Kapitel 1

Elektronisches Bargeld, ein erstes Beispiel

Auch im Zeitalter des bargeldlosen Bezahlens besitzt das klassische Bargeld durchaus noch seine Berechtigung. Es ermöglicht eine einfache, schnelle, unverbindliche und kostengünstige Abwicklung des Bezahlvorgangs. Bei hohen Beträgen wird Bargeld wegen des Verlust- und Diebstahlrisikos selten verwendet. Hier bietet der bargeldlose Zahlungsverkehr klare Vorteile. Wegen der Abwicklung über eine Bank oder ein Kreditkarteninstitut und der damit verbundenen Dokumentation kann ein derartiger Bezahlvorgang später geprüft und rekonstruiert werden, z. B. anhand eines Kontoauszuges.

Eine neue Problemstellung ergibt sich im Electronic Commerce, das heißt beim Bezahlen von Waren, Dienstleistungen oder Informationen, die im Internet angeboten werden. Die Kosten für viele dieser Dienste bewegen sich im Bereich von wenigen Pfennigen (Micro-Payment). Daher ist eine bargeldlose Transaktion wie zum Beispiel eine Überweisung oder die Belastung einer Kreditkarte unrentabel. Auch möchte der Kunde für die einmalige oder seltene Nutzung eines Dienstes eventuell keine persönlichen Daten oder Kontodaten angeben. Hierzu bietet sich das Bezahlen mit elektronischen Münzen an. Der Bezahlvorgang besteht nur aus dem Übertragen von einigen elektronischen Münzen, das heißt Bitfolgen zwischen Kunde und Händler. Wie beim klassischen Bargeld werden zwischen den beiden Partnern Objekte – nämlich elektronische Münzen – ausgetauscht. Gegebenenfalls wird auch Wechselgeld zurückgegeben, allerdings werden Kunde und Händler damit nicht belastet. Wie beim klassischen Bargeld sollte das Bezahlen anonym erfolgen, gleichzeitig aber sicher gegen Betrug sein.

Das Bezahlen mit elektronischen Münzen effizient und sicher zu gestalten, ist eine Aufgabe der modernen Kryptographie. Anhand einiger einfacher Ideen soll nun exemplarisch gezeigt werden, wie die im Buch beschriebenen kryptographi-

schen Protokolle und Algorithmen hierzu verwendet werden. Die technischen Details folgen dann in Kapitel 9, wenn die Voraussetzungen dafür geschaffen sind. Bevor wir uns jedoch auf den faszinierenden, nicht immer ganz einfachen Weg zum Verständnis dieser Techniken machen, wollen wir am Beispiel des elektronischen Bargeldes ohne Theorie einen ersten Eindruck von den teilweise genialen Protokollen und der Mächtigkeit der modernen Kryptographie vermitteln.

Wir werden schrittweise ein Protokoll mit interessanten Eigenschaften vorstellen. Es wurde von David Chaum, dem Gründer der holländischen Firma Digicash entwickelt [Cha85, Cha92] und patentiert.

Die an dem Verfahren beteiligte Bank nennen wir E-Bank und als Zahlungsmittel werden E-Münzen benutzt. Eine solche E-Münze besteht letztlich aus einer (endlichen) Folge von Bytes, analog zu einem Geldschein, der ein spezielles Stück bedrucktes Papier darstellt. Wir versuchen's zuerst mal ganz naiv:

Protokoll Nr. 1

Die E-Bank erzeugt auf ihrem PC eine Datei mit dem Inhalt: „E-Münze, Wert: 5 €", wie in Abbildung 1.1 dargestellt. Dies führt natürlich sofort zur Inflation, wenn die Kunden den Betrag ihrer E-Münzen beliebig ändern.

Protokoll Nr. 2

Wenn die E-Bank jedoch die E-Münze mit einer Unterschrift versieht, die nur sie und kein anderer erstellen kann, so kann der Kunde, der die Münze auf seinem Rechner speichert, den Betrag nicht mehr abändern. Falls er das versucht, wird die digitale Signatur der Bank ungültig.[1] Er kann jedoch immer noch betrügen, indem er einfach beliebig viele Kopien der E-Münze erzeugt (Abbildung 1.1). Dies wird verhindert im

Protokoll Nr. 3

Wie in Abbildung 1.1 dargestellt, vergibt die Bank nun für jede Münze eine eindeutige Seriennummer und signiert den gesamten Text bestehend aus Betrag und Seriennummer[2]. Versucht nun jemand, Kopien einer derartigen Münze herzustellen, so wird der Betrug erkannt. Die E-Bank protokolliert nämlich in einer zentralen Datenbank alle eingegangenen Seriennummern und sobald mindestens zwei Münzen mit der gleichen Seriennummer zur E-Bank zurückkommen werden Hausdetektiv und Staatsanwalt benachrichtigt.

[1] Dies ist ganz analog zu einem unterschriebenen Vertrag, der nicht mehr abgeändert werden *darf*. Bei digitalen Unterschriften ist das Ändern jedoch nicht mehr *möglich*.

[2] In realen Implementierungen wird die Bank weitere Informationen, wie z. B. den Namen der Bank und das Datum, auf der E-Münze speichern. Wir beschränken uns hier jedoch auf die zum Verständnis wesentlichen Daten.

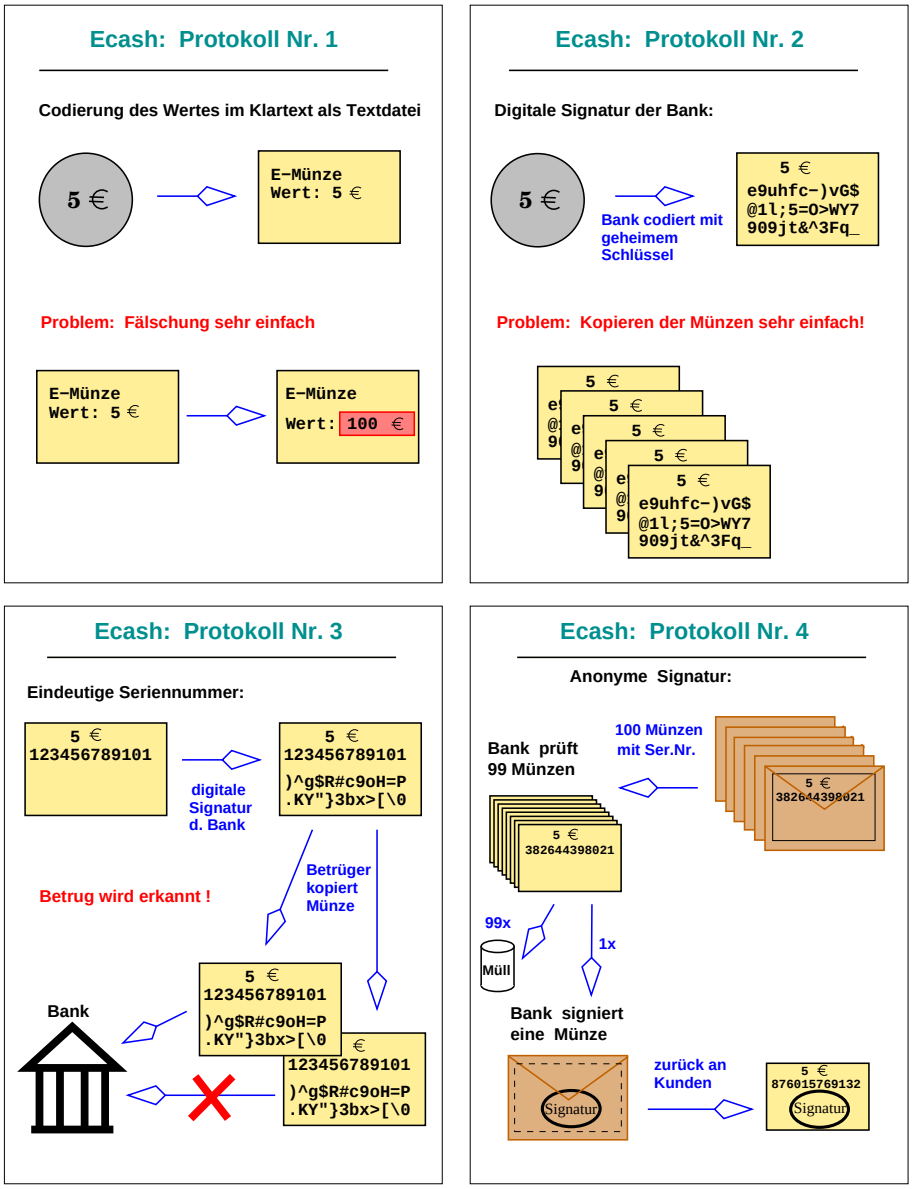

Abbildung 1.1: Protokolle zum Erzeugen einer E-Münze

Seriennummer	Ausgabe	Kunde	Kto.-Nr.	Händler	Rücklauf	Betrag
123456789101	12.2.2001	Maier	7654321	Otto Versand	14.2.2001	50 €
123456789102	12.2.2001	Maier	7654321	Otto Versand	14.2.2001	20 €
123456789103	12.2.2001	Maier	7654321	Otto Versand	14.2.2001	8 €
123456789104	12.2.2001	Maier	7654321	Otto Versand	14.2.2001	0.90 €
123456789105	12.2.2001	Maier	7654321	amazon.de	17.2.2001	20 €
123456789106	12.2.2001	Maier	7654321	amazon.de	17.2.2001	2 €
123456789107	15.2.2001	Huber	0054322	Frisör Kurz	15.2.2001	20 €
123456789108	15.2.2001	Huber	0054322	Frisör Kurz	15.2.2001	20 €
123456789109	15.2.2001	Huber	0054322	Frisör Kurz	15.2.2001	5 €
123456789110	15.2.2001	Huber	0054322	Frisör Kurz	15.2.2001	1 €
123456789111	15.2.2001	Huber	0054322	Tankst. Sprit	16.2.2001	100 €
123456789112	15.2.2001	Huber	0054322	Tankst. Sprit	16.2.2001	2 €
123456789113	15.2.2001	Huber	0054322	Tankst. Sprit	16.2.2001	2 €
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Abbildung 1.2: Beispiel einer möglichen Datenbank von Transaktionen der Kunden der E-Bank

Dieses Protokoll ist sicher, denn jeder Betrug wird erkannt. Es hat aber noch eine Schwäche. Die Anonymität ist nicht gewährleistet, denn die Bank kann aufgrund der Seriennummern ein perfektes Profil jedes Kunden erstellen (siehe Abbildung 1.2). Das Problem wird offensichtlich durch die Seriennummern verursacht, auf die wir jedoch aus Sicherheitsgründen nicht verzichten können.

Protokoll Nr. 4

Den Ausweg aus dem Dilemma lieferte David Chaum [Cha85] mit den von ihm erfundenen *blinden Signaturen*. Wie in Abbildung 1.1 dargestellt, erzeugt nun der Kunde seine E-Münzen selbst. Um eine gültige 5-€ E-Münze zu erhalten, generiert sein PC hundert Dateien, in die jeweils der Text "5 €" sowie eine große zufällig erzeugte Seriennummer geschrieben werden. Die Seriennummer muss so groß sein, dass die Wahrscheinlichkeit für das zufällige Erzeugen von zwei gleichen Nummern (weltweit) sehr klein ist. Nun bittet er die Bank, eine dieser hundert Münzen blind, das heißt ohne Erkennen von Betrag und Seriennummer, zu signieren. Die Bank wird natürlich nur dann blind signieren, wenn sie sicher ist, dass der Betrag auf der Münze wirklich 5 € ist. Daher wählt sie zufällig 99 der 100 Münzen, die der Kunde auspacken und offenlegen muss. Falls der Betrag 99 mal stimmt, signiert die Bank die letzte Münze blind. Das hierzu benutzte Verfahren verwendet zahlentheoretische Eigenschaften von asymmetrischer Verschlüsselung, die in Kapitel 9 beschrieben werden. Daher beschreiben wir hier das Verfahren nur grob in Analogie zu Geldscheinen aus bedrucktem Papier.

Der Kunde erstellt also 100 Fünfeuroscheine mit Betrag und Seriennummer, packt jeden in einen eigenen Umschlag und legt in den Umschlag über den Geldschein ein Kohlepapier. Die Bank signiert nun den von ihr ausgewählten Geldschein blind, indem sie ihren Stempel aus dem Tresor holt und den

Geldschein durch den Umschlag stempelt. Das Kohlepapier hinterlässt auf dem Schein dann den Stempelabdruck. Der Kunde erhält den signierten (gestempelten) Geldschein zurück, packt ihn aus und kann nun damit einkaufen gehen, ohne dass die Bank eine Chance hat, seine Einkäufe zu überwachen. Der Kunde oder auch der Händler kann versuchen, die gültige Münze zu kopieren. Die Bank wird jedoch den Betrug erkennen, weil sie die Seriennummern aller eingehenden Münzen mit den schon eingegangenen in ihrer Datenbank vergleicht. Das Protokoll ist nun also anonym und sicher zugleich.

Ein kleines Problem bleibt jedoch noch zu lösen. Versucht nämlich der Kunde oder der Händler Betrug durch Kopieren der E-Münze, so weiß die Bank zwar, dass der Betrug versucht wurde. Sie weiß jedoch nicht, wer der Betrüger war. David Chaum hat aber auch dieses Problem durch eine elegante Verfeinerung des Protokolls gelöst, die jedoch erst in Kapitel 9 beschrieben werden kann. Hier sei nur so viel verraten: Kopiert der Kunde den Geldschein, so legt die Bank beide eingegangenen Geldscheine übereinander, hält sie gegen das Licht und kann nun den Namen des Betrügers lesen. Ein Geldschein alleine verrät jedoch nichts über die Identität seines Erzeugers.

1.1 Übungen

Aufgabe 1.1

a) Ein Betrüger möchte eine Bank, die Protokoll Nr. 4 benutzt, dazu bringen, blind eine 100-€-Münze zu signieren, seinem Konto aber nur eine Mark zu belasten. Dazu erzeugt er 99 Münzen vom Wert 1 € und eine 100-€-Münze. Wie groß ist die Wahrscheinlichkeit dafür, dass die Bank blind die 100-€-Münze signiert?

b) Wie kann die Bank verhindern, dass der Kunde einen Betrugsversuch unternimmt?

Aufgabe 1.2 Wie viele Bit muss die zufällig generierte Seriennummer einer E-Münze lang sein, damit die Wahrscheinlichkeit für eine zufällige Übereinstimmung von zwei Nummern kleiner ist als die Wahrscheinlichkeit, bei zwei aufeinander folgenden Ziehungen im Lotto (6 aus 49) sechs Richtige zu tippen? Tipp: Berechnen Sie zuerst die Wahrscheinlichkeit, mit einer zufällig erzeugten Seriennummer eine vorgegebene Zahl fester Länge zu treffen. Bestimmen Sie dann deren Länge n. In Abschnitt 6.1.2 wird gezeigt, dass die Seriennummer doppelt so lang (d. h. $2n$) sein muss, um eine gleich geringe Wahrscheinlichkeit für eine zufällige Übereinstimmung von zwei beliebigen Nummern zu erreichen.