S die Sitzungsnummer, die bei der Authentisierung des Kunden von der Bank vergeben wurde.

1. Die Bank und der Kunde erzeugen eine Münze. Die Münze wird als blinde Schnorr Signatur einer Seriennummer berechnet.

   - Der Kunde wählt ein neues Münzschlüsselpaar $(x_m \in_R \mathbb{Z}_q, y_m = g^{x_C})$, einen symmetrischen Rückgabeschlüssel A und den Blendparameter $(\alpha, \beta) \in_R \mathbb{Z}_q^2$. Der Kunde berechnet die Seriennummer als $m = (y_m, MAC_A(\alpha, \beta))$.
   - Protokoll:
     (a) Die Bank wählt $r \in_R \mathbb{Z}_q$, berechnet das Commitment $a = g^r$ und sendet es an den Kunden.
     (b) Der Kunde berechnet die geblendete Challenge c aus der Seriennummer m und dem Commitment a:
        i. $a' = ag^{\alpha}y^{\beta}$
        ii. $c' = H(m, a')$
        iii. $c = c' - \beta \bmod q$.

        Der Kunde sendet die geblendete Challenge c an die Bank.
     (c) Die Bank antwortet mit der Response $s = r - cx \bmod q$.
     (d) Der Kunde blendet die Response zu $s' = s + \alpha \bmod q$ und erhält die Signatur (c', s').
     (e) Der Kunde verifiziert die Signatur (c', s') der Münze: Die Signatur ist gültig, wenn $a' = g^{s'}y^{c'}$ ist.

   Ist die Münze nicht korrekt signiert, wird das Abheben abgebrochen.

2. Der Kunde speichert die Münze (m, c', s'), den privaten Münzschlüssel x_m und die Blendparameter (α, β) in der Münzliste ab. Anschließend erstellt der Kunde die Abhebeauthorisierung $\mathcal{A} = (a, c)_{\text{Kunde}}$ und sendet die Signatur des Zertifikates an die Bank.

3. Die Bank rekonstruiert die Abhebeauthorisierung und überprüft die Signatur des Kunden.

   Schlägt der Test fehl, wird das Abheben abgebrochen.

   Die Bank speichert die blinde Münze (a, c) in der Abhebetabelle und die Abhebeauthorisierung $\mathcal{A}$ in der Abhebeauthorisierungstabelle jeweils unter der Sitzungsnummer S ab.

   Anschließend führt die Bank die Buchung “Kundenkonto an Verrechnungskonto” durch.

4. Die Bank und der Kunde erzeugen für die Münze **drei** Tags. Tags werden als blinde ElGamal Verschlüsselungen berechnet.

- Die Bank wählt eine zufällige, eindeutige Sitzungsmarkierung $M_{session} \in_R G$ und speichert $M_{session}$ in der Markierungstabelle unter der Sitzungsnummer S ab.
  - Sei $M = M_{session}$, wenn Münzverfolgung verwendet werden soll.
  - Sei $M = M_{default}$, wenn der Kunde nicht verfolgt werden soll.
- Protokoll:
  (a) Die Bank wählt $i \in_R \{0, 1\}$ und bestimmt die Permutation der Tags über die Funktion $C_i()$, die das $i$te Element der Argumentenliste auswählt:
    - Das Indextag $t_0 = a^{x_0} P_i$ enthält die Null- oder die Einsmarkierung P_i.
    - Das linke Tag ist $t_1 = a^{x_1} C_i(M, M_{session})$.
    - Das rechte Tag ist $t_2 = a^{x_2} C_i(M_{session}, M)$.

    Die Bank sendet (t_0, t_1, t_2) an den Kunden.
  (b) Der Kunde blendet die Tags t_j zu $t'_j = t_j y_j^{\alpha} z_j^{\beta}$ mit $0 \leq j \leq 2$.

5. Die Bank erstellt das Abhebezertifikat $\mathcal{W} = \{\text{Kunde}, (c, s), (t_0, t_1, t_2)\}_{\text{Bank}}$ und sendet die Signatur des Zertifikates an den Kunden.

6. Der Kunde rekonstruiert das Abhebezertifikat und überprüft die Signatur der Bank.

   Schlägt der Test fehl, gibt der Kunde alle abgehobenen Münzen an die Bank zurück (s. Zurückgeben).

   Der Kunde trägt die Tags (t'_0, t'_1, t'_2) in der Münzliste nach und speichert das Abhebezertifikat $\mathcal{W}$ in der Abhebeliste ab.

Das Abhebeprotokoll ist in Abbildung 3.3 skizziert.

Bezahlen und Einzahlen mit Verfolgung

Der Händler sendet dem Kunden ein Angebot zu:

$$\mathcal{O} = \{\text{Bestellnummer}, \text{Produktbeschreibungen}, \text{Preise}\}_{\text{Händler}}$$

1. Der Kunde speichert das Angebot $\mathcal{O}$ in der Zahlungsliste ab und wählt die Münze (m, c', s') aus, die für die Zahlung verwendet werden sollen. Anschließend signiert der Kunde $o = \{\text{Händler}, \text{Bestellnummer}, \text{Gesamtpreis}\}$ mit dem privaten Münzschlüssel x_m:

   - Der Kunde wählt ein zufälliges $r_m \in_R \mathbb{Z}_q$ und berechnet:
     (a) $a_m = g^{r_m}$
     (b) $c_m = H(o, a_m)$
     (c) $s_m = r_m - c_m x_m \bmod q$
   - Die Münzsignatur ist (c_m, s_m).

Sei $G =< g >$ eine Gruppe mit Primordnung q

Das Schlüsselpaar der Bank für die blinde Signatur:
privat: $x \in_R \mathbb{Z}_q$ öffentlich: $y = g^x$

Das Schlüsselpaar der Bank für die blinde Chiffe $(0 \leq j \leq 2)$**:**
privat: $x_j \in_R \mathbb{Z}_q$ öffentlich: $(y_j = g^{x_j}, z_j = y^{x_j})$

Markierungen:
Standardmarkierung: $M_{default} \in_R G$
Null- und Einsmarkierung: $(P_0, P_1) \in_R G^2$

Bank		Kunde
Markierungen:		**Blendparameter:**
Sitzungsmarkierung: $M_{session} \in_R G$		$(\alpha, \beta) \in_R \mathbb{Z}_q^2$
Ohne Münzverfolgung: $M = M_{default}$		**Authentikationsschlüssel:**
Mit Münzverfolgung: $M = M_{session}$		wähle A zufällig
		Münzschlüsselpaar:
		privat: $x_m \in_R \mathbb{Z}_q$
Sei $C_i(A_0, A_1, \dots) = A_i$		öffentlich: $y_m = g^{x_m}$
		Seriennummer:
		$m = (y_m, MAC_A(\alpha, \beta))$
$r \in_R \mathbb{Z}_q$		
$a = g^r$	$\xrightarrow{a}$	$a' = ag^\alpha y^\beta$
		$c' = H(m, a')$
	$\xleftarrow{c}$	$c = c' - \beta \bmod q$
$s = r - cx \bmod q$	$\xrightarrow{s}$	$s' = s + \alpha \bmod q$
		Überprüfe $a' \stackrel{?}{=} g^{s'} y^{c'}$
Überprüfe $\mathcal{A}$	$\xleftarrow{\mathcal{A}}$	Abhebeauthorisierung $\mathcal{A}$
$i \in_R \{0, 1\}$		
$t_0 = a^{x_0} P_i$		$t'_0 = t_0 y_0^\alpha z_0^\beta$
$t_1 = a^{x_1} C_i(M, M_{session})$	$\xrightarrow{t_0, t_1, t_2}$	$t'_1 = t_1 y_1^\alpha z_1^\beta$
$t_2 = a^{x_2} C_i(M_{session}, M)$		$t'_2 = t_2 y_2^\alpha z_2^\beta$
Abhebezertifikat $\mathcal{W}$	$\xrightarrow{\mathcal{W}}$	Überprüfe $\mathcal{W}$
		Münze: (m, c', s')
		Tags: t'_0, t'_1, t'_2

Abbildung 3.3: Das Abhebeprotokoll von FlexiCash

Der Kunde sendet die Annahmeerklärung $\{o, (c_m, s_m), (m, c', s'), t'_0\}$ an den Händler.

2. Der Händler überprüft die Annahmeerklärung:

   (a) Die Identität des Händlers muß korrekt sein.

   (b) Die Bestellnummer muß vorhanden sein.

   (c) Der Betrag der Münzen muß dem zu zahlenden Betrag entsprechen.

   Schlägt einer der Tests fehl, bricht der Händler die Zahlung ab.

   Der Händler startet die Einzahlung und sendet die Annahmeerklärung an die Bank. Sei S die Sitzungsnummer, die bei der Authentisierung des Händlers von der Bank vergeben wird.

3. Die Bank überprüft die Einzahlung:

   - Die Bank überprüft die Annahmeerklärung:
     - Die Identität des Händlers muß mit dem authentisierten Händler übereinstimmen.
     - Die Münzsignatur (c_m, s_m) muß mit dem Münzschlüssel y_m verifiziert werden können:
       * Die Bank extrahiert y_m aus der Seriennummer m.
       * Die Bank berechnet $a_m = g^{s_m} y_m^{c_m}$.
       * Die Signatur ist gültig, wenn $c_m = H(o, a_m)$ ist.
   - Die Bank überprüft die Münzen:
     - Der Einzahlzeitpunkt muß in der Akzeptanzphase der Münze liegen.
     - Die Seriennummer m darf nicht in der Einzahltabelle enthalten sein.
     - Die Signatur der Münze muß gültig sein:
       * Die Bank berechnet $a' = g^{s'} y^{c'}$.
       * Die Signatur ist gültig, wenn $c' = H(m, a')$ ist.

   Schlägt einer der Tests fehl, wird die Einzahlung abgelehnt.

   Die Bank speichert die Seriennummer m und die Münzsignatur (c_m, s_m) in der Einzahltabelle sowie die Annahmeerklärung o in der Annahmeerklärungstabelle jeweils unter der Sitzungsnummer S ab. Die Münze ist jetzt ungültig. Wird die Einzahlung im folgenden abgebrochen, muß sie später fortgesetzt werden!

   Die Bank entschlüsselt das Indextag, indem sie $P = t'_0 a'^{-x_0}$ berechnet.

   - $P = P_0$: Setze $i = 0$.
     - Das linke Tag (t'_1) ist das Markierungstag.

  - Das rechte Tag (t'_2) ist das Identitätstag.
- $P = P_1$: Setze $i = 1$.
  - Das linke Tag (t'_1) ist das Identitätstag.
  - Das rechte Tag (t'_2) ist das Markierungstag.

Enthält das Indextag einen ungültigen Wert, wird die Einzahlung abgelehnt.

Anschließend verlangt die Bank entweder das linke $(d = 0)$ oder rechte Tag $(d = 1)$ vom Kunden: Soll die Bank Kundenverfolgung bei dem Händler durchführen, berechnet sie $d = 1 - i$, andernfalls $d = i$.

Die Bank stellt das Einzahlzertifikat $\mathcal{D} = \{\text{Händler}, d, m, c', s', t'_0\}_{\text{Bank}}$ aus und sendet d und die Signatur des Zertifikates an den Händler.

4. Der Händler sendet d und die Signatur an den Kunden.

5. Der Kunde rekonstruiert das Einzahlzertifikat und überprüft die Signatur der Bank.

   Schlägt der Tests fehl, wird die Zahlung abgebrochen.

   Der Kunde trägt das Auswahlbit d in der Münzliste sowie das Einzahlzertifikat in der Zahlungsliste nach und antwortet dem Händler mit dem Tag t'_{1+d}.

6. Der Händler sendet das Tag t'_{1+d} an die Bank.

7. Die Bank entschlüsselt das Tag t'_{1+d}, indem sie $M = t'_{1+d} a'^{-x_{1+d}}$ berechnet.

   - Hat die Bank Kundenverfolgung beim Bezahlen durchgeführt, muß M eine gültige Markierung aus der Markierungstabelle sein.
   - Andernfalls kann die Bank Münzverfolgung beim Abheben durchgeführt haben:
     - Ist $M = M_{default}$, ist die Münze anonym.
     - Andernfalls muß M eine Sitzungsmarkierung sein und die Bank ermittelt die zugehörige Kundenidentität aus der Markierungs- und der Sitzungstabelle.

   Die extrahierte Markierung M wird in der Einzahltabelle nachgetragen.

   Enthält ein Tag eine ungültige Markierung oder eine Markierung aus einer Erpressung, wird die Einzahlung abgelehnt.

   Die Bank führt die Buchung “Verrechnungskonto an Händlerkonto” durch und informiert den Händler über die erfolgreiche Einzahlung.

8. Der Händler liefert die Waren an den Kunden.

Das Zahlungsprotokoll ist in Abbildung 3.4 skizziert.

Sei $G =< g >$ eine Gruppe mit Primordnung q

Das Schlüsselpaar der Bank für die blinde Signatur:
privat: $x \in_R \mathbb{Z}_q$ öffentlich: $y = g^x$

Das Schlüsselpaar der Bank für die blinde Chiffe $(0 \leq j \leq 2)$**:**
privat: $x_j \in_R \mathbb{Z}_q$ öffentlich: $(g_j = g^{x_j}, y_j = y^{x_j})$

Markierungen:
Standardmarkierung: $M_{default} \in_R G$
Null- und Einsmarkierung: $(P_0, P_1) \in_R G^2$

Bank	Händler	Kunde
		Münze, Tags:
$a'_m \stackrel{?}{=} g^{s'_m} y_m^{c'_m}$	$(m, c', s'), t'_0 \longleftarrow$	$m = (y_m, k_m)$
$c'_m \stackrel{?}{=} H(o, a'_m)$	$\longleftarrow (o, c_m, s_m)$	$(c', s'), (t'_0, t'_1, t'_2)$
$a' \stackrel{?}{=} g^{s'} y^{c'}$		**Münzsignatur:**
$c' \stackrel{?}{=} H(m, a')$		zu signieren: o
$P = t'_0 a'^{-x_0}$		$r_m \in_R \mathbb{Z}_q$
Wähle i so, daß $P = P_i$		$a_m = g^{r_m}$
Ohne Kundenverfolgung:		$c_m = H(o, a_m)$
$d = i$		$s_m = r_m - c_m x_m$
Mit Kundenverfolgung:		Signatur: (c_m, s_m)
$d = 1 - i$		
Einzahlzertifikat $\mathcal{D}$	$\mathcal{D} \longrightarrow$	Überprüfe $\mathcal{D}$
$M = t'_{1+d} a'^{-x_{1+d}}$	$t'_{1+d} \longleftarrow$	

Ohne Kundenverfolgung:
$M = M_{default}$: Die Münze ist anonym
$M \neq M_{default}$: M ist eine Sitzungsmarkierung (Münzverfolgung)

Mit Kundenverfolgung:
M ist eine Sitzungsmarkierung

Abbildung 3.4: Das Zahlungsprotokoll von FlexiCash

Zurückgeben

Sei S die Sitzungsnummer, die bei der Authentisierung des Kunden gegenüber der Bank vergeben wurde.

1. Der Kunde wählt eine Münze (m, c', s') aus und erstellt eine Rückgabesignatur (c_m, s_m), die mit dem privaten Münzschlüssel x_m berechnet wird:

   - Der Kunde wählt ein zufälliges $r_m \in_R \mathbb{Z}_q$ und berechnet:
     (a) $a_m = g^{r_m}$
     (b) $c_m = H(m, a_m)$
     (c) $s_m = r_m - c_m x_m \bmod q$
   - Die Rückgabesignatur ist (c_m, s_m).

   Der Kunde sendet die Seriennummer m, die Rückgabesignatur (c_m, s_m), die blinde Münze (a, c), den Blendparameter (α, β) und den geheimen Rückgabeschlüssel A an die Bank.

2. Die Bank überprüft die Rückgabe:

   (a) Die blinde Münze (a, c) muß in der Abhebetabelle vorhanden und von dem gleichen Kunden abgehoben worden sein.
   (b) Die Seriennummer m darf nicht in der Einzahltabelle vorhanden sein.
   (c) Der Authentisierungscode muß sich als $MAC_A(\alpha, \beta)$ berechnen lassen.
   (d) Die blinde Münze muß sich aus der Seriennummer m berechnen lassen:
      i. Die Bank berechnet $a' = ag^{\alpha}y^{\beta}$.
      ii. Die Seriennummer ist gültig, wenn $c = H(m, a') - \beta \bmod q$ ist.
   (e) Die Rückgabesignatur (c_m, s_m) muß mit dem Münzschlüssel y_m verifiziert werden können:
      - Die Bank extrahiert y_m aus der Seriennummer m.
      - Die Bank berechnet $a_m = g^{s_m} y_m^{c_m}$.
      - Die Signatur ist gültig, wenn $c_m = H(m, a_m)$ ist.

   Schlägt einer der Tests fehl, wird die Rückgabe abgebrochen.

   Die Bank speichert die Seriennummer m und die Rückgabesignatur (c_m, s_m) in der Einzahltabelle unter der Sitzungsnummer S ab. Die Münze ist jetzt ungültig.

   Die Bank führt die Buchung "Verrechnungskonto an Kundenkonto" durch und informiert den Kunden über die erfolgreiche Rückgabe.

Das Rückgabeprotokoll ist in Abbildung 3.5 skizziert.

Sei $G =< g >$ eine Gruppe mit Primordnung q

Das Schlüsselpaar der Bank für die blinde Signatur:
privat: $x \in_R \mathbb{Z}_q$ öffentlich: $y = g^x$

Das Schlüsselpaar der Bank für die blinde Chiffe $(0 \leq j \leq 2)$**:**
privat: $x_j \in_R \mathbb{Z}_q$ öffentlich: $(g_j = g^{x_j}, y_j = y^{x_j})$

Markierungen:
Standardmarkierung: $M_{default} \in_R G$
Null- und Einsmarkierung: $(P_0, P_1) \in_R G^2$

Bank		Kunde
		Blinde Münze:
$k_m \stackrel{?}{=} MAC_A(\alpha, \beta)$	$(a, c), (\alpha, \beta), A$ ◄	$m = (y_m, k_m)$
$a' = ag^\alpha y^\beta$	(m, c_m, s_m)	$(a, c), (\alpha, \beta), A$
$c \stackrel{?}{=} H(m, a') - \beta \bmod q$		**Münzsignatur:**
$a'_m = g^{s'_m} y_m^{c'_m}$		zu signieren: m
$c'_m \stackrel{?}{=} H(o, a'_m)$		$r_m \in_R \mathbb{Z}_q$
		$a_m = g^{r_m}$
		$c_m = H(o, a_m)$
		$s_m = r_m - c_m x_m$
		Signatur: (c_m, s_m)

Abbildung 3.5: Das Rückgabeprotokoll von FlexiCash

Überprüfen

Die Bank publiziert die privaten Markierungsschlüssel x_j mit $0 \leq j \leq 2$, die Standardmarkierung $M_{default}$ und die Null- bzw. Einsmarkierung (P_0, P_1). Anschließend rekonstruiert der Kunde für jede Münze das Commitment $a' = g^{s'} y^{c'}$ aus der Signatur der Münze und überprüft die Inhalte und die Permutation der Tags:

1. Der Kunde lernt die Zuordnung von Markierungs- und Identitätstag zum linken und rechten Tag, indem er das Indextag zu $P = t'_0 a'^{-x_0}$ entschlüsselt und überprüft, ob die Null- oder Einsmarkierung enthalten ist:

   - $P = P_0$: Setze $i = 0$.
     - Das linke Tag (t'_1) ist das Markierungstag.
     - Das rechte Tag (t'_2) ist das Identitätstag.
   - $P = P_1$: Setze $i = 1$.
     - Das linke Tag (t'_1) ist das Identitätstag.
     - Das rechte Tag (t'_2) ist das Markierungstag.
   - Andernfalls hat der Kunde eine unübliche Form der Münzverfolgung entdeckt.

2. Der Kunde überprüft Münzverfolgung, indem er das Markierungstag zu $M = t'_{1+i} a'^{-x_{1+i}}$ entschlüsselt:
   - Ist $M \neq M_{default}$, hat der Kunde Münzverfolgung festgestellt.
   - Andernfalls ist die Münze anonym.
3. Der Kunde überprüft Kundenverfolgung:
   - Ist $d \neq i$, hat der Kunde Kundenverfolgung festgestellt.
   - Andernfalls ist die Zahlung anonym.

Kann die Bank kein Deanonymisierungszertifikat für den Kunden bzw. Händler vorlegen, hat der Kunde eine illegale Deanonymisierung festgestellt und sendet das Abhebezertifikat $\mathcal{W}$ bzw. Einzahlzertifikat $\mathcal{D}$ an einen Richter.

- **Illegale Münzverfolgung:** Der Richter überprüft das Abhebezertifikat $\mathcal{W}$. Das Abhebezertifikat enthält:
  $$\mathcal{W} = \{\text{Kunde}, (c, s), (t_0, t_1, t_2)\}_{\text{Bank}}$$
  Liegt für den Kunden kein Deanonymisierungszertifikat vor, rekonstruiert der Richter das Commitment $a = g^s y^c$, entschlüsselt das Indextag $P = t_0 a^{-x_0}$, bestimmt dadurch die Permutation i und entschlüsselt das Markierungtag $M = t_{1+i} a^{-x_{1+i}}$. Ist $M \neq M_{default}$, hat der Richter illegale Münzverfolgung festgestellt.
- **Illegale Kundenverfolgung:** Der Richter überprüft das Einzahlzertifikat $\mathcal{D}$. Das Einzahlzertifikate enthält:
  $$\mathcal{D} = \{\text{Händler}, d, m, c', s', t'_0\}_{\text{Bank}}$$
  Liegt für den Händler kein Deanonymisierungszertifikat vor, rekonstruiert der Richter das Commitment $a' = g^{s'} y^{c'}$, entschlüsselt das Indextag $P = t'_0 a'^{-x_0}$ und bestimmt die Permutation i, so daß $P = P_i$ ist. Ist $d \neq i$, hat der Richter illegale Kundenverfolgung festgestellt.

3.3 Alternative Realisierungen

Wir sind bisher nur in der Lage, eine Realisierung auf dem allgemeinen diskreten Logarithmusproblem anzugeben. Obwohl wir glauben, daß Realisierungen auf anderen Basisproblemen existieren, ist es ein offenes Problem, eine andere, sichere und effiziente Lösung zu finden. Prinzipiell kann jede randomisierte blinde Signatur und blinde Chiffre miteinander kombiniert werden. Die beiden Primitive müssen jedoch kompatibel zueinander sein. Kompatible Primitive sind auf dem gleichen Basisproblem aufgebaut und verwenden eine zueinander passende Form der Blendung. Zur Zeit existieren keine passenden Primitive auf anderen Basisproblemen.

Mögliche Kandidaten für eine alternative Realisierung auf Basis des RSA-Problems sind die blinde Guillou-Quisquater Signatur [GQ88] und die blinde Paillier Chiffre [Pai99]. Diese Primitive sind jedoch nicht zueinander kompatibel.

Kapitel 4

Experimentelle Ergebnisse

In diesem Kapitel stellen wir die experimentellen Ergebnisse einer prototypischen Implementierung von FlexiCash vor. Die Implementierung erfolgte in Java mit Hilfe der von uns um blinde Signaturen und blinde Chiffren erweiterten Java Cryptography Architecture. Alle in diesem Kapitel aufgeführten Laufzeiten wurden auf einem bzw. mehreren Intel Pentium II mit 333MHz unter IBM JDK 1.3 für Linux gemessen.

Gliederung des Kapitels

Wir beginnen in Abschnitt 4.1 mit einer Beschreibung der Implementierung der blinden Schnorr Signatur und der blinden ElGamal Chiffre. Darauf aufbauend diskutieren wir in Abschnitt 4.2 die Implementierung der Vorgänge von FlexiCash. Schließlich fassen wir die Ergebnisse in Abschnitt 4.3 zusammen und diskutieren die Skalierbarkeit von FlexiCash.

4.1 Implementierung der Primitive

Im folgenden beschreiben wir die Kenndaten der verwendeten Primitive, insbesondere DSA Systemparameter und Schlüssel, die blinde Schnorr Signatur und die blinde ElGamal Chiffre.

4.1.1 DSA Systemparameter und Schlüssel

Für die blinde Schnorr Signatur und die blinde ElGamal Chiffre verwenden wir DSA [Nat94] Systemparameter und Schlüssel.

Erzeugung des Systemparameters

Der Systemparameter ist eine zyklische Untergruppe $< g >$ der Ordnung q von $\mathbb{Z}_p^*$ mit p und q Primzahlen. Die Länge von q ist auf 160 Bit festgelegt, die Länge von p ist beliebig.

DSA Schlüsselerzeugung	**Laufzeit**		**Ausgabegröße**	
Operation	**512 Bit**	**1024 Bit**	**512 Bit**	**1024 Bit**
Schlüsselerzeugung mit Systemparameter	8 ms	29 ms	64 Byte	128 Byte
Erzeugung abhängiger Schlüssel	17 ms	58 ms	135 Byte	263 Byte

Tabelle 4.1: Daten der DSA Schlüsselerzeugung

Das Erzeugen von Systemparametern ist eine probabilistische Operation. Diese Operation wird einmalig in der Initialisierungsphase durchgeführt. Die Laufzeit der Erzeugung von Systemparametern wird daher im folgenden nicht weiter betrachtet.

Schlüsselerzeugung

Der private Schlüssel ist ein zufälliger Wert $x \in_R \mathbb{Z}_q$, der öffentliche Schlüssel ist $y = g^x \bmod p$. Der vom öffentlichen Schlüssel y abhängige öffentliche Schlüssel ist $(y_1 = g^{x_0} \bmod p, y_2 = y^{x_0} \bmod p)$, wobei $x_0 \in_R \mathbb{Z}_q$ der zugehörige private Schlüssel ist.

Für die Schlüsselerzeugung ergeben sich folgende Kenndaten:

- **Erzeugung von DSA Schlüsseln.** Das Erzeugen von DSA Schlüsseln mit gegebenen Systemparametern benötigt die Zeit t_{keygen}. Ein öffentlicher DSA Schlüssel (ohne den verwendeten Systemparameter) hat die Größe s_{key}.
- **Erzeugung abhängiger DSA Schlüssel.** Die Erzeugung von abhängigen DSA Schlüsseln benötigt die Zeit $t_{depkeygen}$. Ein öffentlicher abhängiger DSA Schlüssel hat die Größe s_{depkey}.

Für Systemparameter der Größe $|p| = 512$ und $|p| = 1024$ sind konkrete Meßwerte in Tabelle 4.1 abgebildet.

ASN.1 Syntax

```
DSAPublicKey ::= INTEGER

DependantDSAPublicKey ::= SEQUENCE{
  y1 INTEGER,
  y2 INTEGER,
}

DSAAlgorithmParameters ::= SEQUENCE {
  p INTEGER,
```

Blinde Schnorr Signaturen	**Laufzeit**		**Ausgabegröße**	
Operation	**512 Bit**	**1024 Bit**	**512 Bit**	**1024 Bit**
Erzeugen eines Commitments	8 ms	29 ms	64 Byte	128 Byte
Wahl eines Blendparameters	≈ 0 ms	≈ 0 ms	47 Byte	47 Byte
Blenden der Nachricht	17 ms	58 ms	20 Byte	20 Byte
Signieren der geblendeten Nachricht	≈ 0 ms	≈ 0 ms	20 Byte	20 Byte
Entblenden der geblendeten Signatur	≈ 0 ms	≈ 0 ms	47 Byte	47 Byte
Verifikation einer (blinden) Signatur	17 ms	58 ms	1 Bit	1 Bit
Erzeugen einer regulären Signatur	9 ms	30 ms	47 Byte	47 Byte

Tabelle 4.2: Daten der blinden Schnorr Signatur

```
  q INTEGER,
  g INTEGER,
}
```

4.1.2 Blinde Schnorr Signaturen

Schnorr Signaturen sind nicht standardisiert. Wir verwenden DSA Systemparameter (g, p, q) und DSA Schlüssel (x, y) und die SHA-1 Hashfunktion [Nat95] zur Erstellung von (blinden) Schnorr Signaturen. Daraus ergeben sich folgende Kenndaten für die Erstellung von blinden bzw. regulären Schnorr Signaturen:

- **Erzeugen eines Commitments.** Das Erzeugen eines Commitments $a = g^r \bmod p$ mit zufällig gewähltem $r \in_R \mathbb{Z}_q$ erfolgt in Zeit $t_{commitment}$. Das Commitment hat die Größe $s_{commitment}$.

- **Wahl eines Blendparameters.** Das Wählen eines Blendparameters $(\alpha, \beta) \in_R \mathbb{Z}_q^2$ erfolgt in Zeit t_{bpgen}. Ein Blendparameter hat die Größe s_{bp}.

- **Blenden der Nachricht.** Das Erzeugen der geblendeten Challenge $c = c' - \beta \bmod q$ aus der Challenge $c' = H(m, a') \bmod q$ und dem geblendeten Commitment $a' = ag^\alpha y^\beta \bmod p$ erfolgt in Zeit $t_{blinding}$. Die geblendete Challenge hat die Größe $s_{blindmessage}$.

- **Signieren der geblendeten Nachricht.** Das Erstellen der geblendeten Response $s = r - cx \bmod q$ erfolgt in Zeit $t_{blindsign}$. Die geblendete Response hat die Größe $s_{blindsignature}$.

- **Entblenden der geblendeten Signatur.** Das Entblenden einer geblendeten Response erfolgt in Zeit $t_{unblind}$. Die entblendete Signatur ist $s' = s + \alpha \bmod q$ und hat die Größe $s_{signature}$.

Blinde ElGamal Chiffren	**Laufzeit**		**Ausgabegröße**	
Operation	**512 Bit**	**1024 Bit**	**512 Bit**	**1024 Bit**
Erzeugen einer Markierung	8 ms	29 ms	64 Byte	128 Byte
Erzeugen einer Randomisierung	8 ms	29 ms	64 Byte	128 Byte
Verschlüsseln einer Markierung	8 ms	29 ms	64 Byte	128 Byte
Blenden eines Tags	17 ms	58 ms	64 Byte	128 Byte
Entschlüsseln eines (geblendeten) Tags	8 ms	29 ms	64 Byte	128 Byte

Tabelle 4.3: Daten der blinden ElGamal Chiffre

- **Verifikation einer (blinden) Signatur.** Die Verifikation einer (blinden) Schnorr Signatur (c, s) für Nachricht m erfolgt in Zeit t_{verify} durch Berechnung von $a = g^s y^c \bmod p$. Die Signatur ist korrekt, wenn $c \stackrel{?}{=} H(m, a)$ ist. Die Ausgabe ist `true` oder `false` und hat die Größe 1 Bit.

- **Erzeugen einer regulären Signatur.** Das Erzeugen einer regulären Signatur erfolgt in Zeit $t_{sign} = t_{commitment} + t_{blindsign}$. Die Signatur hat die Größe $s_{signature}$.

Für Systemparameter der Größe $|p| = 512$ und $|p| = 1024$ sind konkrete Meßwerte in Tabelle 4.2 abgebildet.

ASN.1 Syntax

```
SchnorrSignature ::= SEQUENCE {
 challenge INTEGER,
 response  INTEGER
}

SchnorrBlindingParameter ::= SEQUENCE {
 alpha INTEGER,
 beta  INTEGER
}

SchnorrCommitment ::= INTEGER
SchnorrBlindMessage ::= INTEGER
SchnorrBlindSignature ::= INTEGER
```

4.1.3 Blinde ElGamal Chiffren

Blinde ElGamal Chiffren sind ebenfalls nicht standardisiert. Wir verwenden DSA Systemparameter (g, p, q) und abhängige DSA Schlüssel[1] (x, y_1, y_2) sowie Schnorr-Blendparameter (α, β) zur Erstellung von Markierungen und Tags.

Daraus ergeben sich folgende Kenndaten für die blinde ElGamal Chiffre:

- **Erzeugen einer Markierung.** Das Erzeugen einer Markierung $M = g^r \bmod p$ mit zufällig gewähltem $r \in_R \mathbb{Z}_q$ erfolgt in Zeit t_{mark}. Eine Markierung hat die Größe s_{mark}.
- **Erzeugen einer Randomisierung.** Das Erzeugen einer öffentlichen Randomisierung $a = g^r \bmod p$ mit zufällig gewähltem $r \in_R \mathbb{Z}_q$ erfolgt in Zeit $t_{randomize}$. Eine öffentliche Randomisierung hat die Größe $s_{randomization}$.
- **Verschlüsseln einer Markierung.** Das Verschlüsseln einer Markierung erfolgt in Zeit $t_{encrypt}$. Das Tag $t = a^x M \bmod p$ hat die Größe s_{tag}.
- **Blenden eines Tags.** Das Blenden eines Tags erfolgt in Zeit $t_{tagblinding}$. Das geblendete Tag $t' = t y_1^\alpha y_2^\beta \bmod p$ hat die Größe s_{tag}.
- **Entschlüsseln eines (geblendeten) Tags.** Das Entschlüsseln der Markierung $M = t'/a'^x \bmod p$ erfolgt in Zeit $t_{decrypt}$. Die Markierung hat die Größe s_{mark}.

Für Systemparameter der Größe $|p| = 512$ und $|p| = 1024$ sind konkrete Meßwerte in Tabelle 4.3 abgebildet.

ASN.1 Syntax

```
ElGamalRandomization ::= INTEGER
ElGamalTag ::= INTEGER
ElGamalMark ::= INTEGER
```

4.1.4 Sonstige Primitive

Als zusätzliche Primitive verwenden wir einen Message Authentication Code und ein Signaturverfahren, das zur Erstellung der Zertifikate (Abhebeauthorisierung sowie Abhebe- und Einzahlzertifikat) verwendet wird.

Als Message Authentication Code benutzen wir einen SHA1 basierten HMAC. Dabei haben wir folgende Meßwerte ermittelt:

[1] Die Operationen der blinden ElGamal Chiffre können mit DSA Schlüssel oder mit abhängigen DSA Schlüssel durchgeführt werden. Im folgenden verwenden wir jedoch ausschließlich abhängige DSA Schlüssel.

Münze	1 ct	2 ct	4 ct	8 ct	16 ct	32 ct	64 ct	128 ct	256 ct	512 ct	$\sum$
10€	10	11	10	10	11	11	5	–	–	–	68
100€	10	11	10	11	11	10	10	10	11	9	103

Tabelle 4.4: Verwendete Münzen

- Der Schlüssel hat die Größe $s_{authkey} = 20$ Byte.
- Ein Authentisierungscode hat die Größe $s_{mac} = 20$ Byte.
- Die Zeit zum Erstellen eines Authentisierungscodes beträgt $t_{mac} = 1$ ms.

Für das Ausstellen der Zertifikate verwenden wir eine 1024 Bit RSA Signatur nach PKCS #1 Version 1.5 [RSA78, Lab93]. Dabei haben wir folgende Meßwerte ermittelt:

- Die Zeit zum Ausstellen einer Signatur beträgt $t_{issue(cert)} = 60$ ms.
- Die Zeit zum Überprüfen einer Signatur beträgt $t_{verify(cert)} = 10$ ms.
- Eine Signatur hat die Größe $s_{cert} = 220$ Byte.

4.2 Implementierung der Vorgänge

In diesem Abschnitt diskutieren wir die Laufzeiten der Vorgänge ABHEBEN, BEZAHLEN, EINZAHLEN, ZURÜCKGEBEN und ÜBERPRÜFEN. Für jeden dieser Vorgänge berechnen wir zunächst die zu erwartende Laufzeit und die Menge der zu kommunizierenden Daten. Anschließend vergleichen wir die berechneten Werte mit den tatsächlich gemessenen Werten.

Für die Berechnung verwenden wir die Daten aus den Tabellen 4.1, 4.2 und 4.3.

Für alle Vorgänge verwenden wir 10€ in 68 Münzen und 100€ in 103 Münzen. Tabelle 4.4 gibt einen Überblick über die verwendeten Münzen. Diese Verteilung der Münzen garantiert dem Kunden, mindestens zehn Zahlungen durchführen zu können. Zur besseren Vergleichsmöglichkeit der Vorgänge benutzen wir für jeden Vorgang immer alle Münzen.

4.2.1 Abheben

An dem Vorgang ABHEBEN sind Kunde und Bank beteiligt. Folgende Schritte werden durchgeführt:

1. **Beginn der Sitzung.**

   (a) Die Bank erzeugt eine neue Sitzungsmarkierung in Zeit t_{mark}.

2. **Erzeugen der Münzen.**

   (a) Der Kunde erzeugt eine Seriennummer. Dazu erzeugt er ein Schlüsselpaar mit den gegebenen Systemparametern, wählt den Blendparameter und berechnet einen MAC über den Blendparameter. Die zum Erstellen der Seriennummer benötigte Zeit ist $t_{serial} = t_{keygen} + t_{bpgen} + t_{mac}$.

   (b) Die Bank erzeugt ein Commitment in Zeit $t_{commitment}$. Das Commitment wird an den Kunden gesendet und hat die Größe $s_{commitment}$.

   (c) Der Kunde blendet die Seriennummer in Zeit $t_{blinding}$ und sendet die geblendete Nachricht mit der Größe $s_{blindmessage}$ an die Bank.

   (d) Die Bank signiert die geblendete Nachricht in Zeit $t_{blindsign}$ und sendet die geblendete Signatur mit der Größe $s_{blindsignature}$ an den Kunden.

   (e) Der Kunde entblendet und verifiziert die Signatur in Zeit $t_{unblinding}$ und t_{verify}.

3. **Erzeugen der Abhebeauthorisierung.**

   (a) Der Kunde erstellt die Abhebeauthorisierung in Zeit $t_{issue(\mathcal{A})}$ und sendet die Abhebeauthorisierung mit Größe $s_{certificate(\mathcal{A})}$ an die Bank.

   (b) Die Bank überprüft die Abhebeauthorisierung in Zeit $t_{verify(\mathcal{A})}$.

4. **Erzeugen der Tags.**

   (a) Die Bank verschlüsselt drei Markierungen in Zeit $3 \cdot t_{encrypt}$ und sendet die Tags mit der Größe $3 \cdot s_{tag}$ an den Kunden.

   (b) Der Kunde blendet die drei Tags in Zeit $3 \cdot t_{tagblinding}$.

5. **Erzeugen des Abhebezertifikates.**

   (a) Die Bank erstellt das Abhebezertifikat in Zeit $t_{issue(\mathcal{W})}$ und sendet das Abhebezertifikat mit der Größe $s_{certificate(\mathcal{W})}$ an den Kunden.

   (b) Der Kunde überprüft das Abhebezertifikat in Zeit $t_{verify(\mathcal{W})}$.

Wir berechnen nun die zu erwartenden Laufzeiten und die Menge der zu kommunizierenden Daten für das Abheben einer Münze. Für diese Berechnung vernachlässigen wir die Zeiten zum Erzeugen der Sitzungsmarkierung, zum Erstellen und Überprüfen der Abhebeauthorisierung und des Abhebezertifikats sowie die Größen der Zertifikate.

- Laufzeiten der Operationen des Kunden:

$$t_{withdrawal_c} = t_{serial} + t_{blinding} + t_{unblinding} + t_{verify} + 3 \cdot t_{tagblinding} \tag{4.1}$$

Abheben	**Kunde**		**Bank**		**Gesamt**		**Kommunikation**	
	512	**1024**	**512**	**1024**	**512**	**1024**	**512**	**1024**
Pro Münze	94 ms	320 ms	32 ms	116 ms	126 ms	436 ms	296 Byte	552 Byte
Für 10€	6,4 s	21,8 s	2,2 s	7,9 s	8,6 s	29,6 s	19,7 KB	36,7 KB
Gemessen	7,6 s	22,2 s	2,5 s	8,0 s	10,8 s	25,8 s	21,0 KB	38,4 KB
*Overhead**	18 ms	6 ms	5 ms	2 ms	33 ms	−57 ms	13 Byte	13 Byte
Für 100€	9,7 s	33,0 s	3,3 s	11,9 s	13,0 s	44,9 s	29,8 KB	55,5 KB
Gemessen	11,6 s	33,9 s	3,9 s	12,2 s	17,2 s	39,1 s	31,6 KB	57,9 KB
*Overhead**	19 ms	9 ms	6 ms	2 ms	38 ms	−56 ms	11 Byte	10 Byte

*Overhead pro Münze

Tabelle 4.5: Daten des Vorgangs Abheben

- Laufzeiten der Operationen der Bank:

$$t_{withdrawal_b} = t_{commitment} + t_{blindsign} + 3 \cdot t_{encrypt} \tag{4.2}$$

- Kommunikation zwischen Kunde und Bank:

$$s_{withdrawal} = s_{commitment} + s_{blindmessage} + s_{blindsignature} + 3 \cdot s_{tag} \tag{4.3}$$

Unter der vereinfachenden Annahme, daß die Operationen des Kunden und der Bank sequentiell ausgeführt werden, ergibt sich als Gesamtlaufzeit für einen Abhebevorgang:

$$t_{withdrawal} = t_{withdrawal_c} + t_{withdrawal_b}$$

Auf dieser Basis berechnen wir die Daten für einen Abhebevorgang mit einem 512 Bit Systemparameter und einem 1024 Bit Systemparameter. Die berechneten Werte vergleichen wir anschließend mit den tatsächlich gemessenen Werten. Tabelle 4.5 faßt die berechneten und gemessenen Daten zusammen.

512 Bit Systemparameter

Wir berechnen für die Operationen des Kunden nach Formel 4.1 beim Abheben eine Laufzeit von 94 ms pro Münze. Für die Operationen der Bank berechnen wir nach Formel 4.2 für das Erstellen einer Münze 32 ms Rechenzeit. Nach Formel 4.3 werden pro Münze 296 Byte Daten zwischen Kunde und Bank ausgetauscht.

- Abheben von 10€ (in 68 Münzen):
  - Wir berechnen eine Gesamtlaufzeit von 8,6 s. Dabei entfallen 6,4 s Rechenzeit auf den Kunden und 2,2 s auf die Bank. Insgesamt werden 19,7 KB Daten übertragen.

  - Wir haben eine Gesamtlaufzeit von $10,8$ s gemessen. Dabei entfielen $7,6$ s Rechenzeit auf den Kunden und $2,5$ s auf die Bank. Insgesamt wurden 21 KB Daten übertragen.
  - Es ergibt sich pro Münze ein Overhead von 18 ms für den Kunden und von 5 ms für die Bank.

- Abheben von 100€ (in 103 Münzen):

  - Wir berechnen eine Gesamtlaufzeit von $13,0$ s. Dabei entfallen $9,7$ s Rechenzeit auf den Kunden und $3,3$ s auf die Bank. Insgesamt werden $29,8$ KB Daten übertragen.
  - Wir haben eine Gesamtlaufzeit von $17,2$ s gemessen. Dabei entfielen $11,6$ s Rechenzeit auf den Kunden und $3,9$ s auf die Bank. Insgesamt wurden $31,6$ KB Daten übertragen.
  - Es ergibt sich pro Münze ein Overhead von 19 ms für den Kunden und von 6 ms für die Bank.

1024 Bit Systemparameter

Wir berechnen für die Operationen des Kunden nach Formel 4.1 beim Abheben eine Laufzeit von 320 ms pro Münze. Für die Operationen der Bank berechnen wir nach Formel 4.2 für das Erstellen einer Münze 116 ms Rechenzeit. Nach Formel 4.3 werden pro Münze 552 Byte Daten zwischen Kunde und Bank ausgetauscht.

- Abheben von 10€ (in 68 Münzen):

  - Wir berechnen eine Gesamtlaufzeit von $29,6$ s. Dabei entfallen $21,8$ s Rechenzeit auf den Kunden und $7,9$ s auf die Bank. Insgesamt werden $36,7$ KB Daten übertragen.
  - Wir haben eine Gesamtlaufzeit von $25,8$ s gemessen. Dabei entfielen $22,2$ s Rechenzeit auf den Kunden und $8,0$ s auf die Bank. Insgesamt wurden $38,4$ KB Daten übertragen.
  - Es ergibt sich pro Münze ein Overhead von 6 ms für den Kunden und von 2 ms für die Bank.

- Abheben von 100€ (in 103 Münzen):

  - Wir berechnen eine Gesamtlaufzeit von $44,9$ s. Dabei entfallen $33,0$ s Rechenzeit auf den Kunden und $11,9$ s auf die Bank. Insgesamt werden $55,5$ KB Daten übertragen.
  - Wir haben eine Gesamtlaufzeit von $39,1$ s gemessen. Dabei entfielen $33,9$ s Rechenzeit auf den Kunden und $12,0$ s auf die Bank. Insgesamt wurden $57,9$ KB Daten übertragen.
  - Es ergibt sich pro Münze ein Overhead von 9 ms für den Kunden und von 2 ms für die Bank.

4.2.2 Bezahlen und Einzahlen

An dem Vorgang BEZAHLEN und EINZAHLEN sind Kunde, Händler und Bank beteiligt. Da der Händler im wesentlichen die Daten zwischen Kunde und Bank vermittelt, sind die Laufzeiten des Händlers zu vernachlässigen und werden hier nicht betrachtet. Folgende Schritte werden durchgeführt:

1. **Signieren der Annahmeerklärung.**
   (a) Der Kunde erstellt eine Münzsignatur in Zeit $t_{signature}$ und sendet folgende Daten an die Bank: Die Münze mit der Größe $s_{coin} = s_{key} + s_{mac} + s_{signature}$, ein Indextag mit der Größe s_{tag} und die Münzsignatur mit der Größe $s_{signature}$.
2. **Überprüfen der Annahmeerklärung.**
   (a) Die Bank überprüft die Signatur der Münze und die Münzsignatur in Zeit $2 \cdot t_{verify}$ und entschlüsselt das Indextag in Zeit $t_{decrypt}$.
3. **Erzeugen des Einzahlzertifikates.**
   (a) Die Bank erstellt das Einzahlzertifikat in Zeit $t_{issue(\mathcal{D})}$ und sendet das Einzahlzertifikat mit der Größe $s_{certificate(\mathcal{D})}$ an den Kunden.
   (b) Der Kunde überprüft das Einzahlzertifikat in Zeit $t_{verify\{\mathcal{D})}$.
4. **Überprüfen des Tags.**
   (a) Der Kunde sendet das ausgewählte Tag mit der Größe s_{tag} an die Bank.
   (b) Die Bank entschlüsselt das Tag in Zeit $t_{decrypt}$.

Wir berechnen nun die zu erwartenden Laufzeiten und die Menge der zu kommunizierenden Daten für das Bezahlen mit bzw. Einzahlen von einer Münze. Für diese Berechnung vernachlässigen wir die Zeiten zum Erstellen und Überprüfen sowie die Größe des Einzahlzertifikats.

- Laufzeiten der Operationen des Kunden:

$$t_{payment_c} = t_{signature} \tag{4.4}$$

- Laufzeiten der Operationen der Bank:

$$t_{payment_b} = 2 \cdot (t_{verify} + t_{decrypt}) \tag{4.5}$$

- Kommunikation zwischen Kunde, Händler und Bank:

$$s_{payment} = s_{coin} + s_{signature} + 2 \cdot s_{tag} \tag{4.6}$$

Bezahlen/	**Kunde**		**Bank**		**Gesamt**		**Kommunikation**	
Einzahlen	**512**	**1024**	**512**	**1024**	**512**	**1024**	**512**	**1024**
Pro Münze	9 ms	30 ms	50 ms	174 ms	59 ms	204 ms	306 Byte	498 Byte
Für 10€	0,6 s	2,0 s	3,4 s	11,8 s	4,0 s	13,9 s	20,3 KB	33,1 KB
Gemessen	0,8 s	2,2 s	4,0 s	12,5 s	4,7 s	14,7 s	22,1 KB	35,2 KB
*Overhead**	3 ms	2 ms	9 ms	10 ms	10 ms	12 ms	19 Byte	20 Byte
Für 100€	0,9 s	3,1 s	5,2 s	17,9 s	6,1 s	21,0 s	30,8 KB	50,1 KB
Gemessen	1,2 s	3,3 s	6,0 s	19,0 s	7,1 s	22,3 s	33,3 KB	53,1 KB
*Overhead**	3 ms	2 ms	8 ms	10 ms	10 ms	13 ms	17 Byte	18 Byte

*Overhead pro Münze

Tabelle 4.6: Daten der Vorgänge Be- und Einzahlen

Die Operationen des Kunden und der Bank werden praktisch sequentiell ausgeführt. Daher ergibt sich als Gesamtlaufzeit für einen Bezahlvorgang:

$$t_{payment} = t_{payment_c} + t_{payment_b}$$

Auf dieser Basis berechnen wir die Daten für einen Bezahl- und Einzahlvorgang mit einem 512 Bit Systemparameter und einem 1024 Bit Systemparameter. Die berechneten Werte vergleichen wir anschließend mit den tatsächlich gemessenen Werten. Tabelle 4.6 faßt die berechneten und gemessenen Daten zusammen.

512 Bit Systemparameter

Wir berechnen für die Operationen des Kunden nach Formel 4.4 beim Bezahlen eine Laufzeit von 9 ms pro Münze. Für die Operationen der Bank berechnen wir nach Formel 4.5 für das Einzahlen einer Münze 50 ms Rechenzeit. Nach Formel 4.6 werden pro Münze 306 Byte Daten zwischen Kunde und Bank ausgetauscht.

- Bezahlen mit 10€ (in 68 Münzen):
  - Wir berechnen eine Gesamtlaufzeit von $4,0$ s. Dabei entfallen $0,6$ s Rechenzeit auf den Kunden und $3,4$ s auf die Bank. Insgesamt werden $20,3$ KB Daten übertragen.
  - Wir haben eine Gesamtlaufzeit von $4,7$ s gemessen. Dabei entfielen $0,8$ s Rechenzeit auf den Kunden und $4,0$ s Rechenzeit auf die Bank. Insgesamt wurden $22,1$ KB Daten übertragen.
  - Es ergibt sich pro Münze ein Overhead von 3 ms für den Kunden und von 9 ms für die Bank.
- Bezahlen mit 100€ (in 103 Münzen):

  - Wir berechnen eine Gesamtlaufzeit von $6,1$ s. Dabei entfallen $0,9$ s Rechenzeit auf den Kunden und $5,2$ s auf die Bank. Insgesamt werden $30,8$ KB Daten übertragen.
  - Wir haben eine Gesamtlaufzeit von $7,1$ s gemessen. Dabei entfielen $1,2$ s Rechenzeit auf den Kundenund $5,9$ s auf die Bank. Insgesamt wurden $33,3$ KB Daten übertragen.
  - Es ergibt sich pro Münze ein Overhead von 3 ms für den Kunden und von 8 ms für die Bank.

1024 Bit Systemparameter

Wir berechnen für die Operationen des Kunden nach Formel 4.4 beim Bezahlen eine Laufzeit von 30 ms pro Münze. Für die Operationen der Bank berechnen wir nach Formel 4.5 für das Einzahlen einer Münze 174 ms Rechenzeit. Nach Formel 4.6 werden pro Münze 498 Byte Daten zwischen Kunde und Bank ausgetauscht.

- Bezahlen mit 10€ (in 68 Münzen):
  - Wir berechnen eine Gesamtlaufzeit von $13,9$ s. Dabei entfallen $2,0$ s Rechenzeit auf den Kunden und $11,8$ s auf die Bank. Insgesamt werden $33,1$ KB Daten übertragen.
  - Wir haben eine Gesamtlaufzeit von $14,7$ s gemessen. Dabei entfielen $2,2$ s Rechenzeit auf den Kunden und $12,5$ s auf die Bank. Insgesamt wurden $35,2$ KB Daten übertragen.
  - Es ergibt sich pro Münze ein Overhead von 2 ms für den Kunden und von 10 ms für die Bank.
- Bezahlen mit 100€ (in 103 Münzen):
  - Wir berechnen eine Gesamtlaufzeit von $21,0$ s. Dabei entfallen $3,1$ s Rechenzeit auf den Kunden und $17,9$ s auf die Bank. Insgesamt werden $50,1$ KB Daten übertragen.
  - Wir haben eine Gesamtlaufzeit von $22,3$ s gemessen. Dabei entfielen $3,3$ s Rechenzeit auf den Kunden $19,0$ s auf die Bank. Insgesamt wurden $53,1$ KB Daten übertragen.
  - Es ergibt sich pro Münze ein Overhead von 2 ms für den Kunden und von 10 ms für die Bank.

4.2.3 Zurückgeben

An dem Vorgang ZURÜCKGEBEN sind Kunde und Bank beteiligt. Folgende Schritte werden durchgeführt:

1. **Erstellen einer Rückgabesignatur.**

   (a) Der Kunde erstellt eine Rückgabesignatur in Zeit $t_{signature}$ und sendet folgende Daten an die Bank: Die Seriennummer mit der Größe $s_{serial} = s_{key} + s_{mac}$, die blinde Münze mit der Größe $s_{blindcoin} = s_{commitment} + s_{blindmessage}$, den Blendparameter mit der Größe s_{bp}, den Rückgabeschlüssel mit der Größe $s_{authkey}$ und die Rückgabesignatur mit der Größe $s_{signature}$.

2. **Überprüfen der Rückgabe.**

   (a) Die Bank überprüft die Rückgabesignatur in Zeit t_{verify}, sowie den Authentisierungscode in Zeit t_{mac} und entblendet die Münze in Zeit $t_{blinding}$.

Wir berechnen nun die zu erwartenden Laufzeiten und die Menge der zu kommunizierenden Daten für das Zurückgeben einer Münze.

- Laufzeiten der Operationen des Kunden:

$$t_{redemption_c} = t_{signature} \tag{4.7}$$

- Laufzeiten der Operationen der Bank:

$$t_{redemption_b} = t_{verify} + t_{mac} + t_{blinding} \tag{4.8}$$

- Kommunikation zwischen Kunde und Bank:

$$s_{redemption} = s_{key} + s_{mac} + s_{commitment} + s_{blindmessage} + s_{bp} + s_{authkey} + s_{signature} \tag{4.9}$$

Die Operationen des Kunden und der Bank werden sequentiell ausgeführt. Daher ergibt sich für einen Rückgabevorgang folgende Gesamtlaufzeit:

$$t_{redemption} = t_{redemption_c} + t_{redemption_b}$$

Auf dieser Basis berechnen wir Daten für einen Rückgabevorgang mit einem 512 Bit Systemparameter und einem 1024 Bit Systemparameter. Die berechneten Werte vergleichen wir anschließend mit den tatsächlich gemessenen Werten. Tabelle 4.7 faßt die berechneten und gemessenen Daten zusammen.

Zurückgeben	Kunde		Bank		Gesamt		Kommunikation	
	512	**1024**	**512**	**1024**	**512**	**1024**	**512**	**1024**
Pro Münze	9 ms	30 ms	35 ms	117 ms	44 ms	147 ms	282 Byte	410 Byte
Für 10€	0,6 s	2,0 s	2,4 s	8,0 s	3,0 s	10,0 s	18,7 KB	27,2 KB
Gemessen	1,3 s	2,3 s	2,7 s	8,2 s	4,0 s	10,6 s	20,0 KB	28,7 KB
*Overhead**	10 ms	4 ms	5 ms	4 ms	15 ms	9 ms	12 Byte	12 Byte
Für 100€	0,9 s	3,1 s	3,6 s	12,1 s	4,5 s	15,1 s	28,4 KB	41,2 KB
Gemessen	1,4 s	3,6 s	4,0 s	12,4 s	5,4 s	16,0 s	30,2 KB	43,5 KB
*Overhead**	5 ms	5 ms	4 ms	3 ms	8 ms	8 ms	11 Byte	12 Byte

*Overhead pro Münze

Tabelle 4.7: Daten des Vorgangs Zurückgeben

512 Bit Systemparameter

Wir berechnen für die Operationen des Kunden nach Formel 4.7 beim Zurückgeben eine Laufzeit von 9 ms pro Münze. Für die Operationen der Bank berechnen wir nach Formel 4.8 für das Zurücknehmen einer Münze 35 ms Rechenzeit. Nach Formel 4.9 werden pro Münze 282 Byte Daten zwischen Kunde und Bank ausgetauscht.

- Zurückgeben von 10€ (in 68 Münzen):
  - Wir berechnen eine Gesamtlaufzeit von 3,0 s. Dabei entfallen 0,6 s Rechenzeit auf den Kunden und 2,4 s auf die Bank. Insgesamt werden 18,7 KB Daten übertragen.
  - Wir haben eine Gesamtlaufzeit von 4,0 s gemessen. Dabei entfielen 1,3 s Rechenzeit auf den Kunden und 2,7 s auf die Bank. Insgesamt wurden 20,0 KB an Daten übertragen.
  - Es ergibt sich pro Münze ein Overhead von 10 ms für den Kunden und von 5 ms für die Bank.
- Zurückgeben von 100€ (in 103 Münzen):
  - Wir berechnen eine Gesamtlaufzeit von 4,5 s. Dabei entfallen 0,9 s Rechenzeit auf den Kunden und 3,6 s auf die Bank. Insgesamt werden 28,4 KB Daten übertragen.
  - Wir haben eine Gesamtlaufzeit von 5,4 s gemessen. Dabei entfielen 1,4 s Rechenzeit auf den Kunden und 4,0 s auf die Bank. Insgesamt wurden 30,2 KB an Daten übertragen.
  - Es ergibt sich pro Münze ein Overhead von 5 ms für den Kunden und von 4 ms für die Bank.

1024 Bit Systemparameter

Wir berechnen für die Operationen des Kunden nach Formel 4.7 beim Zurückgeben eine Laufzeit von 30 ms pro Münze. Für die Operationen der Bank berechnen wir nach Formel 4.8 für das Zurücknehmen einer Münze 117 ms Rechenzeit. Nach Formel 4.9 werden pro Münze 410 Byte Daten zwischen Kunde und Bank ausgetauscht.

- Zurückgeben von 10€ (in 68 Münzen):
  - Wir berechnen eine Gesamtlaufzeit von $10,0$ s. Dabei entfallen $2,0$ s Rechenzeit auf den Kunden und $8,0$ s auf die Bank. Insgesamt werden $27,2$ KB Daten übertragen.
  - Wir haben eine Gesamtlaufzeit von $10,6$ s gemessen. Dabei entfielen $2,3$ s Rechenzeit auf den Kunden und $8,2$ s auf die Bank. Insgesamt wurden $28,7$ KB an Daten übertragen.
  - Es ergibt sich pro Münze ein Overhead von 4 ms für den Kunden und von 4 ms für die Bank.
- Zurückgeben von 100€ (in 103 Münzen):
  - Wir berechnen eine Gesamtlaufzeit von $15,1$ s. Dabei entfallen $3,1$ s Rechenzeit auf den Kunden und $12,1$ s auf die Bank. Insgesamt werden $41,2$ KB Daten übertragen.
  - Wir haben eine Gesamtlaufzeit von $16,0$ s gemessen. Dabei entfielen $3,6$ s Rechenzeit auf den Kunden und $12,4$ s auf die Bank. Insgesamt wurden $43,5$ KB an Daten übertragen.
  - Es ergibt sich pro Münze ein Overhead von 5 ms für den Kunden und von 3 ms für die Bank.

4.2.4 Überprüfen

An dem Vorgang ÜBERPRÜFEN ist ausschließlich der Kunde beteiligt. Folgende Schritte werden durchgeführt:

1. **Entschlüsselung der Indextags.** Der Kunde entschlüsselt das Indextag in Zeit $t_{decrypt}$ und vergleicht die entschlüsselte Markierung mit der Null- und Einsmarkierung.
2. **Überprüfung von Münzverfolgung.** Der Kunde entschlüsselt das Markierungstag in Zeit $t_{decrypt}$ und vergleicht die entschlüsselte Markierung mit der Standardmarkierung.
3. **Überprüfung von Kundenverfolgung.** Der Kunde überprüft, ob er das Markierungstag oder das Identitätstag beim Bezahlen herausgegeben hat.

Überprüfen	**Kunde**	
	512	**1024**
Pro Münze	16 ms	58 ms
Für 10€	1,1 s	3,9 s
Gemessen	1,4 s	4,4 s
*Overhead**	5 ms	7 ms
Für 100€	1,6 s	6 s
Gemessen	2,1 s	6,6 s
*Overhead**	4 ms	6 ms

*Overhead pro Münze

Tabelle 4.8: Daten des Vorgangs Überprüfen

Wir berechnen nun die zu erwartende Laufzeit für das Überprüfen einer Münze.

$$t_{audit} = 2 \cdot t_{decrypt} \tag{4.10}$$

Auf dieser Basis berechnen wir Daten für einen Überprüfungsvorgang mit einem 512 Bit Systemparameter und einem 1024 Bit Systemparameter. Die berechneten Werte vergleichen wir anschließend mit den tatsächlich gemessenen Werten. Tabelle 4.8 faßt die berechneten und gemessenen Daten zusammen.

512 Bit Systemparameter

Wir berechnen für die Operationen des Kunden nach Formel 4.10 beim Überprüfen eine Laufzeit von 16 ms pro Münze.

- Überprüfen von 10€ (in 68 Münzen):
  - Wir berechnen eine Gesamtlaufzeit von 1,1 s.
  - Wir haben eine Gesamtlaufzeit von 1,4 s gemessen.
  - Es ergibt sich pro Münze ein Overhead von 5 ms.
- Überprüfen von 100€ (in 103 Münzen):
  - Wir berechnen eine Gesamtlaufzeit von 1,6 s.
  - Wir haben eine Gesamtlaufzeit von 2,1 s gemessen.
  - Es ergibt sich pro Münze ein Overhead von 4 ms.

1024 Bit Systemparameter

Wir berechnen für die Operationen des Kunden nach Formel 4.10 beim Überprüfen eine Laufzeit von 58 ms pro Münze.

- Überprüfen von 10€ (in 68 Münzen):
  - Wir berechnen eine Gesamtlaufzeit von $3,9$ s.
  - Wir haben eine Gesamtlaufzeit von $4,4$ s gemessen.
  - Es ergibt sich pro Münze ein Overhead von 7 ms.
- Überprüfen von 100€ (in 103 Münzen):
  - Wir berechnen eine Gesamtlaufzeit von $6,0$ s.
  - Wir haben eine Gesamtlaufzeit von $6,6$ s gemessen.
  - Es ergibt sich pro Münze ein Overhead von 6 ms.

4.3 Effizienz und Skalierbarkeit

FlexiCash stellt auf Kundenseite keine besonderen Anforderungen an die verwendete Hardware. Unsere Berechnungen und Messungen haben gezeigt, daß FlexiCash selbst auf wenig leistungsstarker Hardware, wie z.B. auf zukünftigen mobilen Computern gut funktioniert. Auch die zu übertragende Datenmenge stellt selbst für schmalbandige Übertragungswege kein Problem dar.

Der für die Bank notwendige Aufwand wächst linear mit der Anzahl der gleichzeitig zu bedienenden Kunden und Händler. Das bedeutet, daß die Bank leistungsfähigere Hardware benötigt. Aus diesem Grund vergleichen wir zum Abschluß den notwendigen Aufwand der Bank für die Vorgänge ABHEBEN und EINZAHLEN und ZURÜCKGEBEN bei Verwendung von FlexiCash mit dem vollständig anonymen Zahlungssystem aus Abschnitt 1.2. Wir zeigen, daß FlexiCash mit einem ähnlichen Aufwand behaftet ist, wie das vollständig anonyme Zahlungssystem. FlexiCash hat jedoch den Vorteil, daß alle anonymitätsbezogenen Probleme gelöst werden können.

Abheben

- Bei dem vollständig anonymen Zahlungssystem muß die Bank pro Münze ein Commitment erstellen. Alle anderen Operationen können wir vernachlässigen.
- Bei FlexiCash muß die Bank neben dem Commitment drei Tags erstellen. Das Erstellen eines Tags benötigt etwa die gleiche Zeit, wie das Erstellen des Commitments.

Der Aufwand der Bank ist beim Abheben mit FlexiCash etwa viermal so hoch wie bei dem vollständig anonymen Zahlungssystem. Commitments und Tags können jedoch von der Bank vorberechnet werden. In diesem Fall ist der Aufwand bei beiden Zahlungssystemen äquivalent.

Einzahlen

- Bei dem vollständig anonymen Zahlungssystem muß die Bank pro Münze zwei Signaturen verifizieren.
- Bei FlexiCash muß die Bank neben den zwei Verifikationen zusätzlich zwei Markierungen entschlüsseln. Bei einer gewöhnlichen Implementierung der Verifikation (d.h. ohne Multiexponentationen), beträgt der Aufwand für das Entschlüsseln eines Tags die Hälfte des Aufwands einer Verifikation.

Der Aufwand der Bank ist beim Einzahlen mit FlexiCash um etwa 50% höher, als bei dem vollständig anonymen Zahlungssystem.

Zurückgeben

- Bei dem vollständig anonymen Zahlungssystem muß die Bank pro Münze zwei Signaturen verifizieren.
- Bei FlexiCash muß die Bank pro Münze eine Signatur verifizieren und eine Blendung durchführen. Der Aufwand für das Blenden entspricht dem Aufwand einer Verifikation.

Der Aufwand der Bank ist beim Zurückgeben mit FlexiCash äquivalent zu dem vollständig anonymen Zahlungssystem. Das Rückgabeprotokoll von FlexiCash ist jedoch wesentlich sicherer (vgl. Bankraub in Abschnitt 1.3.3 und 2.4). Das Rückgabeprotokoll von FlexiCash ist praktisch auf jedes münzbasierte Zahlungssystem übertragbar.

Ausblick

Wir haben in dieser Dissertation ein neues anonymes elektronisches Zahlungssystem vorgestellt, das – verglichen mit Bargeld – sowohl stärkere Anonymität, als auch wesentlich bessere Deanonymisierungsmechanismen bietet.

Der in FlexiCash vorhandene Kontrollmechanismus für Deanonymisierungen existiert in keinem anderen Zahlungssystem: In regelmäßigen Abständen können vorhergehende Zahlungen durch den Zahlenden selbst überprüft werden. Bei dieser Überprüfung kann der Zahlende feststellen, ob in der Vergangenheit die Anonymität einer Zahlung aufgehoben wurde. In diesem Fall muß eine richterliche Anordnung dafür vorhanden sein. Andernfalls wurde beweisbar eine illegale Deanonymisierung durchgeführt, die rechtlich verfolgt werden kann. Wird bei einer Überprüfung keine Deanonymisierung festgestellt, hat der Zahlende die Garantie, daß diese Zahlungen tatsächlich anonym durchgeführt wurden und anonym bleiben werden. Kein anderes Zahlungssystem mit eingeschränkter Anonymität kann diese starke Anonymität garantieren.

In einer Hinsicht ist Bargeld unserem Zahlungssystem jedoch überlegen. FlexiCash ist in Online-Zahlungssystem, d.h. die Bank ist an jeder Zahlung direkt beteiligt. Das ist einerseits von Vorteil, da nur auf diese Weise das Problem des mehrfachen Ausgebens einer Münze verhindert werden kann. Andererseits ist es auch von Nachteil, da Zahlungen nicht möglich sind, wenn die Bank nicht erreichbar ist bzw. eine Kommunikation mit der Bank zu aufwendig ist (z.B. Fahrkartenautomaten). Für diesen Fall wurden Offline-Zahlungssysteme entwickelt, in denen ein Teil der Bankfunktionalität in sichere Hardware verlagert wird.

Zum Abschluß möchten wir auf eine Offline-Variante von FlexiCash hinweisen. In [KV02] beschreiben wir ein Zahlungssystem, das auf den gleichen Ideen wie FlexiCash basiert, aber die Bank während der Zahlung nicht benötigt. Allerdings unterscheidet sich die Realisierung erheblich von dem hier vorgestellten Zahlungssystem.

Literaturverzeichnis

[ASW97] N. Asokan, Matthias Schunter und Michael Waidner. Optimistic Protocols for Fair Exchange. In *4th ACM Conference on Computer and Communications Security – CCS '97*, S. 6–17. ACM Press, Zürich, Switzerland, 1997.

[BGK95] Ernie Brickell, Peter Gemmell und David Kravitz. Trustee-based Tracing Extensions to Anonymous Cash and the Making of Anonymous Change. In *6th Annual Symposium on Discrete Algorithms - SODA '95*, S. 457–466. ACM Press, 1995.

[Bon98] Dan Boneh. The Decision Diffie-Hellman Problem. In *Algorithmic Number Theory – ANTS-III*, Lecture Notes in Computer Science 1423, S. 48–63. Springer-Verlag, 1998.

[Bra93] Stefan Brands. Untraceable Off-Line Cash in Wallets with Observers. In *Advances in Cryptology – CRYPTO '93*, Lecture Notes in Computer Science 773, S. 302–318. Springer-Verlag, 1993.

[CFN88] David Chaum, Amos Fiat und Moni Naor. Untraceable Electronic Cash. In *Advances in Cryptology – CRYPTO '88*, Lecture Notes in Computer Science 401, S. 319–327. Springer-Verlag, 1988.

[Cha81] David Chaum. Untraceable Electronic Mail, Return Adresses and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[Cha83] David Chaum. Blind Signatures for Untraceable Payments. In *Advances in Cryptology – CRYPTO '82*, S. 199–203. Plenum, 1983.

[CMS96] Jan Camenisch, Ueli Maurer und Markus Stadler. Digital Payment Systems with Passive Anonymity-Revoking Trustees. In *Computer Security – ESORICS '96*, Lecture Notes in Computer Science 1146, S. 31–43. Springer-Verlag, 1996.

[CMS97] Jan Camenisch, Ueli Maurer und Markus Stadler. Digital Payment Systems with Passive Anonymity-Revoking Trustees. *Journal of Computer Security*, 5(1), 1997.

[DDN91] Danny Dolev, Cynthia Dwork und Moni Naor. Non-Malleable Cryptography. In *23rd annual ACM Symposium on Theory of Computing*, S. 542–552. ACM Press, 1991.

[DFTY97] George Davida, Yair Frankel, Yiannis Tsiounis und Moti Yung. Anonymity Control in E-Cash Systems. In *Financial Cryptography – FC '97*, Lecture Notes in Computer Science 1318, S. 1–16. Springer-Verlag, 1997.

[ElG85] Taher ElGamal. A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[FS90] Uriel Feige und Adi Shamir. Witness Indistinguishable and Witness Hiding Protocols. In *22nd Symposium on Theory of Computing – STOC '90*, S. 416–426. ACM Press, 1990.

[FTY96] Yair Frankel, Yiannis Tsiounis und Moti Yung. "Indirect Discourse Proofs": Achieving Efficient Fair Off-Line E-Cash. In *Advances in Cryptology – ASIACRYPT '96*, Lecture Notes in Computer Science 1163, S. 286–300. Springer-Verlag, 1996.

[GM82] Shafi Goldwasser und Silvio Micali. Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, S. 365–377. ACM Press, 5–7 Mai 1982.

[GM84] Shafi Goldwasser und Silvio Micali. Probabilistic Encryption. *Journal of Computer Security*, 28:270–299, 1984.

[GQ88] Louis C. Guillou und Jean-Jacques Quisquater. A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission And Memory. In *Advances in Cryptology – EUROCRYPT '88*, Lecture Notes in Computer Science 330, S. 123–128. Springer-Verlag, 1988.

[Jak99] Markus Jakobsson. Mini-Cash: A Minimalistic Approach to E-Commerce. In *Public Key Cryptography – PKC'99*, Lecture Notes in Computer Science 1560, S. 122–135. Springer-Verlag, 1999.

[JLO97] Ari Juels, Michael Luby und Rafail Ostrovsky. Security of Blind Digital Signatures. In *Advances in Cryptology – CRYPTO '97*, Lecture Notes in Computer Science 1294, S. 150–164. Springer-Verlag, 1997.

[JM98] Markus Jakobsson und David M'Raïhi. Mix-based Electronic Payments. In *Annual International Workshop on Selected Areas in Cryptography – SAC '98*, Lecture Notes in Computer Science 1556, S. 157–173. Springer-Verlag, 1998.

[JM99] Markus Jakobsson und J. Müller. Improved Magic Ink Signatures Using Hints. In *Financial Cryptography – FC '99*, Lecture Notes in Computer Science 1648, S. 253–267. Springer-Verlag, 1999.

[JY96] Markus Jakobsson und Moti Yung. Revokable and Versatile Electronic Money. In *3rd ACM Conference on Computer and Communications Security – CCS '96*, S. 76–87. ACM Press, 1996.

[JY97] Markus Jakobsson und Moti Yung. Distributed "Magic Ink" Signatures. In *Advances in Cryptology – EUROCRYPT '97*, Lecture Notes in Computer Science 1233, S. 450–464. Springer-Verlag, 1997.

[Küg01] Dennis Kügler. Enabling Privacy Protection in E-Commerce Applications. In *Electronic Commerce, Second International Workshop – WELCOM 2001*, Lecture Notes in Computer Science 2232, S. 127–138. Springer-Verlag, 2001.

[KV01a] Dennis Kügler und Holger Vogt. Marking: A Privacy Protecting Approach Against Blackmailing. In *Public Key Cryptography – PKC 2001*, Lecture Notes in Computer Science 1992, S. 137–152. Springer-Verlag, 2001.

[KV01b] Dennis Kügler und Holger Vogt. Fair Tracing without Trustees. In *Financial Cryptography – FC 2001*, Lecture Notes in Computer Science 2339, S. 136–148. Springer-Verlag, 2001.

[KV01c] Dennis Kügler und Holger Vogt. Auditable Tracing with Unconditional Anonymity. In *Proceedings of the 2nd International Workshop on Information Security Applications (WISA 2001)*, S. 151–163. Seoul, Korea, 2001.

[KV01d] Dennis Kügler und Holger Vogt. Unsichtbare Markierungen in elektronischem Geld. In *Kommunikationssicherheit – Schwerpunkt Internet*, S. 262–271. Vieweg Verlag, 2001.

[KV02] Dennis Kügler und Holger Vogt. Offline Payments with Auditable Tracing. In *Financial Cryptography – FC 2002*. Preproceedings, 2002.

[Lab93] RSA Laboratories. PKCS#1: RSA Encryption Standard. RSA Laboratories Technical Note, 1993.

[Nat94] National Institute of Standards and Technology (NIST). The Digital Signature Standard. FIPS PUB 186, 1994.

[Nat95] National Institute of Standards and Technology (NIST). Secure Hash Standard. FIPS PUB 180-1, 1995.

[Oka92] Tatsuaki Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In *Advances in Cryptology – CRYPTO '92*, Lecture Notes in Computer Science 740, S. 31–53. Springer-Verlag, 1992.

[Pai99] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology – EUROCRYPT '99*, Lecture Notes in Computer Science 1592, S. 223–238. Springer-Verlag, 1999.

[PK01] Andreas Pfitzmann und Marit Köhntopp. Anonymity, Unobservability, and Pseudonymity – A Proposal for Terminology. In *Designing Privacy Enhancing Technologies – International Workshop on Design Issues in Anonymity and Unobservability 2000*, Lecture Notes in Computer Science 2009, S. 1–9. Springer-Verlag, 2001.

[Poi98] David Pointcheval. Strengthened Security for Blind Signatures. In *Advances in Cryptology – EUROCRYPT '98*, Lecture Notes in Computer Science 1403, S. 391–405. Springer-Verlag, 1998.

[Poi01] David Pointcheval. Self-Scrambling Anonymizers. In *Financial Cryptography – FC 2000*, Lecture Notes in Computer Science 1962, S. 259–275. Springer-Verlag, 2001.

[PP97] Holger Petersen und Guillaume Poupard. Efficient Scalable Fair Cash with Off-line Extortion Prevention. In *International Conference on Information and Communications Security – ICICS '97*, Lecture Notes in Computer Science 1334, S. 463–477. Springer-Verlag, 1997.

[PS96] David Pointcheval und Jacques Stern. Provably Secure Blind Signature Schemes. In *Advances in Cryptology – ASIACRYPT '96*, Lecture Notes in Computer Science 1163, S. 252–265. Springer-Verlag, 1996.

[PS00] David Pointcheval und Jacques Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 2000.

[PS01] Birgit Pfitzmann und Ahmad-Reza Sadeghi. Self-Escrowed Cash Against User Blackmailing. In *Financial Cryptography – FC 2000*, Lecture Notes in Computer Science 1962, S. 42–52. Springer-Verlag, 2001.

[RIP] Regulation of Investigatory Powers Act 2000.

[RSA78] Ronald L. Rivest, Adi Shamir und Leonard Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[Sch91] Claus Peter Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[Sch97] Berry Schoenmakers. Security Aspects of the ecash Payment System. In *COSIC '97 Course*, Lecture Notes in Computer Science 1528, S. 338–352. Springer-Verlag, 1997.

[Sch01a] Claus Peter Schnorr. Security of DL-Encryption and Signatures Against Generic Attacks, a Survey. In *Public-Key Cryptography and Computational Number Theory 2000*. Walter De Gruyter, 2001.

[Sch01b] Claus Peter Schnorr. Security of Blind Discrete Log Signatures Against Interactive Attacks. In *3rd International Conference On Information And Communication Security – ICICS 2001*, Lecture Notes in Computer Science 2229. Springer-Verlag, 2001.

[SPC95] Markus Stadler, Jean-Marc Piveteau und Jan Camenisch. Fair Blind Signatures. In *Advances in Cryptology – EUROCRYPT '95*, Lecture Notes in Computer Science 921, S. 209–219. Springer-Verlag, 1995.

[STS99a] Tomas Sander und Amnon Ta-Shma. Auditable, Anonymous Electronic Cash. In *Advances in Cryptology – CRYPTO '99*, Lecture Notes in Computer Science 1666, S. 555–572. Springer-Verlag, 1999.

[STS99b] Tomas Sander und Amnon Ta-Shma. Flow Control: A New Approach for Anonymity Control in Electronic Cash Systems. In *Financial Cryptography – FC '99*, Lecture Notes in Computer Science 1648, S. 46–61. Springer-Verlag, 1999.

[STSY01] Tomas Sander, Amnon Ta-Shma und Moti Yung. Blind, Auditable Membership Proofs. In *Financial Cryptography – FC 2000*, Lecture Notes in Computer Science 1962, S. 53–71. Springer-Verlag, 2001.

[TDD] Teledienstedatenschutzgesetz. Artikel 2 des Informations- und Kommunikationsdienste-Gesetz.

[vSN92] B. von Solms und David Naccache. On Blind Signatures and Perfect Crimes. *Computers and Security*, 11(6):581–583, 1992.

Index

Curriculum Vitae

18.06.1973	Geboren in Hamburg
1979-1983	Grundschule in Auerbach
1983-1992	Altes Kurfürstliches Gymnasium Bensheim
1992-1998	Diplomstudiengang Informatik an der Technischen Hochschule Darmstadt
1998-2002	Promotion an der Technischen Hochschule Darmstadt
seit 2002	Referent im Bundesamt für Sicherheit in der Informationstechnik