Analyse und Konstruktion blinder interaktiver Signaturen

Inaugural-Dissertation

zur Erlangung des Doktorgrades
an den Naturwissenschaftlichen Fachbereichen (Mathematik)

der Justus-Liebig-Universität Gießen

vorgelegt von
Christine Fremdt

Gießen, Mai 2005

Dekan: Prof. Dr. Volker Metag

1. Berichterstatter: Prof. Dr. Albrecht Beutelspacher (Gießen)
2. Berichterstatter: Prof. Dr. Jörg Schwenk (Bochum)

Datum der Disputation: 24.06.2005

Einleitung

Noch nie haben sich Technologien so schnell entwickelt wie in den letzten Jahren. Besonders die Kommunikations- und die Informationstechnik haben einen enormen Aufschwung erlebt, und schon heute profitiert ein Großteil der Bevölkerung von Technologien wie dem Internet. Dabei wird das Internet vorrangig zur Informationsbeschaffung, etwa über Suchmaschinen, zum Online-Shopping und zum Homebanking verwendet. So gaben in einer Studie der Universität Karlsruhe ([K04]), in der über 13.000 Internet-Nutzer befragt wurden, mehr als die Hälfte der Befragten an, das Internet häufig zum Online-Shopping zu nutzen. Gut 40 Prozent gaben an, das Internet fast immer zum Online-Banking zu verwenden. Diese Zahlen machen einerseits deutlich, dass eine virtuelle Welt, gerade im Bereich des Handels, bereits Realität ist. Andererseits zeigen sie, dass diese Bereiche trotz des großen Zulaufs der letzten Jahre noch ein erhebliches Potential bergen. Damit ist gerade dieses Gebiet der Informationstechnik von großem Interesse und wissenschaftliche Forschung kann hier zukunftsweisend sein.

Besonders im Kontext des Online-Bankings oder des Online-Shoppings ist es für die Anbieter solcher Dienstleistungen unabdingbar, für einen hohen Grad an Sicherheit zu sorgen. Das heißt, es müssen Methoden entwickelt und umgesetzt werden, die sowohl die Sicherheitsinteressen der Anbieter als auch die der einzelnen Nutzer schützen. Dies ist ohne den Einsatz von Kryptographie nicht möglich. Die Beiträge der Kryptographie gehen dabei deutlich über digitale Signaturen und Verschlüsselungsmethoden hinaus. So wurden zahlreiche Mechanismen entwickelt, die bereits auf spezielle Anforderungen zugeschnitten sind. Dazu zählen die digitalen Münzsysteme, die ein digitales Pendant zu Bargeld darstellen. Hier handelt es sich um einen Mechanismus von hoher Komplexität, in dem verschiedene kryptographische Bausteine zusammengefügt werden, um die Sicherheitsinteressen aller Parteien zu gewährleisten. Der wesentliche Unterschied zu herkömmlichen Bezahlsystemen, wie zum Beispiel der Kreditkartenbezahlung, ist die Anonymität des Nutzers gegenüber seinem Kreditinstitut: Während Kredikartenzahlungen die Erstellung eines Kundenprofils ermöglichen, kann ein Kreditinstitut bei digitalen Münzsystemen nicht mehr

feststellen, welcher Kunde wo und für welchen Betrag eingekauft hat.

Ein kryptographischer Baustein hat sich in mehreren Vorschlägen (z.B. in [Ch82], [Br93], [Fe93]) für digitale Münzsysteme als besonders wichtig herausgestellt: die blinde digitale Signatur. Dieser Mechanismus stellt in einem Münzsystem sowohl die Echtheit und die Unfälschbarkeit einer Münze als auch die Anonymität des Kunden sicher. Aufbauend auf diesem Ansatz lässt sich ein generisches Konstruktionsprinzip für digitale Münzsysteme angeben, das unabhängig von den zugrundeliegenden Algorithmen ist (s. [N99]). Das Vorliegen eines solchen Konstruktionsprinzips bietet erhebliche Vorteile für die Sicherheitsanalyse des Gesamtsystems, insbesondere wenn man die Kommunikationsumgebung mit berücksichtigt. Jüngere Vorschläge für digitale Münzsysteme, wie zum Beispiel [STY00] oder [CHL05], kommen jedoch ohne den Baustein der blinden Signatur aus, sodass man auf den ersten Blick die Existenz von zwei wesentlich verschiedenen Klassen von digitalen Münzsystemen vermuten könnte. Die Frage, ob in diesen beiden Klassen tatsächlich wesentlich verschiedene Sicherheitsmechanismen verwendet werden, liegt nahe. Die vorliegende Arbeit unternimmt einen ersten Schritt zur Beantwortung dieser Frage.

Dazu ist es notwendig, sich die Anforderungen an blinde Signaturen klar zu machen. Blinde Signaturen sind, vereinfacht gesagt, digitale Unterschriften auf ein Dokument, die folgende Zusatzeigenschaft besitzen: Ein Signierer, der seine digitale Unterschrift auf ein Dokument setzt und dieses einem Empfänger übergibt, kann nach dem Signaturprozess weder das Dokument selbst noch das unterschriebene Dokument wiedererkennen. Die Idee zu blinden Signaturen stammt von David Chaum ([Ch82]), der als Veranschaulichung folgendes Szenario vorschlug: Ein Signierer unterzeichnet nicht direkt auf dem Dokument, sondern auf einem Briefumschlag, der das Dokument enthält. Zwischen dem Umschlag und dem Dokument befindet sich ein Kohlepapier, sodass der Empfänger der Signatur tatsächlich eine Unterschrift auf sein Dokument erhält. Der Signierer hat das unterschriebene Dokument während des Signaturprozesses jedoch nie zu Gesicht bekommen und wird es später nicht wiedererkennen können.

Aufbauend auf der Idee von Chaum wurden in den folgenden Jahren verschiedene blinde Signaturschemata entwickelt (z.B. [Ch82],[Oka92],[Fe93],[JLO97]). In [PS00] und [JLO97] wurde eine fundierte Untersuchung der benötigten Sicherheitseigenschaften sowie die Formalisierung eines geeigneten Sicherheitsbegriffs durchgeführt. Ferner gab es verschiedene Vorschläge, blinde Signaturen mit Zusatzeigenschaften auszustatten, wie zum Beispiel die partiell blinden Signaturen (z.B. [A96],[AO00]).

Alle diese Vorschläge haben gemeinsam, dass die Blendung im Signaturprotokoll erreicht, der Verifikationsprozess dabei jedoch nicht berücksichtigt wird. Allerdings lässt man sich typischerweise eine Nachricht signieren, um sowohl die Nachricht als auch die Signatur später gegenüber einer dritten Partei, die wir den Verifizierer nennen, zu verwenden. In der Verifikationsphase wird eine Interaktion des Signierers und des Empfängers der Signatur nachgewiesen, in deren Verlauf der Empfänger eine gültige Signatur des Signierers auf eine Nachricht erhalten hat. Dazu ist es jedoch nicht unbedingt notwendig, dass der Verifizierer die signierte Nachricht erhält. Vielmehr genügt es, wenn der Verifizierer nach dem Verifikationsprozess überzeugt ist, dass der Empfänger eine Signatur besitzt. Dieses Vorgehen wurde im Kontext von blinden Signaturen bisher noch nicht untersucht.

Der erste Teil der vorliegenden Arbeit beleuchtet den oben beschriebenen Ansatz. Dabei erfordert die Untersuchung dieses Verfahrens insbesondere eine genaue Analyse der Blindheitseigenschaft. Das heißt, es ist von besonderem Interesse, wie Blindheit grundsätzlich erreicht werden kann, und somit stellt sich die Frage nach (von konkreten Algorithmen unabhängigen) Bedingungen an den Signatur- bzw. den Verifikationsprozess.

Im zweiten Teil der Arbeit wird untersucht, inwiefern sich die gewonnenen Ergebnisse zur Konstruktion und Analyse blinder Signaturen einsetzen lassen:
Die oben beschriebenen Bedingungen liefern einen Rahmen für die Konstruktion von blinden Signaturen, unabhängig davon, an welcher Stelle die Blindheit umgesetzt wird und welche Algorithmen verwendet werden. Dies ist im Hinblick auf bereits bekannte Signaturen aus zwei Gründen interessant: Zum einen wird die Blindheit in vielen Protokollen als Eigenschaft mit extrem hohem Sicherheitsniveau umgesetzt. In diesem Zusammenhang spricht man von perfekt blinden Signaturen. Die Blindheit solcher perfekt blinden Signaturen wird im Allgemeinen durch den Einsatz von echten Zufallszahlen erreicht. In der Praxis ist die Erzeugung echter Zufallszahlen jedoch nicht praktikabel, sodass sich die Frage stellt, welchen Blindheitsgrad eine gegebene blinde Signatur in der Praxis erreicht. Hier können abstrakte Bedingungen an die Blindheit (eines verallgemeinerten Schemas) einen wichtigen Beitrag leisten. Zum anderen gibt es blinde Signaturen, die in zahlreichen Varianten vertreten sind. So stellt sich die Frage, inwiefern sich diese Varianten wesentlich unterscheiden. Hier führen abstrakte Bedingungen an die Blindheit zu einem übergeordneten Schema.

Ferner stellt sich die Frage, inwiefern man die neu eingeführte Möglichkeit der Interaktion während des Verifikationsprozesses ausnutzen kann, um eine blinde Signatur zu erhalten. Abstrakte Bedingungen an die Signatur stecken hier den Rahmen für

einen solchen Vorschlag ab und ermöglichen letztendlich eine Realisierung einer blinden Signatur nach dem angestrebten Vorgehen.

Überblick und Ergebnisse

Kapitel 1 stellt die benötigten Grundlagen zur Verfügung. Wie oben beschrieben, gliedert sich die weitere Arbeit in zwei Teile:

Im ersten Teil wird zunächst in Kapitel 2 ein kurzer Überblick über die bestehenden Begrifflichkeiten in Hinblick auf blinde Signaturen gegeben. Ferner werden ausgewählte Beispiele vorgestellt. In Kapitel 3 wird der neue Begriff der interaktiven Signatur eingeführt, die aus einem interaktiven Signaturprotokoll sowie einem interaktiven Verifikationsprotokoll besteht. Dabei stellen interaktive Signaturen eine Verallgemeinerung der klassischen digitalen Signaturen dar: Durch die Interaktivität werden die in einer digitalen Signatur implizit enthaltenen Kommunikationsschritte der beteiligten Parteien explizit beschrieben. Genauso fallen die bisher bekannten blinden Signaturen in die Klasse der interaktiven Signaturen: Dort wurde die Interaktion zwischen Signierer und Empfänger bereits ausgenutzt, um die Blindheitseigenschaft umzusetzen. Damit sind interaktive Signaturen ein geeignetes Konzept zur Untersuchung der oben beschriebenen Fragestellungen. In Kapitel 4 und 5 werden auf einem abstrakten Niveau hinreichende Bedingungen an die Blindheit einer interaktiven Signatur formuliert und diskutiert.

Im zweiten Teil der Arbeit werden, exemplarisch für Signaturen mit bereits bekannten blinden Varianten, in Kapitel 6 die RSA- und die Schnorr-Signatur unter den aufgeführten Fragestellungen beleuchtet. Es wird unter der Verwendung von sogenannten Blendungsfunktionen eine allgemeine Konstruktion blinder Signaturen angegeben und diskutiert. Hier erweisen sich die gewonnenen Ergebnisse als hilfreich: So konnten einerseits neue Varianten der blinden RSA-Signatur gefunden werden, die verschiedene Blindheitsgrade besitzen. Insbesondere lassen sich diese Ergebnisse für eine im oben beschriebenen Sinne praxisnahe Behandlung der blinden RSA-Sigantur verwenden. Andererseits konnte erstmals ein übergeordnetes Signaturschema für die Schnorr-Signatur angegeben werden, das verschiedene, bereits bekannte Varianten erfasst, aber ebenfalls neue Varianten ermöglicht. In Kapitel 7 wird schließlich die Blendung während des Verifikationsprozesses untersucht. Dazu wird zunächst eine generische Konstruktion einer blinden interaktiven Signatur angegeben und diskutiert. Mit Hilfe dieser Ergebnisse kann ein blindes Signaturschema des betrachteten Typs angegeben werden, sodass die Existenz solcher Signaturen gezeigt ist. Damit

konnte insbesondere nachgewiesen werden, dass die Betrachtungen dieser Arbeit zu einer neuen Klasse von blinden Signaturen führen.

Die in dieser Arbeit untersuchten Fragestellungen sind insbesondere durch das von der Deutschen Forschungsgemeinschaft geförderten Schwerpunktprogramm „Sicherheit in der Informations- und Kommunikationstechnik“ motiviert, das in den Jahren 1999 bis 2004 am Lehrstuhl von Herrn Prof. Dr. Albrecht Beutelspacher durch das Projekt *Digitale Geldbörsen* vertreten wurde. Einige Teile der Arbeit wurden im Rahmen dieses Projektes angefertigt.

An dieser Stelle möchte ich Herrn Prof. Dr. Albrecht Beutelspacher für die Gelegenheit, diese Arbeit zu verfassen, und für die hervorragende Betreuung danken. Genauso gilt mein Dank Herrn Prof. Dr. Jörg Schwenk für die Bereitschaft, die Arbeit zu begutachten. Weiterhin bedanke ich mich bei Frau Dr. Heike Neumann für die wertvollen Gespräche und das Korrekturlesen der Arbeit sowie bei meinen Kollegen Jörn Schweisgut, Thomas Schwarzpaul, Dr. Matthias Baumgart und besonders bei Björn Fay für die fruchtbaren Diskussionen. Ferner danke ich meiner Schwester Heike Fremdt, Mirko Kowarsch und Dr. Christoph Neuhoff für ihre hilfreichen Anregungen. Vor allem aber möchte ich an dieser Stelle Mirko Kowarsch und meiner Familie für all ihre Unterstützung während der letzten Jahre danken, ohne die diese Arbeit nicht möglich gewesen wäre.

Inhaltsverzeichnis

Kapitel 1

Grundlagen

1.1 Kryptographische Grundlagen

Dieser Arbeit liegt das in der Kryptographie übliche Rechnermodell der Turing-Maschine zugrunde, wobei insbesondere Algorithmen als Turing-Maschine beschrieben werden. Eine (deterministische) Turing-Maschine besteht aus je einem Ein- und einem Ausgabeband, einem Schreibkopf und einer Zustandskontrolle. Eine probabilistische Turing-Maschine besitzt zusätzlich ein Zufallsband, wobei das Zufallsband eine Reihe von unendlich vielen Bits enthält, die man sich als das Ergebnis von unendlich vielen Münzwürfen vorstellen kann. Lässt sich die Laufzeit einer Turing-Maschine durch ein Polynom in der Länge der Eingabe nach oben abschätzen, so sprechen wir von einer polynomiellen Turingmaschine. Probabilistische polynomielle Turingmaschinen heißen auch effiziente Turing-Maschinen. Ist im Folgenden die Rede von einem effizienten Algorithmus, so ist darunter zu verstehen, dass die zugrundeliegende Turing-Maschine effizient ist.

In dieser Arbeit werden meist zwei Turing-Maschinen betrachtet, die miteinander kommunizieren. In diesem Zusammenhang spricht man auch von interaktiven Turing-Maschinen. Dabei ist eine interaktive Turing-Maschine eine probabilistische Turing-Maschine, die zusätzlich zwei Kommunikationsbänder und ein Zustandsband sowie ein Extra-Eingabeband besitzt. Eins der Kommunikationsbänder ist für eingehende, das andere Kommunikationsband für ausgehende Nachrichten zuständig. Das Zustandsband besteht aus einer einzigen Zelle, deren Inhalt angibt, ob die Maschine aktiv oder inaktiv ist. Eine Menge Π von Vorschriften, gemäß der zwei interaktive Turing-Maschinen $\mathcal{A}$ und $\mathcal{B}$ miteinander interagieren, heißt interaktives Protokoll. Dabei haben beide Turing-Maschinen sowohl das Eingabeband als auch die Kommunikationsbänder gemeinsam. Weiterhin sind $\mathcal{A}$ und $\mathcal{B}$ abwechselnd aktiv.

Die Protokollansicht von $\mathcal{A}$, d.h. alle Inhalte beider Kommunikationsbänder und des Zufallsbandes von $\mathcal{A}$ sowie die Inhalte des Eingabebandes bezeichnen wir mit $view_{\mathcal{A}}(\Pi)$. Dabei fassen wir $view_{\mathcal{A}}(\Pi)$ als Tupel von diesen Werten auf, wobei die Reihenfolge der Einträge durch die Reihenfolge der Kommunikationsschritte festgelegt ist.

In der Kryptographie unterscheidet man zwischen symmetrischen und asymmetrischen (oder Public-Key-) Verfahren. Letztere beruhen auf der Vermutung, dass es sogenannte schwere Probleme gibt, die sich im Durchschnittsfall durch den Einsatz von effizienten Turing-Maschinen nicht lösen oder approximieren lassen. Dementsprechend werden verschiedene Klassen von Problemen unterschieden. Die beiden wichtigsten Klassen sind:

P: Die Klasse aller Probleme P, für die eine deterministische polynomielle Turing-Maschine existiert, die P löst.

NP: Die Klasse aller Probleme P, für die eine deterministische polynomielle Turingmaschine existiert, die eine beliebige Lösung von P verifiziert.

Eine Grundannahme, auf der die Public-Key-Kryptographie beruht, ist, dass $\mathbf{P} \neq \mathbf{NP}$ gilt. Diese Aussage konnte bisher noch nicht bewiesen werden, sodass Public-Key-Verfahren im Allgemeinen auf einer komplexitätstheoretischen Annahme beruhen.

Um die wichtigsten kryptographischen Annahmen formulieren zu können, benötigt man den folgenden Begriff:

Definition 1.1 *Es sei $\nu : \mathbb{N} \to \mathbb{R}$ eine Funktion. Dann heißt ν* **vernachlässigbar**, *falls es für alle $c \in \mathbb{N}$ ein $k_c \in \mathbb{N}$ gibt, sodass für alle $k \geq k_c$ gilt:*

$$|\nu(k)| \leq k^{-c}.$$

Ferner verwenden wir die Notation $x \in_R X$, um die zufällige Wahl eines Elementes x gemäß einer Gleichverteilung aus der Menge X zu beschreiben (vgl. Abschnitt 1.2).

Die grundlegenden kryptographischen Annahmen, die in dieser Arbeit benötigt werden, sind:

Annahme 1.1 (RSA-Annahme) *Es sei n eine aus zwei verschiedenen Primzahlen p und q zusammengesetzte Zahl, wobei p und q jeweils eine Länge von k Bit haben*[1].

[1] Die Zahl n aus der RSA-Annahme nennen wir **RSA-Modul**.

Weiterhin sei e eine zu $\phi(n)$[2] teilerfremde Zahl. Die RSA-Annahme besagt, dass es für ein gemäß den genannten Bedingungen zufällig gewähltes Paar (n, e) und eine zufällig gewählte Zahl $y \in_R \mathbb{Z}_n^*$ keinen probabilistischen polynomiellen Algorithmus $\mathcal{A}$ gibt, der mit in k nicht vernachlässigbarer Wahrscheinlichkeit bei Eingabe von $1^k, n, e$ und y eine Zahl x mit $y = x^e \bmod n$ berechnen kann.

Annahme 1.2 (Starke RSA-Annahme) *Für einen zufällig gewählten RSA-Modul n und eine zufällig gewählte Zahl $y \in_R \mathbb{Z}_n^*$ gibt es keinen effizienten Algorithmus $\mathcal{A}$, der bei Eingabe von $1^k, n$ und y mit in k nicht vernachlässigbarer Wahrscheinlichkeit zwei Zahlen x und e mit $x^e = y \bmod n$ berechnen kann.*

Annahme 1.3 (Diskreter-Logarithmus-Annahme) *Für eine zufällig gwählte Primzahl p der Länge k Bit, für eine zufällig gewähltes erzeugendes Element g von $\mathbb{Z}_p^*$ und ein zufällig gewähltes Element $h \in_R \mathbb{Z}_p^*$ gibt es keinen effizienten Algorithmus $\mathcal{A}$, der mit in k nicht vernachlässigbarer Wahrscheinlichkeit bei Eingabe von k, p, g und h eine Zahl $x \in \mathbb{N}$ mit $h = g^x \bmod p$ berechnen kann.*

1.1.1 Einweg-Hash-Funktionen

Einer der grundlegenden Bausteine für kryptographische Primitive sind die Einweg-Hash-Funktionen, d.h. Algorithmen, die sich effizient berechnen lassen, aber schwer zu invertieren sind. Diese finden Anwendungen z.B. in digitalen Signaturen oder in nicht-interaktiven Zero-Knowledge-Beweisen.

Definition 1.2 *Sei $n \in \mathbb{N}$. Eine Abbildung $\mathcal{H} : \{0,1\}^* \longrightarrow \{0,1\}^n$ heißt* **Hash-Funktion**, *wenn es einen polynomiellen Algorithmus gibt, der für jede Nachricht $m \in \{0,1\}^*$ den Wert $\mathcal{H}(m)$ berechnet.*

In der Kryptographie fordert man im Allgemeinen von Hash-Funktionen die starke Kollisionsresistenz:

Definition 1.3 *Eine* **kryptographische oder kollisionsresistente Hash-Funktion** *ist eine Hash-Funktion $\mathcal{H}$, sodass gilt:*
Es gibt keinen effizienten Algorithmus $\mathcal{A}$, der zwei Werte $m \neq m'$ mit $\mathcal{H}(m) = \mathcal{H}(m')$ findet.

Eine weitere wichtige Eigenschaft für die Sicherheit kryptographischer Bausteine ist die Einweg-Eigenschaft.

[2] ϕ ist die Eulersche Phi-Funktion: $\phi : \mathbb{N} \to \mathbb{N}$, mit $\phi(n) = |\{1 \leq a < n | \gcd(a, n) = 1\}|$.

Definition 1.4 *Eine* **(kryptographische) Einweg-Hash-Funktion** *ist eine (kryptographische) Hash-Funktion $\mathcal{H}$, die folgende Eigenschaft besitzt:*
Einweg-Eigenschaft: *Es gibt keinen effizienten Algorithmus, der bei Eingabe eines zufällig gewählten $y \in_R \{0,1\}^n$ ein m mit $\mathcal{H}(m) = y$ ausgibt.*

Ist in dieser Arbeit von Hash-Funktionen die Rede, so sind stets kryptographische Einweg-Hash-Funktionen gemeint.

1.1.2 Verschlüsselungssysteme

Eine der wichtigsten Aufgaben der Kryptographie ist die Bereitstellung von Methoden zum vertraulichen Austausch von Nachrichten. Da in dieser Arbeit lediglich symmetrische Verfahren eine Rolle spielen, betrachten wir das folgende Szenario: Zwei Parteien möchten Nachrichten austauschen. Sie sind beide im Besitz des gleichen Geheimnisses κ, das sie verwenden, um die Vertraulichkeit ihrer Kommunikation herzustellen. Ein Verfahren, das diese Vertraulichkeit herstellt, heißt Kryptosystem oder Verschlüsselungsystem.

Definition 1.5 *Es sei $\mathcal{M}$ die Menge aller* **Klartexte**, *$\mathcal{C}$ die Menge aller* **Geheimtexte** *und $\mathcal{K}$ die Menge aller* **Schlüssel**. *Ferner sei*

$$\mathcal{F} := \{f_\kappa : \mathcal{M} \to \mathcal{C} | \kappa \in \mathcal{K}\}$$

eine Menge von injektiven Abbildungen. Dann heißt $\mathcal{F}$ **Kryptosystem**.

Im Allgemeinen ist zu einem Kryptosystem ein Sicherheitsparameter k gegeben, der die Länge der verwendeten Schlüssel vorgibt.

Es gibt verschiedene Sicherheitsbegriffe für Kryptosysteme, die sich je nach Fähigkeiten und Erfolgen des Angriffs unterscheiden. Wir werden die im Folgenden vorgestellten Sicherheitsbegriffe benötigen:

Definition 1.6 *Ein Kryptosystem $\mathcal{F}$ heißt* **perfekt sicher**, *wenn die a priori Wahrscheinlichkeit eines beliebigen Klartextes gleich seiner a posteriori Wahrscheinlichkeit ist.*

Die perfekte Sicherheit ist der stärkste Sicherheitsbegriff für Verschlüsselungssysteme, der in der Praxis oft nicht zu realisieren ist. Man kann diesen Begriff abschwächen, ohne die praktische Anwendbarkeit der Kryptosysteme zu gefährden, indem man in Betracht zieht, dass real vorkommende Angreifer rechnerisch nicht unbeschränkt sind. In der Literatur (vgl. zum Beispiel [Go04]) sind zwei äquivalente Sicherheitsbegriffe bekannt: Die **semantische Sicherheit** und die **Sicherheit im**

Sinne der polynomiellen Ununterscheidbarkeit. Wir werden in dieser Arbeit letzteren Begriff verwenden. In der Darstellung dieses Begriffs richten wir uns nach [BNS05].

Die Definition der polynomiellen Ununterscheidbarkeit beruht wesentlich auf dem folgenden Spiel, das der Angreifer zusammen mit einem Orakel $\mathcal{O}$ spielt. Dabei verstehen wir unter einem Orakel einen probabilistischen polynomiellen Algorithmus, der sich stets an die ihm vorgegebenen Schritte hält.

Spiel 1.1

1. *Es wird ein Schlüssel $\kappa \in_R \mathcal{K}$ und ein Bit $b \in_R \{0,1\}$ gewählt. Beide werden vor $\mathcal{A}$ geheimgehalten. Das Orakel $\mathcal{O}$ erhält b und κ.*
2. *Der Angreifer $\mathcal{A}$ wählt zwei Klartexte m_0 und m_1 und sendet diese an $\mathcal{O}$.*
3. *$\mathcal{O}$ berechnet $c := f_\kappa(m_b)$.*
4. *$\mathcal{A}$ erhält c und gibt nun ein Bit b' aus. Er gewinnt, falls $b' = b$ ist.*

Offensichtlich müssen für die Sicherheitsanalyse eines Kryptosystems die Fähigkeiten eines Angreifers in Betracht gezogen werden. Mit der im Folgenden vorgestellten Definition 1.7 werden aktive Angriffe beschrieben. Die Unterscheidung in aktive und passive Angriffe beruht auf der Vorstellung, dass es zwei Angreifertypen gibt. Der passive Angreifer hört eine Kommunikation ab und erhält so Informationen über die verwendete Verschlüsselungsfunktion. Der aktive Angreifer ist hingegen in der Lage, Informationen zu von ihm gewählten Klartexten zu erhalten, zum Beispiel, indem er sich Klartexte verschlüsseln oder schon verschlüsselte Klartexte wieder entschlüsseln lässt. Ein aktiver Angriff wird mit Hilfe eines zweiten Orakels formalisiert. Dieses Orakel ist, genau wie $\mathcal{O}$ in Spiel 1.7 im Besitz des Schlüssels κ. Aus diesem Grund bezeichnen wir dieses Orakel mit $\mathcal{O}_\kappa$. Der Sinn von $\mathcal{O}_\kappa$ ist, je nach Angriffsart, Geheimtexte zu von dem Angreifer gewählten Klartexten oder auch Klartexte zu von dem Angreifer gewählten Geheimtexten unter dem Schlüssel κ zu erzeugen. Im ersten Fall sprechen wir von einem **Angriff mit gewählten Klartexten**. Falls $\mathcal{O}_\kappa$ sogar von dem Angreifer gewählte Geheimtexte wieder entschlüsselt, sprechen wir von einem **Angriff mit gewählten Geheimtexten**. Da wir in dieser Arbeit im Allgemeinen die Situation eines Angriffs mit gewählten Klartexten betrachten, werden wir hier Orakel betrachten, die ausschließlich Geheimtexte liefern. Diese nennen wir **Verschlüsselungsorakel**.

Definition 1.7 *Sei $k \in \mathbb{N}$ ein Sicherheitsparameter. Ein Kryptosystem $\mathcal{F}$ heißt* **rechnerisch sicher** *oder* **sicher im Sinne der polynomiellen Ununterscheidbarkeit (von Geheimtexten)**, *falls es für jeden effizienten Algorithmus $\mathcal{A}$ mit Zugriff auf ein Verschlüsselungsorakel $\mathcal{O}_\kappa$ eine in k vernachlässigbare Funktion ν gibt, sodass für die Erfolgswahrscheinlichkeit $\varepsilon(k)$ von $\mathcal{A}$ in Spiel 1.1 gilt:*

$$1/2 - \varepsilon(k) = \nu(k).$$

Das Orakel $\mathcal{O}$ kann in Spiel 1.1 verschiedene Eigenschaften haben: Man kann fordern, dass nur eine Anfrage an das Orakel $\mathcal{O}$ möglich ist. Eine andere Möglichkeit ist, polynomiell viele Anfragen an das Orakel zu erlauben, wobei sogar adaptive Anfragen zugelassen sind (vgl. [GB01]). Für eine ausführliche Diskussion der beiden Möglichkeiten sei auf Kapitel 5 in [Go04] verwiesen. In dieser Arbeit wird die letztgenannte Möglichkeit verwendet.

Im weiteren Verlauf der Arbeit wird ein etwas anderer Sicherheitsbegriff benötigt, der im Folgenden vorgestellt wird. Hier werden im Prinzip zwei Schlüssel gleichzeitig angegriffen. Dazu betrachten wir folgendes

Spiel 1.2

1. *Es werden zwei Schlüssel $\kappa_0, \kappa_1 \in_R \mathcal{K}$ und ein Bit $b \in_R \{0,1\}$ gewählt, welche vor $\mathcal{A}$ geheimgehalten werden. Das Orakel $\mathcal{O}$ erhält b, sowie κ_0 und κ_1.*
2. *Der Angreifer $\mathcal{A}$ wählt zwei Klartexte m_0 und m_1 und sendet diese an $\mathcal{O}$.*
3. *$\mathcal{O}$ berechnet $c_b = f_{\kappa_b}(m_b)$ und $c_{1-b} = f_{\kappa_{1-b}}(m_{1-b})$.*
4. *$\mathcal{A}$ erhält c_b und c_{1-b} und gibt nun ein Bit b' aus. Er gewinnt, falls $b' = b$ ist.*

Wie oben ist es dem Angreifer im beschriebenen Spiel erlaubt, Anfragen an zwei weitere Orakel $\mathcal{O}_0$ (bezüglich des Schlüssels κ_0) und $\mathcal{O}_1$ (bezüglich des Schlüssels κ_1) zu stellen.

Definition 1.8 *Sei $k \in \mathbb{N}$ ein Sicherheitsparameter. Ein Kryptosystem $\mathcal{F}$ heißt* **rechnerisch sicher unter einem Angriff auf zwei Schlüssel**, *falls es für jeden effizienten Algorithmus $\mathcal{A}$ eine in k vernachlässigbare Funktion ν gibt, sodass für die Erfolgswahrscheinlichkeit $\varepsilon(k)$ von $\mathcal{A}$ mit Zugriff auf die Verschlüsselungsorakel $\mathcal{O}_0$ und $\mathcal{O}_1$ in Spiel 1.2 gilt:*

$$1/2 - \varepsilon(k) = \nu(k).$$

Es wurde oben bereits angedeutet, dass es verschiedene Möglichkeiten gibt, das Orakel $\mathcal{O}$ zu handhaben. Lässt man nur eine einzige Anfrage an $\mathcal{O}$ zu, so ist der durch Definition 1.8 gegebene Sicherheitsbegriff im Fall $\kappa_0 = \kappa_1$ stärker als die polynomielle Ununterscheidbarkeit. Im Fall, dass mehrere Anfragen an $\mathcal{O}$ zugelassen sind, sind beide Sicherheitbegriffe äquivalent:

Satz 1.1 *Ein Kryptosystem $\mathcal{F}$ ist genau dann rechnerisch sicher im Sinne der polynomiellen Ununterscheidbarkeit unter einem Angriff mit gewählten Klartexten, falls die Erfolgswahrscheinlichkeit eines effizienten Angreifers $\mathcal{A}$ mit Zugriff auf ein Verschlüsselungsorakel $\mathcal{O}_\kappa$ in folgendem Spiel höchstens vernachlässigbar von $1/2$ abweicht.*

1. *Es wird ein Schlüssel $\kappa \in_R \mathcal{K}$ und ein Bit $b \in_R \{0,1\}$ gewählt. Beide werden vor $\mathcal{A}$ geheimgehalten.*
2. *Das Orakel $\mathcal{O}$ erhält b und κ.*
3. *Der Angreifer $\mathcal{A}$ wählt zwei Klartexte m_0 und m_1 und sendet diese an $\mathcal{O}$.*
4. *$\mathcal{O}$ berechnet $c_b := f_\kappa(m_b)$ und $c_{1-b} := f_\kappa(m_{1-b})$.*
5. *$\mathcal{A}$ erhält (c_b, c_{1-b}). Er gibt nun ein Bit b' aus. $\mathcal{A}$ gewinnt, falls $b' = b$ ist.*

Beweis. Wir nehmen an, dass $\mathcal{F}$ nicht rechnerisch sicher im Sinne der polynomiellen Ununterscheidbarkeit ist. Dann gibt es einen effizienten Angreifer $\mathcal{A}$, der das in Definition 1.7 beschriebene Spiel mit einer Wahrscheinlichkeit $\varepsilon(k)$ gewinnt, wobei $1/2 - \varepsilon(k)$ nicht vernachlässigbar im Sicherheitsparameter k ist. Wir konstruieren einen effizienten Angreifer $\mathcal{A}^*$, der das Spiel aus Satz 1.1 mit der gleichen Wahrscheinlichkeit gewinnt.

1. Es wird ein Schlüssel $\kappa \in_R \mathcal{K}$ und ein Bit $b \in_R \{0,1\}$ gewählt. Beide werden vor $\mathcal{A}^*$ geheimgehalten.
2. Das Orakel $\mathcal{O}$ erhält b und κ.
3. Der Angreifer $\mathcal{A}^*$ startet $\mathcal{A}$.
4. $\mathcal{A}$ wählt zwei Klartexte m_0 und m_1 und sendet diese an $\mathcal{A}^*$, der sie an $\mathcal{O}$ weitergibt.
5. $\mathcal{O}$ berechnet $c_b := f_\kappa(m_b)$ und $c_{1-b} := f_\kappa(m_{1-b})$.
6. $\mathcal{A}^*$ erhält c_b und c_{1-b} und gibt einen der beiden Werte c_b an $\mathcal{A}$ weiter.

7. $\mathcal{A}$ gibt nun ein Bit b' aus. Die Ausgabe von $\mathcal{A}^*$ ist ebenfalls b'.

Offensichtlich gewinnt $\mathcal{A}^*$ das Spiel genau dann, wenn $\mathcal{A}$ das richtige Bit ausgibt. Damit ist seine Erfolgswahrscheinlichkeit ebenfalls $\varepsilon(k)$. Ferner ist $\mathcal{A}^*$ effizient, da $\mathcal{A}$ effizient ist.

Sei nun $\mathcal{F}$ rechnerisch sicher im Sinne der polynomiellen Ununterscheidbarkeit gegen einen Angriff mit gewählten Nachrichten. Angenommen, es gibt einen effizienten Angreifer $\mathcal{A}$, der das Spiel aus Satz 1.1 mit einer Wahrscheinlichkeit von $\varepsilon(k)$ gewinnt, wobei $1/2-\varepsilon(k)$ nicht vernachlässigbar ist. Wir konstruieren einen effizienten Angreifer $\mathcal{A}^*$, der das Spiel aus Definition 1.7 ebenfalls mit Wahrscheinlichkeit $\varepsilon(k)$ gewinnt.

1. Es wird ein Schlüssel $\kappa \in_R \mathcal{K}$ und ein Bit $b \in_R \{0,1\}$ gewählt. Beide werden vor $\mathcal{A}$ geheimgehalten.

2. Das Orakel $\mathcal{O}$ erhält b und κ.

3. Der Angeifer $\mathcal{A}^*$ startet $\mathcal{A}$. $\mathcal{A}$ erzeugt zwei Klartexte m_0 und m_1 und gibt diese an $\mathcal{A}^*$.

4. $\mathcal{A}^*$ sendet (m_0, m_0) an $\mathcal{O}$ und erhält einen Geheimtext $c_b := f_\kappa(m_0)$ zu der Nachricht m_0. $\mathcal{A}^*$ sendet nun (m_1, m_1) an $\mathcal{O}$ und erhält so einen Geheimtext $c_{1-b} := f_\kappa(m_1)$ zur Nachricht m_1.

5. Er gibt c_b und c_{1-b} an $\mathcal{A}$.

6. $\mathcal{A}^*$ gibt das Bit b' aus, das er von $\mathcal{A}$ erhält.

Die Erfolgswahrscheinlichkeit von $\mathcal{A}^*$ ist gerade $\varepsilon(k)$. Weiterhin muss $\mathcal{A}^*$ nur zwei Anfragen an das Orakel stellen und ist damit immer noch effizient, wenn $\mathcal{A}$ effizient ist. □

1.1.3 Digitale Signaturen

Neben der Geheimhaltung von Nachrichten ist die Sicherung der Nachrichtenauthentizität das zweite große Aufgabenfeld der Kryptographie. Diese wird im Falle der Public-Key-Kryptographie im Allgemeinen durch digitale Signaturen sichergestellt. Die nächste Definition erklärt, was wir unter einer digitalen Signatur formal verstehen. Dabei sei $\mathcal{M}$ die Menge der Nachrichten, die sich mit der digitalen Signatur unterzeichnen lassen.

Definition 1.9 *Es sei $k \in \mathbb{N}$ und $\mathcal{M}$ die Menge aller Nachrichten. Ferner sei $m \in \mathcal{M}$. Gegeben sind die folgenden (probabilistischen) polynomiellen Algorithmen:*

1. *Der* **Schlüsselerzeugungsalgorithmus** *G gibt bei Eingabe des Sicherheitsparameters 1^k einen* **öffentlichen Schlüssel** pk *und einen* **geheimen Schlüssel** sk *aus:* $(\mathsf{pk}, \mathsf{sk}) \in G(1^k)$.

2. *Der* **Signaturalgorithmus** *σ gibt bei Eingabe des Sicherheitsparameters k, des geheimen Schlüssels sk und des öffentlichen Schlüssels pk sowie einer Nachricht m einen String s aus.*

3. *Der* **Verifikationsalgorithmus** *V gibt bei Eingabe des öffentlichen Schlüssels pk, einer Nachricht m und eines Strings s die Werte $1(=\mathsf{true})$ oder $0(=\mathsf{false})$ aus.*

Das Tripel $\Sigma = (G, \sigma, V)$ heißt **digitale Signatur**, *wenn folgende Bedingung erfüllt ist: Für jedes Paar $(\mathsf{sk}, \mathsf{pk}) \in G(1^k)$ und für alle $m \in \mathcal{M}$ gilt*

$$\mathcal{P}\left(V(\mathsf{pk}, m, \sigma(\mathsf{sk}, m)) = 1\right) = 1,$$

wobei die Wahrscheinlichkeit sich auf die Inhalte der Zufallsbänder von σ und V bezieht.

Notation: Wir bezeichnen mit $\sigma(m)$ die Ausgabe s von σ und mit $S(m)$ die Menge aller Signaturen auf die Nachricht m bzgl. eines festgelegten geheimen Schlüssels sk. Es wird in der Arbeit immer klar sein, welcher Sicherheitsparameter und welcher Schlüssel verwendet wird, sodass wir diese Parameter in beiden Bezeichnungen nicht mehr aufführen.

Bemerkung 1.1 *Nur der Schlüsselerzeugungsalgorithmus muss als probabilistisch vorausgesetzt werden, für σ und V sind auch deterministische Algorithmen erlaubt. Man beachte, dass $S(m)$ nur ein Element enthält, falls der Signaturalgorithmus σ deterministisch ist.*

Bezüglich der Sicherheit unterscheidet man zunächst vier grundlegende Angriffstypen und schließlich vier unterschiedliche Erfolgsstufen für einen Angreifer. Diese sind im Folgenden dargestellt, wobei wir mit den Angriffstypen beginnen:

Angriff ohne bekannte Signaturen: Der Angreifer kennt nur den öffentlichen Schlüssel des Signierers.

Angriff mit bekannten Signaturen: Der Angreifer kennt den öffentlichen Schlüssel des Signierers, aber auch Paare von Signaturen und Nachrichten, die der Signierer erstellt hat. Auf die Wahl der Nachrichten-Signatur-Paare hat der Angreifer keinen Einfluss.

Angriff mit gewählten Nachrichten: Der Angreifer kennt den öffentlichen Schlüssel des Signierers und darf mehrere Nachrichten wählen, zu denen der Signierer korrekte Signaturen erzeugt.

Adaptiver Angriff mit gewählten Nachrichten: Der Angreifer darf sich eine bestimmte Anzahl von Signaturen vom Signierer erzeugen lassen. Dabei bestimmt er selbst die Nachrichten, die signiert werden, insbesondere darf eine Anfrage an den Signierer von den vorangegangenen abhängen.

Die Erfolgsstufen sind

Existentielle Fälschbarkeit: Der Angreifer kann die Signatur zu einer Nachricht fälschen, wobei er die Nachricht nicht selbst gewählt hat.

Selektive Fälschbarkeit: Der Angreifer kann Signaturen zu einer oder mehreren von ihm selbst gewählte Nachrichten fälschen.

Universelle Fälschbarkeit: Der Angreifer kann die Signatur zu einer beliebigen Nachricht erstellen, ist jedoch nicht im Besitz des geheimen Signaturschlüssels.

Kompromittierung des Schlüssels: Der Angreifer kann den geheimen Schlüssel des Signierers berechnen.

Von einem sicheren Signaturschema erwartet man, dass ein Angreifer, der möglichst viel Handlungsspielraum hat, die geringste Erfolgsstufe nur mit vernachlässigbarer Wahrscheinlichkeit erreichen kann. Dieser Sicherheitsbegriff ist in der folgenden Definition formal dargestellt.

Definition 1.10 *Sei Σ ein digitales Signaturschema. Dann heißt* Σ **sicher (gegen existentielle Fälschbarkeit unter einem adaptiven Angriff mit gewählten Nachrichten)**, *falls jeder effiziente Angreifer $\mathcal{A}$ in folgendem Spiel gegen einen ehrlichen Signierer $\mathcal{S}$ höchstens eine vernachlässigbare Erfolgswahrscheinlichkeit hat.*

1. *$\mathcal{S}$ erzeugt bei Eingabe des Sicherheitsparameters k ein Schlüsselpaar $(\mathsf{pk}, \mathsf{sk}) \in G(k)$.*

2. *Sei ℓ polynomiell in k. Der Angreifer $\mathcal{A}$ erhält pk. Er wählt einen Klartext m_1 und sendet diesen an $\mathcal{S}$.*

3. *$\mathcal{A}$ erhält $\sigma(m_1)$ von $\mathcal{S}$.*

4. *$\mathcal{A}$ wählt eine neue Nachricht m_2, wobei m_2 von m_1 und $\sigma(m_1)$ abhängen kann, und erhält von $\mathcal{S}$ die Signatur $\sigma(m_2)$. $\mathcal{A}$ stellt auf diese Art und Weise ℓ Anfragen an $\mathcal{S}$ und erhält so die Nachrichten-Signatur-Paare*

$$(m_1, \sigma(m_1)), \ldots (m_\ell, \sigma(m_\ell)).$$

5. *$\mathcal{A}$ gibt ein Nachrichten-Signatur-Paar $(m, \sigma(m))$ aus. Er gewinnt, falls $m \neq m_i$ für alle $1 \leq i \leq \ell$.*

Es gibt zahlreiche Beipiele für digitale Signaturschemata, die auf unterschiedlichen schweren zahlentheoretischen Problemen beruhen. Wir stellen im Folgenden diejenigen Signaturen vor, die für diese Arbeit wichtig sind.

1.1.3.1 Die RSA-Signatur

Schlüsselerzeugung. Es werden zwei Primzahlen p und q sowie ein e mit

$$\gcd(e, \phi(n)) = 1$$

für $n = p \cdot q$ gewählt. Weiterhin berechnet man ein d mit $d \cdot e \equiv 1 \bmod \varphi(n)$. Das Paar $\mathsf{pk} = (n, e)$ wird als öffentlicher Schlüssel, die Zahl $\mathsf{sk} = d$ als geheimer Schlüssel verwendet. Weiterhin werden die Primzahlen p, q und $\phi(n)$ geheimgehalten.

Signaturalgorithmus. Bei Eingabe von d, n und m wird die Signatur auf die Nachricht m ist als $s = m^d \bmod n$ berechnet. Die Ausgabe ist $\sigma(m) = s$.

Verifikation. Bei Eingabe von (n, e) und (m, s) ist die Ausgabe genau dann 1, wenn die Gleichung $m = s^e \bmod n$ erfüllt ist.

Die RSA-Signatur ist existentiell fälschbar unter einem Angriff ohne bekannte Signaturen (siehe z.B. [BNS05]). Signiert man statt der Nachricht m das Bild $\mathcal{H}(m)$ von m unter einer geeigneten Hash-Funktion $\mathcal{H} : \{0,1\}^* \longrightarrow \mathbb{Z}_n$, so lässt sich unter der RSA-Annahme die Sicherheit der RSA-Signatur im Random-Oracle Modell nachweisen (s. [Wei99]). Hier wird von der Hash-Funktion verlangt, dass sie eine Full-Domain-Hash-Funktion ist.

1.1.3.2 Die Schnorr-Signatur

Die im Folgenden vorgestellte Schnorr-Signatur wurde in [Schn89] bzw. [Schn91] vorgestellt. Sie ist in zweierlei Hinsicht bemerkenswert: Einerseits ist sie eine Variante der ElGamal-Signatur (s. [HMP94a]). Andererseits lässt sie sich aus einem

Zero-Knowledge-Identifikations-Protokoll konstruieren und fällt daher in die gleiche Klasse von Signaturen wie zum Beispiel die Fiat-Shamir-Signatur oder die Guillou-Quisquater-Signatur (s. zum Beispiel [Sch96]).

Schlüsselerzeugung. Es werden zwei Primzahlen p und q gewählt, wobei q ein Teiler von $p-1$ der Länge k ist. Ferner wird ein Element $g \in \mathbb{Z}_p^*$ der Ordnung q bestimmt. Der geheime Schlüssel x wird zufällig und gemäß einer Gleichverteilung aus $\mathbb{Z}_q$ gewählt. Der öffentliche Schlüssel besteht aus den Parametern p, q und g sowie $y := g^x \bmod p$. Der Schlüsselerzeugungsalgorithmus gibt das folgende Schlüsselpaar aus:

$$(\mathsf{pk}, \mathsf{sk}) = ((p, q, g, y), x)\,.$$

Signaturalgorithmus. Bei Eingabe von $(p, q, g, y), x$ und m wird eine Signatur auf die Nachricht m wie folgt erzeugt.

- Wähle $w \in_R \mathbb{Z}_q$ und berechne $r := g^w \bmod p$.
- Bestimme $c := \mathcal{H}(r, m) \in \mathbb{Z}_q$.
- Die Signatur auf m ist $\sigma(m) = (r, s)$, mit $s := cx + w \bmod q$.

Verifikationsalgorithmus. Bei Eingabe einer Signatur $\sigma(m) = (r, s)$ auf die Nachricht m ist die Ausgabe des Verifikationsalgorithmus' genau dann 1, wenn die folgende Gleichung gilt:

$$g^s \equiv y^c r \pmod{p}.$$

Die Schnorr-Signatur ist unter der Diskreten-Logarithmus-Annahme in Untergruppen von $\mathbb{Z}_p$ mit Primzahlordnung sicher im Random-Oracle-Modell (s. [PS00])

1.1.3.3 Die Camenisch-Lysyanskaya-Signatur

Im Jahre 2002 stellten Lysyanskaya und Camenisch in [CL02] eine Signatur vor, die für die Zwecke dieser Arbeit ausgezeichnet geeignet ist. Sie hat unter anderem den Vorteil, dass ihre Sicherheit nicht auf der Verwendung einer Hash-Funktion beruht. Im Weiteren bezeichne

$$\mathcal{QR}_n = \{z \in \mathbb{Z}_n^* \mid \text{es existiert ein } y \in \mathbb{Z}_n^* \text{ mit } y^2 = z \bmod n\}$$

die Menge der quadratischen Reste modulo n. Ferner heißt eine Primzahl p **sicher**, wenn es eine Primzahl p' gibt, sodass $p = 2p' + 1$ gilt.

Schlüsselerzeugung. Seien k und l Sicherheitsparameter und

$$(\mathsf{pk}, \mathsf{sk}) = ((n, a, b, c), p)\,,$$

wobei p, q zwei sichere Primzahlen der Länge k sind und $n = pq$ ein RSA-Modul. Ferner seien $a, b, c \in \mathcal{QR}_n$. Weiterhin sei l_m die Länge aller zulässigen Nachrichten und $l_s := l_n + l_m + l$ mit $l_n = 2k$.

Signaturalgorithmus. Bei Eingabe einer Nachricht m der Länge l_m und des Schlüsselpaares $((n, a, b, c), p)$ wird die Signatur in den folgenden Schritten berechnet:

- Es werden zufällig eine Primzahl e der Länge $l_e \geq l_m + 2$ und eine Zahl s der Länge l_s gewählt.
- Man berechnet v mit $v^e \equiv a^m b^s c \pmod{n}$.

Verifikationsalgorithmus. Bei Eingabe von m, $\sigma(m) = (e, s, v)$ und dem öffentlichen Schlüssel (n, a, b, c) sowie des Parameters l_e ist die Ausgabe des Verifikationsalgorithmus genau dann 1, wenn folgende Bedingungen erfüllt sind.

$$v^e \equiv a^m b^s c \pmod{n} \tag{1.1}$$

$$2^{l_e - 1} < e < 2^{l_e} \tag{1.2}$$

Wird der Signaturalgorithmus korrekt durchgeführt, so erfüllt das augegebene Tupel (e, s, v) offensichtlich die in der Verifikation geforderten Bedingungen. Man beachte, dass in der Verifikation nicht geprüft wird, ob e eine Primzahl ist. Damit ist eine Fälschung (e^*, s^*, v^*), wobei e^* keine Primzahl ist, möglich.

Für die Sicherheitsparameter wurden zum Zeitpunkt der Veröffentlichung folgende Größenordnungen als ausreichend angesehen: Die Autoren schlagen $l_n = 1024$, $l_m = 160$ und genauso $l = 160$ vor. Daraus ergeben sich die übrigen Parameter als $l_e = 162$ und $l_s = 1346$. Dabei haben die Parameter l und l_e eine eher technische Funktion im Sicherheitsbeweis: Der Parameter l sorgt dafür, dass die Verteilung von auf eine bestimmte Art und Weise erzeugten Werte s statistisch ununterscheidbar von einer Gleichverteilung auf der Menge aller Bitstrings der Länge l_s ist. Gibt es einen Angreifer auf die Signatur, so kann auf diese Weise die starke RSA-Annahme gebrochen werden. Ähnlich ist die Bedingung $l_e \geq l_m + 2$ motiviert.

Für eine ausführliche Beschreibung der Signatur und die Ausführung des Sicherheitsbeweises sei auf [CL02] verwiesen. Wir halten hier lediglich die Sicherheitseigenschaften der oben beschriebenen Signatur fest:

Die Camenisch-Lysyanskaya-Signatur ist unter der starken RSA-Annahme sicher gegen existentielle Fälschbarkeit unter einem adaptiven Angriff mit gewählten Nachrichten. (vgl. [CL02], Theorem 1)

1.1.4 Zero-Knowledge-Beweise

Zero-Knowledge-Beweise sind, grob gesagt, Interaktionen zwischen zwei Parteien, dem Verifizierer $\mathcal{V}$ und dem Prover $\mathcal{P}r$, die folgendes Ziel verfolgen: $\mathcal{P}r$ kennt ein Geheimnis und möchte $\mathcal{V}$ davon überzeugen, ohne sein Geheimnis preiszugeben. Man kann sich zum Beispiel eine Teilnehmerauthentifikation vorstellen, in der die Identität von $\mathcal{P}r$ als ein kryptographischer Schlüssel gegeben ist. Möchte $\mathcal{P}r$ sich gegenüber $\mathcal{V}$ authentifizieren, so muss er $\mathcal{V}$ davon überzeugen, dass er tatsächlich im Besitz seines Schlüssels ist. Andererseits liegt es jedoch in seinem Interesse, den Schlüssel nicht preiszugeben. Allgemein heißen Interaktionen zweier Turing-Maschinen, die den Zweck des Nachweises einer Behauptung erfüllen, interaktive Beweise. Besitzt der Beweis sogar die Eigenschaft, dass der Verifizierer während der Interaktion nichts neues über das Geheimnis erfahren hat, spricht man auch von Zero-Knowledge-Beweisen. Dies wird im Weiteren genauer formalisiert werden, wobei die Darstellung an [BNS05] angelehnt ist.

Definition 1.11 *Sei L eine Sprache, d.h. $L \subseteq \{0,1\}^*$. Ein Paar $(\mathcal{P}r, \mathcal{V})$ von interaktiven Turing-Maschinen heißt* **interaktiver Beweis** *für die Behauptung „$x \in L$“, wenn es* **durchführbar** *und* **korrekt** *ist, d.h.*

- *(Durchführbarkeit) Für alle ehrlichen Prover $\mathcal{P}r$, alle $x \in L$ der Länge k und alle vernachlässigbaren Funktionen ν gilt:*

$$\mathcal{P}(\mathcal{V} \text{ akzeptiert den Beweis}) \geq 1 - \nu(k).$$

- *(Korrektheit) Für alle interaktiven Turing-Maschinen $\mathcal{P}r$, alle $x \notin L$ der Länge k und alle vernachlässigbaren Funktionen ν gilt:*

$$\mathcal{P}(\mathcal{V} \text{ lehnt den Beweis ab}) \geq 1 - \nu(k).$$

In der Korrektheitsbedingung der obigen Definition ist es nicht relevant, welche interaktive Turing-Maschine mit $\mathcal{V}$ interagiert. Es sind also beliebige Prover $\mathcal{P}r^*$ zugelassen, deren Laufzeit, Rechen- und Speicherkapazität keinen Einschränkungen unterliegen. Lässt man hier nur effiziente Prover $\mathcal{P}r^*$ zu, so spricht man von **interaktiven Argumenten** für „$x \in L$“.

Von einem interaktiven Beweis, der die Zero-Knowledge-Eigenschaft besitzt, fordert man, dass es einen effizienten Algorithmus M, den sogenannten Simulator, gibt, der bei den gleichen Eingaben wie $\mathcal{V}$ die Protokollansicht des Verifizierers simulieren kann. Damit stellt man sicher, dass $\mathcal{V}$ aus der Interaktion mit $\mathcal{P}r$ keine Informationen gewinnt, die auf das Geheimnis von $\mathcal{P}r$ hinweisen. In der folgenden Definition

wird vorausgesetzt, dass der sogenannte Simulator M aus der Definition mit nicht vernachlässigbarer Wahrscheinlichkeit eine gültige Ausgabe hat.

Definition 1.12 *Es sei ν eine vernachlässigbare Funktion, L ein Sprache und $\Pi = (\mathcal{P}r, \mathcal{V})$ ein interaktiver Beweis für die Behauptung „$x \in L$“. Sei $z \in \{0,1\}^*$ die eigene Eingabe von $\mathcal{V}$. Dann heißt Π*

- *ein* **perfekter Zero-Knowledge-Beweis**, *wenn es für alle interaktiven probabilistischen Algorithmen $\mathcal{V}^*$ eine probabilistische polynomielle Turing-Maschine M gibt, sodass für alle $x \in L$ und $z \in \{0,1\}^*$ die Ausgaben von $M(x,z)$ und $view_{\mathcal{V}^*}(\Pi)$ identisch verteilt sind.*

- **statistischer Zero-Knowledge-Beweis**, *wenn es für alle effizienten Algorithmen $\mathcal{V}^*$ eine probabilistische polynomielle Turing-Maschine M gibt, die folgende Eigenschaft besitzt: Für alle $x \in L$ der Länge k Bit und für alle $z \in \{0,1\}^*$ gibt es eine vernachlässigbare Funktion ν, sodass gilt*

$$\sum_{\alpha} |\mathcal{P}(M(x,z) = \alpha) - \mathcal{P}(view_{\mathcal{V}^*}(\Pi) = \alpha)| = \nu(k).$$

- **rechnerischer Zero-Knowledge-Beweis**, *wenn es es für alle effizienten Algorithmen $\mathcal{V}^*$ eine probabilistische polynomielle Turing-Maschine M gibt, die folgende Eigenschaft besitzt: Für alle $x \in L$ der Länge k Bit, für alle $z \in \{0,1\}^*$ und für alle probabilistischen polynomiellen Algorithmen D gibt es eine vernachlässigbare Funktion ν, sodass gilt*

$$|\mathcal{P}(D(x,z,M(x,z)) = 1) - \mathcal{P}(D(x,z,view_{\mathcal{V}^*}(\Pi)) = 1)| = \nu(k).$$

Für interaktive Argumente definiert man die perfekte, die statistische und die rechnerische Zero-Knowledge-Eigenschaft analog.

Ein weiteres wichtiges Instrument sind die Proofs of Knowledge. Dies sind interaktive Beweise, für die ein probabilistisches polynomielles Orakel existiert, das in der Lage ist, durch Kommunikation mit $\mathcal{P}r$ das Geheimnis zu berechnen. Dabei hat das Orakel einen größeren Handlungsspielraum als andere Verifizierer, zum Beispiel kann es den Prover dazu zwingen, in zwei Protokolldurchführungen die gleiche Zufallszahl zu verwenden. Dieses Konzept ist insbesondere im Zusammenhang mit Zero-Knowledge Beweisen interessant. Hier spricht man auch von Zero-Knowledge-Proofs of Knowledge.
Für eine genauere Betrachtung von Zero-Knowledge-Beweisen sei auf die sehr ausführliche Darstellung in [Go03] verwiesen.

Zum Schluss dieses Abschnitts betrachten wir Commitmentschemata. Diese sind wesentlich für die Konstruktion von Zero-Knowledge-Beweisen, aber auch für die Konstruktion anderer kryptographischer Bausteine. Ein Commitmentschema $\mathcal{C}$ ist ein Tupel (G, C), wobei G ein Schlüsselerzeugungsalgorithmus ist, der einen öffentlichen Parameter v erzeugt. C ist ein interaktives Zwei-Parteien-Protokoll zwischen einem Sender $\mathcal{S}$ und einem Empfänger $\mathcal{R}$. Es besteht aus zwei Phasen, die man sich anschaulich wie folgt vorstellen kann: In der ersten Phase, der Commitphase, legt sich der Sender auf einen bestimmten Wert m fest, indem er das sogenannte commitment $\mathsf{com}(m)$ berechnet und dieses an den Empfänger schickt. Dazu verwendet er ggf. Zufallsbits r. Der Empfänger kennt zu diesem Zeitpunkt nur $\mathsf{com}(m)$ und pk, nicht aber m selbst. In der zweiten Phase, der Decommitphase, gibt der Sender zusätzliche Informationen preis, die es dem Empfänger ermöglichen, zu verifizieren, dass das Commitment auf die Nachricht m gebildet wurde. Im Allgemeinen wird die Decommitphase so realisiert, dass der Sender dem Empfänger die Werte m und ggf. r zuschickt. Wir betrachten hier nicht interaktive Commitmentschemata. Die Darstellung lehnt sich an die in [Ly02] an.

In der Beschreibung der Eigenschaft, dass $\mathcal{C}$ verbergend ist, verwenden wir wie bei den Verschlüsselungen ein Orakel $\mathcal{O}_b$. In diesem Fall bestimmt $\mathcal{O}_b$ korrekte Commitments auf Nachrichten, die vom Angreifer gewählt wurden. Im Gegensatz zu $\mathcal{O}_b$ ist das Orakel $\mathcal{O}$ wie oben dazu da, das dargestellte Spiel mit $\mathcal{A}$ zu spielen.

Definition 1.13 *Es sei $k \in \mathbb{N}$ ein Sicherheitsparameter. Es sei ein probabilistischer, polynomieller Algorithmus G gegeben, der bei Eingabe von 1^k einen öffentlichen Parameter v ausgibt. Weiterhin sei ein polynomieller Algorithmus com gegeben, der bei Eingabe einer Nachricht m, ggf. einer Zufallszahl r und des öffentlichen Parameters $v \in G(1^k)$ ein Commitment c ausgibt. Dann heißt das Paar $\mathcal{C} = (G, \mathsf{com})$ (nicht-interaktives public-key)* **Commitmentschema**. *Wir schreiben für c auch $c = \mathsf{com}(m)$. Das Commitmentschema $\mathcal{C}$ heißt* **sicher**, *falls die folgenden Bedingungen erfüllt sind:*

1. *($\mathcal{C}$ ist verbergend): Für jeden effizienten Algorithmus $\mathcal{A}$, der Zugriff auf ein Orakel $\mathcal{O}_b$ hat, gibt es eine in k vernachlässigbare Funktion ν, sodass für die Erfolgswahrscheinlichkeit $\varepsilon(k)$ von $\mathcal{A}$ im folgenden Spiel gilt:*

$$1/2 - \varepsilon(k) = \nu(k).$$

   (a) *Der Algorithmus G erzeugt bei Eingabe von 1^k den öffentlichen Parameter v. Ferner wird ein Bit $b \in_R \{0, 1\}$ gewählt.*

   (b) *Das Orakel $\mathcal{O}$ erhält als Eingabe v und b, beide Werte werden vor $\mathcal{A}$ geheimgehalten.*

(c) *$\mathcal{A}$ wählt $m_0, m_1 \in \mathcal{M}$ und sendet diese an das Orakel.*

(d) *$\mathcal{O}$ berechnet $c = \mathsf{com}(m_b)$.*

(e) *$\mathcal{A}$ erhält $\mathsf{com}(m_b)$. Er gibt ein Bit b' aus.*

(f) *$\mathcal{A}$ gewinnt das Spiel, wenn $b' = b$ ist.*

2. *$\mathcal{C}$ ist rechnerisch bindend, d.h. die Wahrscheinlichkeit, dass ein effizienter Algorithmus $\mathcal{A}$ zwei Nachrichten $m_0 \neq m_1$ mit $\mathsf{com}(m_0) = \mathsf{com}(m_1)$ findet, ist vernachlässigbar in k.*

In dem Fall, dass eine Zufallszahl r verwendet wird, beachte man, dass in der Definition der Sicherheitseigenschaften die Wahl der Zufallszahl r und die Ausführung von com zusammengefasst wurden: Mit $\mathsf{com}(m)$ ist gemeint, dass zunächst die Zufallszahl r gewählt wird und dann der Algorithmus com mit den oben beschriebenen Eingaben durchgeführt wird.
In beiden Sicherheitseigenschaften werden effiziente Algorithmen betrachtet, obwohl dies nicht zwingend erforderlich ist. Ist die zweiten Eigenschaft sogar für alle Algorithmen $\mathcal{A}$ erfüllt, so nennen wir $\mathcal{C}$ **perfekt bindend**. Das Commitmentschema $\mathcal{C}$ heißt **perfekt verbergend**, falls $\mathsf{com}(m)$ und m stochastisch unabhängig sind.

Man beachte, dass es nicht möglich ist, beide Eigenschaften gleichzeitig in der perfekten Variante zu erreichen.
Im Folgenden werden wir ein Commitment-Schema benötigen, das auf Dåmgard und Fujisaki [DF01] und auf Fujisaki und Okamoto [FO98] zurückgeht. Wir werden uns in der Darstellung jedoch an [CL02] halten.

Schlüsselerzeugungsalgorithmus. Bei Eingabe des Sicherheitsparameters k werden zwei sichere Primzahlen p und q gewählt und $n = pq$ berechnet. Ferner wird eine Element $h \in \mathcal{QR}_n$ gewählt sowie ein Element g aus der von h erzeugten Untergruppe von $\mathcal{QR}_n$. Der öffentliche Parameter v ist dann das Tripel (n, g, h).

Commitmentphase. Die Commitphase besteht wie oben beschrieben aus zwei Schritten:

1. Es wird eine Zufallszahl $r \in_R \mathbb{Z}_n$ gewählt.

2. Es wird der Algorithmus com mit Eingabe von m, r und (n, g, h) ausgeführt. Dieser bestimmt $c = \mathsf{com}(m) = g^m h^r \bmod n$.

Decommitphase. In der Decommitphase wird bei Eingabe von m, r und (n, g, h) wiederum der Wert $g^m h^r \bmod n$ berechnet und überprüft, ob dieser gleich c ist.

1.2 Informationstheoretische Grundlagen

In diesem Abschnitt werden die benötigten informationstheoretischen Grundlagen für die vorliegende Arbeit bereitgestellt. Während der gesamten Arbeit gehen wir stets von diskret auf endlichen Mengen verteilten Zufallsvariablen aus, ohne dies explizit zu erwähnen. Ist $\mathcal{T}$ eine endliche Menge und ist X gemäß einer Gleichverteilung auf $\mathcal{T}$ verteilt, so schreiben wir dafür kurz $X \in_R \mathcal{T}$. In diesem Zusammenhang ist mit der Bezeichnung $x \in_R \mathcal{T}$ die Realisierung der Zufallsvariable $X \in_R \mathcal{T}$ gemeint. Weiterhin sei erwähnt, dass zwar in diesem Abschnitt die übliche Schreibweise, in der Zufallsvariablen mit Großbuchstaben bezeichnet sind, verwendet wird. Dies wird jedoch in der weiteren Arbeit nicht immer beibehalten. So werden in Protokollen im Allgemeinen die Realisierungen von Zufallsvariablen betrachtet, dort werden also Kleinbuchstaben verwendet. Werden die Kleinbuchstaben in der Analyse der Protokolle beibehalten, auch wenn hier die zugrunde liegenden Zufallsvariablen relevant sind, so wird dies ausdrücklich erwähnt sein.

Bevor wir uns der Informationstheorie zuwenden, werden wir ein nützliches Ergebnis festhalten, das sich mittels elementarer Stochastik ergibt.

Lemma 1.1 *Seien $\mathcal{T}_1$ mit $n := |\mathcal{T}_1| < \infty$, $\mathcal{T}_2$ und $\mathcal{T}_3$ Mengen, X eine auf $\mathcal{T}_1$ gleichverteilte und Y eine auf $\mathcal{T}_2$ verteilte Zufallsgröße sowie $f_y : \mathcal{T}_1 \to \mathcal{T}_3$ für alle $y \in T_2$ eine invertierbare Abbildung. Sind X und Y unabhängig, so gilt*

(a) *Die Zufallsvariable $f_Y(X)$ ist gleichverteilt auf der Menge $\mathcal{T}_3$.*

(b) *$f_Y(X)$ und Y sind stochastisch unabhängig.*

Beweis. Wir zeigen zunächst Teil (a). Es bezeiche f_y^{-1} für alle $y \in \mathcal{T}_2$ die inverse Abbildung zu f_y. Da X gleichverteilt auf der Menge $\mathcal{T}_1$ mit Mächtigkeit n ist, gilt

$$\mathcal{P}(X = x) = \frac{1}{n}$$

für alle $x \in \mathcal{T}_1$. Man beachte, dass die Mengen $\mathcal{T}_1$ und $\mathcal{T}_3$ unter den oben genannten Voraussetzungen von gleicher Mächtigkeit sind. Es folgt für alle $z \in \mathcal{T}_3$:

$$\begin{aligned}
\mathcal{P}(f_Y(X) = z) &= \sum_{y \in \mathcal{T}_2} \mathcal{P}(\{f_Y(X) = z\} \cap \{Y = y\}) \\
&= \sum_{y \in \mathcal{T}_2} \mathcal{P}(\{f_y(X) = z\} \cap \{Y = y\}) \\
&= \sum_{y \in \mathcal{T}_2} \mathcal{P}(\{X = f_y^{-1}(z)\} \cap \{Y = y\}) \\
&= \sum_{y \in \mathcal{T}_2} \mathcal{P}(X = f_y^{-1}(z))\mathcal{P}(Y = y) = \sum_{y \in \mathcal{T}_2} \frac{1}{n}\mathcal{P}(Y = y) = \frac{1}{n}. \quad (1.3)
\end{aligned}$$

Dabei gilt in Gleichung (1.3) das erste Gleichheitszeichen wegen der Unabhängigkeit von X und Y. Das zweite Gleichheitszeichen folgt aus der Gleichverteilung von X auf $\mathcal{T}_1$.
Teil (b) folgt mit Teil (a): Sei $A \subseteq \mathcal{T}_3$ und $B \subseteq \mathcal{T}_2$. Dann gilt

$$\begin{aligned}
&\mathcal{P}(\{f_Y(X) \in A\} \cap \{Y \in B\}) \\
&= \sum_{z\in A, y\in B} \mathcal{P}(\{f_Y(X) = z\} \cap \{Y = y\}) \;=\; \sum_{z\in A, y\in B} \mathcal{P}(\{f_y(X) = z\} \cap \{Y = y\}) \\
&= \sum_{z\in A, y\in B} \mathcal{P}(\{X = f_y^{-1}(z)\} \cap \{Y = y\}) \\
&= \sum_{z\in A, y\in B} \mathcal{P}(X = f_y^{-1}(z))\mathcal{P}(Y = y) \;=\; \sum_{z\in A, y\in B} \frac{1}{n}\mathcal{P}(Y = y) \qquad (1.4) \\
&= \sum_{z\in A} \frac{1}{n} \sum_{y\in B} \mathcal{P}(Y = y) \;=\; \sum_{z\in A} \mathcal{P}(f_Y(X) = z) \sum_{y\in B} \mathcal{P}(Y = y) \qquad (1.5) \\
&= \mathcal{P}(f_Y(X) \in A)\mathcal{P}(Y \in B).
\end{aligned}$$

Das erste Gleichheitszeichen in Gleichung (1.4) gilt wegen der Unabhängigkeit von X und Y und das zweite wegen der Gleichverteiltheit von X auf $\mathcal{T}_1$. Weiterhin folgt das zweite Gleichheitszeichen in Gleichung (1.5), da $f_Y(X)$ nach Teil (a) auf $\mathcal{T}_3$ gleichverteilt ist. □

1.2.1 Entropie

In den nächsten beiden Abschnitten werden die Begriffe Entropie und Information kurz vorgestellt und einige grundlegende Eigenschaften angegeben. Die Ergebnisse sind im wesentlichen [ME81], [M77] und [R92] entnommen. Es sei daran erinnert, dass wir stets diskrete Zufallsvariablen mit endlicher Trägermenge betrachten.

Realisiert sich eine Zufallsgröße X in x, so gewinnt man gewisse Informationen über die Verteilung von x. Den Informationsgehalt von x kann man sich als Unsicherheit des Auftretens von x vor Durchführung des zugehörigen Zufallsexperiments vorstellen. Die Entropie ist schließlich der mittlere Informationsgehalt, den eine Realisierung von X besitzt. Dementsprechend kann man sich unter der gemeinsamen Entropie zweier Zufallsvariablen X und Y den mittleren Informationsgehalt der Realisierung des Paares (X, Y) vorstellen. Die bedingte Entropie von X unter Y ist hingegen der mittlere Informationsgehalt einer Realisierung von X unter der Bedingung, dass Y schon eingetreten ist. Formal sind diese Begriffe durch die folgende Definition gegeben.

Definition 1.14 *Seien X und Y auf $\mathcal{T}_X = \{x_1, \ldots, x_n\}$ bzw. $\mathcal{T}_Y = \{y_1, \ldots, y_m\}$, $n, m \in \mathbb{N}$, verteilte Zufallsvariablen mit den Massen*

$$p(x_1), \ldots, p(x_n) \quad \text{bzw.} \quad p(y_1), \ldots, p(y_m).$$

Dann heißt

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

Entropie von X,

$$H(X,Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i, y_j) \log_2 p(x_i, y_j)$$

heißt **gemeinsame Entropie von X und Y** *und*

$$H(X|Y) = H(X,Y) - H(Y)$$

heißt **bedingte Entropie von X unter Y.**

Als Beispiele betrachten wir die zwei möglichen Extremfälle: Ist X eine Zufallsvariable, die nur eine mögliche Realisierung x besitzt, so wird man nicht überrascht sein, wenn x eintritt, man würde also erwarten, dass die Entropie von X gerade 0 ist. Berechnet man $H(X)$, so erhält man in der Tat $H(X) = 0$. Andererseits wird man im Mittel aus Realisierungen einer Zufallsgröße X mit n Massen am meisten Information gewinnen, wenn alle Realisierungen mit der gleichen Wahrscheinlichkeit auftreten, da man hier am wenigsten vorhersagen kann, welche der n möglichen Realisierung beobachtet werden. Diese und weitere Eigenschaften der Entropie sind im nächsten Satz zusammengestellt.

Satz 1.2 *Es seien X und Y wie in Definition 1.14 gegeben. Dann gelten folgende Aussagen:*

(a) *Die gemeinsame Entropie zweier Zufallsvariablen ist symmetrisch:*

$$H(X,Y) = H(Y,X).$$

(b) *Es gilt $0 \leq H(X) \leq \log |\mathcal{T}_X|$.*

(c) *Es gilt $H(X,Y) \leq H(X) + H(Y)$.*

(d) *Es gilt $0 \leq H(X|Y) \leq H(X)$.*

(e) *Es gilt genau dann $H(X,Y) = H(X) + H(Y)$, wenn X und Y unabhängig sind.*

Beweis. Die Beweise zu den Aussagen finden sich z.B. in [R92]. □

Bemerkung 1.2 *Die gemeinsame Entropie von endlichen vielen diskret verteilten Zufallsvariablen definiert man analog zu Defintion 1.14. Somit lassen sich sowohl die Definitionen als auch die vorgestellten Ergebnisse auf Zufallsvektoren verallgemeinern, da die Verteilung von $(X_1, \ldots, X_n)$ als gemeinsame Verteilung der X_i definiert ist.*

Zum Schluss unserer Betrachtungen der Entropie halten wir das folgende nützliche Ergebnis fest:

Lemma 1.2 *Es seien $\mathcal{T}_1$, $\mathcal{T}_2$ und $\mathcal{T}_3$ endliche Mengen, X eine Zufallsgröße, die sich in $\mathcal{T}_1$ realisiert und Y eine Zufallsgröße mit Trägermenge $\mathcal{T}_2$ sowie $f : \mathcal{T}_1 \to \mathcal{T}_3$ eine Abbildung. Dann gelten die folgenden Gleichungen:*

$$H(X, f(X)) = H(X) \tag{1.6}$$
$$H(X, Y, f(X)) = H(X, Y). \tag{1.7}$$

Beweis. Wir zeigen zunächst Gleichung (1.6). Für alle $x \in \mathcal{T}_1$ und $z \in \mathcal{T}_3$ gilt

$$\mathcal{P}(X = x, f(X) = z) = \mathcal{P}(X = x, f(x) = z) = \begin{cases} \mathcal{P}(X = x), & \text{falls } f(x) = z \\ 0, & \text{sonst.} \end{cases}$$

Damit folgt

$$\begin{aligned} H(X, f(X)) &= -\sum_{x \in \mathcal{T}_1} \sum_{z = f(x)} \mathcal{P}(X = x, f(X) = z) \log \mathcal{P}(X = x, f(X) = z) \\ &= -\sum_{x \in \mathcal{T}_1} \sum_{z = f(x)} \mathcal{P}(X = x) \log \mathcal{P}(X = x) \\ &= -\sum_{x \in \mathcal{T}_1} \mathcal{P}(X = x) \log \mathcal{P}(X = x) \sum_{z = f(x)} 1 \\ &= H(X) \sum_{z = f(x)} 1 = H(X). \end{aligned}$$

Gleichung (1.7) zeigt man analog unter Verwendung von

$$\begin{aligned} &\mathcal{P}(X = x, Y = y, f(X) = z) \\ &\quad = \mathcal{P}(X = x, Y = y, f(x) = z) = \begin{cases} \mathcal{P}(X = x, Y = y), & \text{falls } f(x) = z, \\ 0, & \text{sonst,} \end{cases} \end{aligned}$$

für alle $x \in \mathcal{T}_1$, $y \in \mathcal{T}_2$ und $z \in \mathcal{T}_3$. □

1.2.2 Information

Ist man an einem Maß für den Grad der Abhängigkeit zweier Zufallsvektoren interessiert, so motiviert Aussage (e) aus Satz 1.2 die folgende

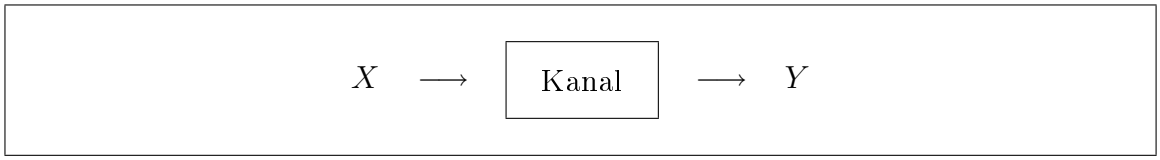
Definition 1.15 *Seien X und Y Zufallsvektoren. Dann heißt*

$$\mathcal{I}[X,Y] := H(X) + H(Y) - H(X,Y)$$

die **gemeinsame Information von X und Y**.

Die gemeinsame Information zweier Zufallsvariablen X und Y ist also die Information, die man im Mittel durch eine Realisierung von X gewinnt, abzüglich der Information, die man im Mittel durch eine Realisierung von X gewinnt, falls sich Y bereits realisiert hat. Man kann sich somit vorstellen, dass $\mathcal{I}(X,Y)$ die Information ist, die Y über X enthält. Sind X und Y unabhängig, sollte Y keine Information über X enthalten, d.h. in diesem Fall würden wir erwarten, dass $\mathcal{I}(X,Y) = 0$ ist. Dies lässt sich auch formal zeigen. Bevor wir die wichtigsten Eigenschaften der Information festhalten, betrachten wir zunächst zur Anschauung eine typische Problemstellung aus der Informationstheorie:

Es sei eine Datenquelle X gegeben. Die gemäß X erzeugten Daten werden über einen Kanal an einen Empfänger verschickt. Abhängig von den Störungen, denen der Kanal unterliegt, kommen Daten Y bei dem Empfänger an. Man interessiert sich nun dafür, wie viel Information der Kanal „verschluckt“, d.h. man ist gerade interessiert daran, wie viel Information über die gesendeten Daten noch in den empfangenen Daten enthalten sind. Im Allgemeinen ist dabei nicht bekannt, welchen Störungen der Kanal unterliegt. Beispielsweise kann man sich in diesem Kontext die Kommunikation mit einem Satelliten vorstellen, der gewisse Daten an seine Basisstation sendet. Dabei kommen an der Basisstation im Allgemeinen gestörte Daten an, wobei die Störungen durch verschiedene Einflüsse erzeugt sein können. Hier ist von Interesse, herauszufinden, wie groß die Störungen im Mittel sind, bzw. wie viel Information an der Basisstation ankommt.



Satz 1.3 Eigenschaften der Information
Seien X und Y Zufallsvektoren. Dann gelten folgende Aussagen:

(a) *Die Information ist symmetrisch: $\mathcal{I}[X,Y] = \mathcal{I}[Y,X]$.*

(b) *Es gilt $0 \leq \mathcal{I}[X,Y]$.*

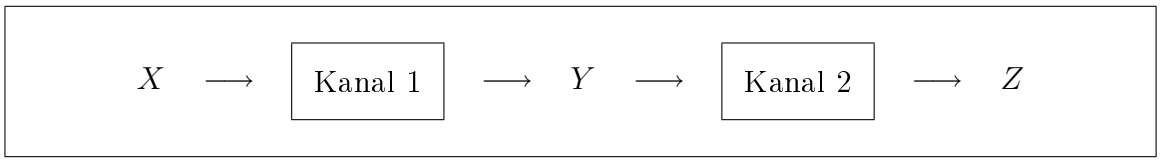
(c) *Es ist genau dann $\mathcal{I}[X,Y] = 0$, wenn X und Y unabhängig sind.*

Beweis. Die Behauptungen folgen mit Satz 1.2 direkt aus der Definition und

$$\mathcal{I}[X,Y] = H(X) + H(Y) - H(X,Y) = H(X) - H(X|Y).$$

□

Das für diese Arbeit wichtigste Ergebnis ist der **Hauptsatz der Datenverarbeitung**. Anschaulich befasst sich dieser Satz mit der Situation zweier hintereinandergeschalteter Kanäle. Die Aussage des Hauptsatzes der Datenverarbeitung ist, dass unter gewissen Voraussetzungen in einer solchen Kombination zweier Kanäle nicht mehr Information übertragen werden kann, als durch jeden der einzelnen Kanäle.



Dies werden wir im Folgenden genauer formulieren. Dazu betrachten wir zunächst etwas allgemeiner eine endliche Folge von n auf einer endlichen Menge $\mathcal{T}$ diskret verteilten Zufallsgrößen $(X_i)_{i=0}^n$. Dann besitzt die Folge $X_0, X_1, \ldots, X_n$ die **Markov-Eigenschaft**, wenn für alle $0 \leq i < n$ und alle $x_0, \ldots x_{i+1} \in \mathcal{T}$ mit $\mathcal{P}(X_i = x_i, \ldots, X_0 = x_0) > 0$ die folgende Gleichung erfüllt ist:

$$\mathcal{P}(X_{i+1} = x_{i+1} | X_i = x_i, \ldots X_0 = x_0) = \mathcal{P}(X_{i+1} = x_{i+1} | X_i = x_i). \tag{1.8}$$

Damit können wir den Hauptsatz der Datenverarbeitung formulieren.

Satz 1.4 *Sei (X,Y,Z) eine Folge von Zufallsgrößen, welche die Markov-Eigenschaft besitzt. Dann gilt*

$$\mathcal{I}[X,Z] \leq \begin{cases} \mathcal{I}[X,Y] \\ \mathcal{I}[Y,Z]. \end{cases}$$

Beweis. Siehe zum Beispiel [ME81]. □

Teil I

Blinde interaktive Signaturen

Kapitel 2

Blinde digitale Signaturen

Dieses Kapitel wird einen kurzen Überblick über die für die Arbeit relevanten Ergebnisse bezüglich blinder digitaler Signaturen geben: Zunächst werden zwei Beispiele blinder Signaturen vorgestellt. Daraufhin werden eine formale Definition blinder Signaturen und ein allgemeines Sicherheitsmodell für blinde Signaturen angegeben. Zum Schluss dieses Kapitels wird der Rahmen für unsere Untersuchungen abgesteckt.

Blinde digitale Signaturen wurden 1982 von David Chaum zum ersten Mal beschrieben ([Ch82]). Als Anschauung schlug Chaum das folgende Szenario vor: Der Empfänger einer Signatur steckt die zu signierende Nachricht zusammen mit einem Kohlepapier in einen Briefumschlag und schickt diesen zu dem Signierer. Der Signierer schickt den mit seiner Unterschrift versehenen Briefumschlag zurück zum Empfänger, der durch das Kohlepapier eine Unterschrift auf die Nachricht erhalten hat. Der Signierer hat die Nachricht nie gesehen und kann sie, wenn er zu einem späteren Zeitpunkt mit der Signatur konfrontiert wird, nicht mehr einem konkreten Signaturvorgang zuordnen. Dies bedeutet insbesondere, dass der Signierer nicht in der Lage ist zu bestimmen, für wen er die Signatur ausgestellt hat. In Abbildung 2.1 ist diese Vorgehensweise schematisch dargestellt.

Dieser Vorgehensweise folgend wurden in den letzten Jahren weitere blinde Signaturen entwickelt, darunter [Oka92], [CPS94], [Fe93], [HMP94b], [HP94], [HMP95], [JLO97] und [CKW04]. In den folgenden Abschnitten werden die zwei Varianten vorgestellt, die wir in dieser Arbeit näher betrachten.

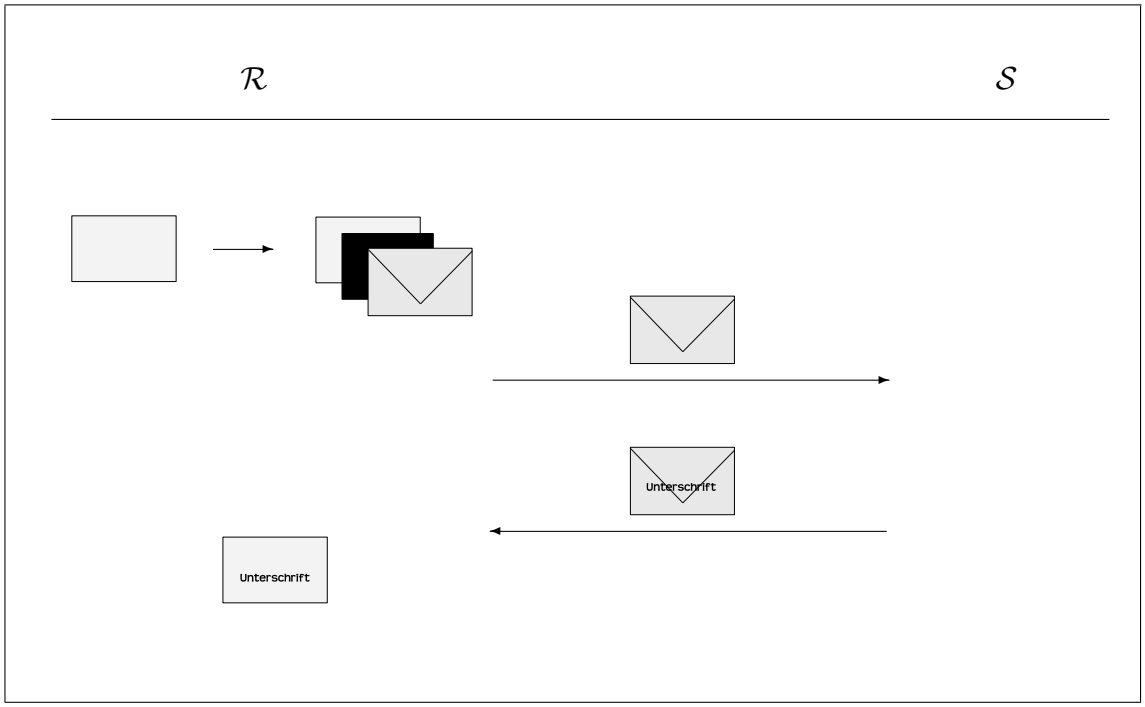


Abbildung 2.1: Idee der RSA-Signatur nach Chaum

2.1 Die blinde RSA-Signatur von Chaum

Als Umsetzung seiner Idee auf Protokollebene schlug Chaum das im Folgenden dargestellte Verfahren vor. Dazu sei eine öffentliche Hash-Funktion $\mathcal{H} : \{0,1\}^* \longrightarrow \mathbb{Z}_n$ gegeben.

Schlüsselerzeugung. Der Signierer $\mathcal{S}$ führt bei Eingabe des Sicherheitsparameters 1^k den Schlüsselerzeugungsalgorithmus G der RSA-Signatur mit dem Ergebnis durch, dass er im Besitz zweier Primzahlen p und q der Länge k sowie ihrem Produkt $n = p \cdot q$, eines öffentlichen Schlüssels (n, e) und eines geheimen Schlüssels d ist, wobei $((n, e), d)$ die in Abschnitt 1.1.3.1 beschriebenen Eigenschaften besitzt.

Signaturprotokoll. Das Signaturprotokoll ist in Abbildung 2.2 dargestellt. Nach der Durchführung des Protokolls erhält $\mathcal{R}$ das Nachrichten-Signatur-Paar (m, s).

Verifikation. Der Verifikationsalgorithmus gibt bei Eingabe des öffentlichen Schlüssels (n, e) und des Nachrichten-Signatur-Paares (m, s) genau dann 1 aus, wenn gilt

$$s^e = \mathcal{H}(m) \pmod{n}.$$

Die blinde Chaum-Signatur erreicht den größtmöglichen Blindheitsgrad: Wenn der

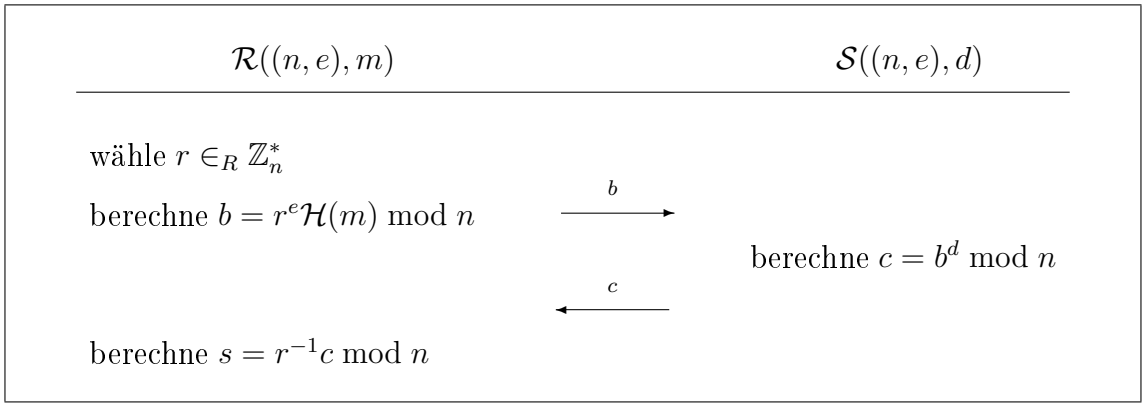


Abbildung 2.2: Signaturprotokoll der RSA-Signatur nach Chaum

Signierer eine Signatur zur Verifikation erhält, kann er nur raten, zu welchem Signaturprotokoll diese passt. Diese Eigenschaft werden wir später formal erfassen und sie **perfekte Blindheit** nennen. Die Sicherheitseigenschaften dieser Signatur wurden in [BNPS03] ausführlich untersucht. Dort wurde gezeigt, dass die Chaum-Signatur unter der *known-target*-RSA-Annahme im Random-Oracle-Modell rechnerisch sicher gegen one-more-Fälschungen ist. Auf den Begriff der one-more-Fälschung werden wir im Weiteren noch genauer eingehen.

Es gibt weitere blinde Signaturen, die auf der RSA-Signatur beruhen, wie zum Beispiel die von Ferguson, die Anwendung in dessen 1993 vorgestellten Münzsystem findet ([Fe93]). Diese blinde Signatur hat die Eigenschaft, dass der Empfänger der Signatur die zu signierende Nachricht nicht vollständig selbst bestimmen kann.

2.2 Die blinde Schnorr-Signatur

Zur Schnorr-Signatur gibt es mehrere blinde Varianten. Die Idee zur Blendung von Schnorr-Signaturen geht auf Okamoto zurück ([Oka92]). Die hier vorgestellte Variante stammt von Horster ([HMP94b]). Sie bildet in dieser Form die Basis des Geldsystems von Brands ([Br93]). Wie bei der Chaum-Signatur wird eine öffentliche Hash-Funktion $\mathcal{H} : \mathbb{Z}_p \times \{0,1\}^* \to \mathbb{Z}_q$ benötigt.

Schlüsselerzeugung. Bei Eingabe des Sicherheitsparameters 1^k wird der Schlüsselerzeugungsalgorithmus der Schnorr-Signatur aus Abschnitt 1.1.3.2 durchgeführt. Die Ausgabe ist das so erzeugte Schlüsselpaar

$$(\mathsf{pk}, \mathsf{sk}) = ((p,q,g,y),x)\,,$$

das die in Abschnitt 1.1.3.2 beschriebenen Eigenschaften besitzt.

Signaturprotokoll. Das Signaturprotokoll ist Abbildung (2.3) zu entnehmen, wobei der dort verwendete Wert u^{-1} die multiplikative Inverse zu u modulo q ist. Die Ausgabe des Empfängers $\mathcal{R}$ ist das Nachrichten-Signatur-Paar $(m,(r,s))$.

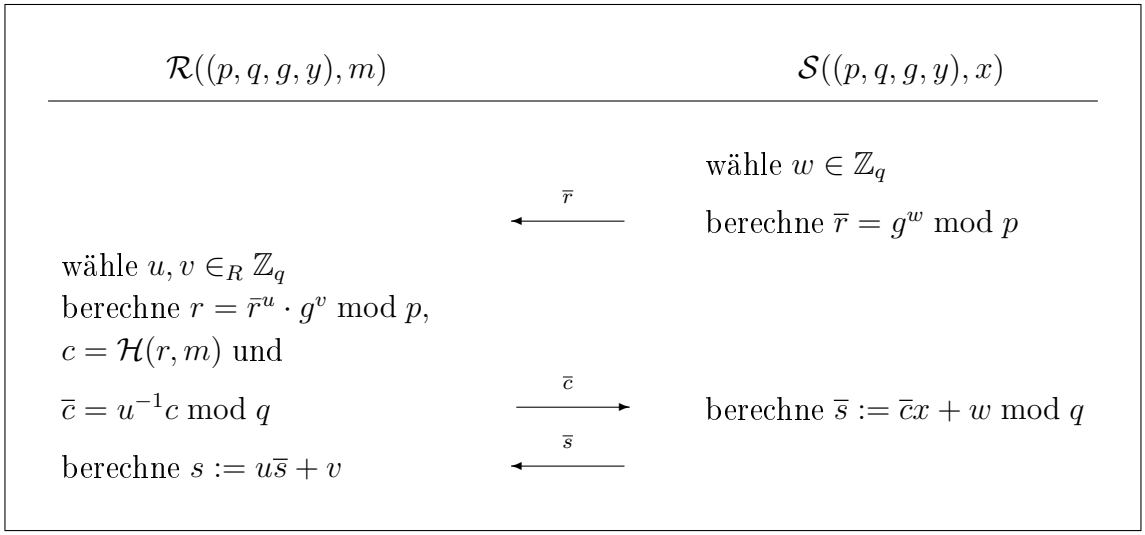


Abbildung 2.3: Signaturprotokoll der blinden Schnorr-Signatur

Verifikationsalgorithmus. Der Verifikationsalgorithmus gibt bei Eingabe des öffentlichen Schlüssels (p,q,g,y) und des Nachrichten-Signatur-Paares $(m,(r,s))$ genau dann 1 aus, wenn die folgende Gleichung gilt:

$$g^s = y^{\mathcal{H}(r,m)} r \pmod{p}.$$

Die perfekt blinde Schnorr-Signatur steht stellvertretend für perfekt blinde Signaturen vom Fiat-Shamir-Typ. Der in [Oka92] angegebene Vorschlag zur Blendung der Schnorr-Signatur lässt sich für alle Signaturen umsetzen, die aus Authentifikationsprotokollen hervorgehen, wie zum Beispiel das Fiat-Shamir-Protokoll, die Schnorr-Identifikation und andere. Ein vollständiger Überblick aller dieser blinden Signaturen findet sich in [Sch96].

Im Gegensatz zur blinden RSA-Signatur ist noch kein expliziter Sicherheitsbeweis für die blinde Schnorr-Signatur bekannt. Es wurde jedoch in [BP02] die Sicherheit der interaktiven Variante (mit einer challenge $c \in \{0,1\}^{\mathsf{poly}(k)}$ statt des im Protokoll aufgeführten Hashwertes $c = \mathcal{H}(r,m)$) ohne Blendung unter der *known-target-inversion*-DL-Annahme gezeigt.

Die oben vorgestellte Variante erreicht wie die Chaum-Signatur perfekte Blindheit.

Nachdem wir zwei blinde Signaturen kennengelernt haben, wird im folgenden Abschnitt dargestellt, was formal unter einer blinden Signatur zu verstehen ist und welche Sicherheitseigenschaften diese haben sollte.

2.3 Sicherheit blinder digitaler Signaturen

Die formale Beschreibung von blinden Signaturen und ein allgemeines Sicherheitsmodell wurden in [JLO97] angegeben. Die Ergebnisse sollen im Weiteren kurz wiedergegeben werden. Es bezeichne $\mathcal{M}$ die Menge aller zulässigen Nachrichten. Formal besteht eine blinde Signatur aus einem Schlüsselerzeugungsalgorithmus, einem Signaturprotokoll und einem Verifikationsalgorithmus. Dies beschreibt die folgende Definition genauer.

Definition 2.1 *Es sei $k \in \mathbb{N}$ ein Sicherheitsparameter und es seien G und V zwei Algorithmen mit den folgenden Eigenschaften:*

1. *Der* **Schlüsselerzeugungsalgorithmus** *G sei ein probabilistischer, polynomieller Algorithmus, der bei Eingabe des Sicherheitsparameters 1^k ein Schlüsselpaar $(\mathsf{pk}, \mathsf{sk})$ ausgibt.*

2. *Der* **Verifikationsalgorithmus** *V sei ein deterministischer, polynomieller Algorithmus, der bei Eingabe von (pk, m, s) die Werte $1 = (\mathsf{true})$ oder $0 = (\mathsf{false})$ ausgibt.*

Ferner seien $\mathcal{S}$ und $\mathcal{R}$ zwei probabilistische polynomielle interaktive Turing-Maschinen, die ein interaktives Protokoll, das **Signaturprotokoll**, *durchführen, dessen Rundenzahl polynomiell in k ist. Für $(\mathsf{pk}, \mathsf{sk}) \in G(1^k)$ sei pk die gemeinsame Eingabe von $\mathcal{R}$ und $\mathcal{S}$, sk die eigene Eingabe von $\mathcal{S}$ sowie $m \in \mathcal{M}$ die eigene Eingabe von $\mathcal{R}$. Die Ausgabe von $\mathcal{R}$ sei entweder $\sigma(m)$ oder fail. Im ersten Fall sei die Ausgabe von $\mathcal{S}$ $\mathsf{completed}$, sonst $\mathsf{not\ completed}$.*
Das 4-Tupel $(G, \mathcal{S}, \mathcal{R}, V)$ heißt **blinde digitale Signatur**, *falls für alle Nachrichten m und alle $(\mathsf{pk}, \mathsf{sk}) \in G(1^k)$ folgende Bedingung erfüllt ist:*

Sind $\mathcal{S}$ und $\mathcal{R}$ dem Protokoll gefolgt, so hat $\mathcal{R}$ als Ausgabe $\sigma(m)$ und es gilt

$$\mathcal{P}(V(\mathsf{pk}, m, \sigma(m)) = 1) = 1,$$

wobei sich die Wahrscheinlichkeit auf die Inhalte der Zufallsbänder von G, $\mathcal{R}$ und $\mathcal{S}$ bezieht.

Die erste genauere Untersuchung der Sicherheitsanforderungen des Signierers an blinde Signaturen stammt von Pointcheval und Stern (in [PS00]). Die Autoren unterscheiden zunächst grundsätzlich zwischen zwei Angriffsarten:

1. Der **sequentielle** Angriff wird durchgeführt, indem sich der Angreifer unabhängig voneinander verschiedene Signaturen ausstellen lässt.

2. Der **parallele** Angriffe wird durchgeführt, indem sich der Angreifer gleichzeitig und möglicherweise abhängig voneinander Signaturen ausstellen lässt. Das heißt, er kann zu einem beliebigen Zeitpunkt ein neues Signaturprotokoll starten und die Nachrichten in beliebiger Abhängigkeit von zuvor erhaltenen Nachrichten-Signatur-Paaren wählen. Offensichtlich ist der parallele Angriff der stärkere der beiden Angriffe.

Pointcheval und Stern formulieren die folgende Sicherheitsanforderung an eine blinde Signatur: Nach ℓ Protokolldurchführungen des Signaturprotokolls darf kein (effizienter) Angreifer in der Lage sein, $\ell + 1$ Signaturen auf verschiedene Nachrichten vorzuweisen. Dabei spielt die Größe von ℓ offensichtlich eine entscheidende Rolle, da sie die Rechenkapazität des Angreifers betrifft: Die Autoren unterscheiden

1. **die $\ell, (\ell+1)-$Fälschung:** Ein Angreifer $\mathcal{A}$ produziert $\ell + 1$ Signaturen nach ℓ Interaktionen mit dem Signierer $\mathcal{S}$.

2. **die one-more-Fälschung:** $\mathcal{A}$ produziert $\ell+1$ Signaturen nach ℓ Interaktionen mit $\mathcal{S}$, wobei ℓ polynomiell in einem Sicherheitsparameter k beschränkt ist.

3. **die starke one-more-Fälschung:** $\mathcal{A}$ produziert $\ell + 1$ Signaturen nach ℓ Interaktionen mit $\mathcal{S}$, wobei es eine Konstante c gibt, sodass $\ell \leq (\log k)^c$ gilt. Dabei ist k der Sicherheitsparameter der Signatur.

Aufbauend auf [PS00] wurde in [JLO97] die folgende Sicherheitsdefinition für blinde Signaturschemata vorgestellt, die Unfälschbarkeit im Sinne einer one-more-Fälschung formalisiert. Man beachte, dass in dem vorgeschlagenen Sicherheitsmodell die Eigenschaften der Unfälschbarkeit und der Blindheit einer blinden Signatur im folgenden Sinne zusammengefasst sind: Die Erfolgswahrscheinlichkeit eines Algorithmus' $\mathcal{A}$ wird in beiden Spielen in beiden Fällen durch die gleiche vernachlässigbare Funktion ν beschrieben.

Definition 2.2 *Es sei $k \in \mathbb{N}$ ein Sicherheitparameter. Ein blindes digitales Signaturschema ist* **sicher**, *falls es für alle probabilistischen, polynomiellen Algorithmen $\mathcal{A}$ eine vernachlässigbare Funktion ν gibt, sodass die beiden folgenden Bedingungen erfüllt sind.*

Blindheit: *Es sei $\mathcal{O}$ ein Orakel, das in dem folgenden Spiel die Rolle des Empfängers übernimmt. Dann gilt für die Erfolgswahrscheinlichkeit $\varepsilon(k)$ von $\mathcal{A}$ in dem folgenden Spiel gegen einen ehrlichen Empfänger $\mathcal{R}$:*

$$1/2 - \varepsilon(k) = \nu(k).$$

1. *Der Empfänger $\mathcal{R}$ wählt ein Bit $b \in_R \{0,1\}$. Das Orakel $\mathcal{O}$ erhält als Eingabe das Bit b.*
2. *Der Angreifer $\mathcal{A}$ erzeugt durch Eingabe des Sicherheitsparameters 1^k in den Schlüsselerzeugungsalgorithmus G ein Schlüsselpaar:*

$$(\mathsf{pk}, \mathsf{sk}) \in G(1^k).$$

3. *$\mathcal{A}$ erzeugt zwei Nachrichten m_0 und m_1, deren Länge polynomiell in k ist, und die von pk oder sk abhängen können.*
4. *$\mathcal{A}$ sendet (m_0, m_1) und pk an das Orakel.*
5. *$\mathcal{A}$ führt parallel und nicht zwingend unabhängig voneinander das interaktive Protokoll σ_I mit dem Orakel $\mathcal{O}$ aus. Dabei übernimmt das Orakel die Rolle von zwei ehrlichen Empfängern $\mathcal{R}_b$ und $\mathcal{R}_{1-b}$. Die Eingabe von $\mathcal{A}$ in beiden Protokollen ist $(1^k, \mathsf{pk}, \mathsf{sk}, m_0, m_1)$, die Eingabe von $\mathcal{O}$ in der Rolle von $\mathcal{R}_b$ ist (pk, m_b), die von $\mathcal{O}$ in der Rolle von $\mathcal{R}_{1-b}$ ist (pk, m_{1-b}).*
6. *Ist die Ausgabe des Orakels in beiden Protokolldurchführungen $\sigma(m_b)$ bzw. $\sigma(m_{1-b})$, so erhält der Angreifer von $\mathcal{O}$ die Nachrichten-Signatur-Paare*

$$((m_0, \sigma_I(m_0)), (m_1, \sigma_I(m_1)).$$

7. *$\mathcal{A}$ gibt ein Bit b' aus.*
8. *Der Angreifer $\mathcal{A}$ gewinnt das Spiel, wenn $b = b'$ gilt.*

Unfälschbarkeit: *Für die Erfolgswahrscheinlichkeit $\varepsilon(k)$ von $\mathcal{A}$ in folgendem Spiel gilt $\varepsilon(k) = \nu(k)$. Dabei hat $\mathcal{A}$ in dem beschriebenen Spiel Zugriff auf $\mathcal{R}$, jedoch nicht auf $\mathcal{S}$, und versucht, eine neue Signatur zu erzeugen.*

1. *$\mathcal{S}$ führt G mit Eingabe 1^k aus, um ein Schlüsselpaar $(\mathsf{pk}, \mathsf{sk})$ zu erhalten.*
2. *$\mathcal{A}$ führt adaptiv parallele und nicht notwendigerweise unabhängige Protokolle mit identischen Kopien von $\mathcal{S}$ (jeweils mit Eingabe $(\mathsf{pk}, \mathsf{sk})$) aus, wobei $\mathcal{A}$ adaptiv entscheidet, wann der Angriff aufhört. Dabei ist die Anzahl der Protokolldurchführungen polynomiell in k. Sei l die Anzahl der Protokolldurchführungen, in denen $\mathcal{S}$ als Ausgabe $\mathsf{completed}$ hat.*

3. *$\mathcal{A}$ gibt gültige Nachrichten-Signatur-Paare $(m_1, \sigma(m_1)), \ldots (m_j, \sigma(m_j))$ zu paarweise verschiedenen Nachrichten aus, d.h. es gilt*

$$V(\mathsf{pk}, m_i, \sigma(m_i)) = 1$$

für alle $1 \leq i \leq j$.

4. *$\mathcal{A}$ gewinnt das Spiel, falls $j > l$ ist.*

Diese Definition wird in der späteren Arbeit von großer Wichtigkeit sein, sodass einige Erläuterungen von Nutzen sind:

Bemerkung 2.1

Erläuterungen zur Blindheitseigenschaft. *In [JLO97] wird festgehalten, dass die vorgestellte Definition implizit die Angriffsphase erfasst: Wir betrachen einen effizienten Angreifer $\mathcal{A}$, der in einer Vorphase eine gewisse Anzahl von mehreren Signaturprotokollen parallel und nicht zwingend unabhängig bezüglich selbst gewählter Nachrichten mit den entsprechenden Empfängern durchführt, wobei die Anzahl der Protokolldurchführungen polynomiell in k ist. Erst nach dieser Vorphase führt $\mathcal{A}$ das Spiel aus der Definition durch. Weicht die Erfolgswahrscheinlichkeit $\varepsilon(k)$ von $\mathcal{A}$ nicht vernachlässigbar von $1/2$ ab, so gibt es einen Algorithmus $\mathcal{A}^*$, der unter Verwendung von $\mathcal{A}$ das oben beschriebene Spiel gewinnt: Der Algorithmus $\mathcal{A}^*$ übernimmt in der Vorphase die Rolle des Empfängers und beginnt das Spiel mit dem Orakel erst dann, wenn $\mathcal{A}$ zwei Nachrichten m_0 und m_1 generiert hat. Da $\mathcal{A}$ die Protokollmitschriften und die so erzeugten Nachrichten-Signatur-Paare effizient mit Wahrscheinlichkeit $\varepsilon(k)$ unterscheiden kann, gilt das gleiche auch für $\mathcal{A}^*$.*
Gibt es umgekehrt einen Angreifer, der ohne eine vorangegangene Angriffsphase das angegebene Spiel mit nicht vernachlässigbar von $1/2$ abweichender Erfolgswahrscheinlichkeit gewinnt, so gewinnt er das Spiel mit Angriffsphase erst recht.
Man beachte, dass in der Angriffsphase auch sequentielle Protokolldurchführungen des Angreifers möglich sind.

Erläuterungen zur Unfälschbarkeit. *Die paarweise Verschiedenheit der Nachrichten wird in [JLO97] nicht gefordert. Diese Bedingung geht auf [CKW04] zurück. Hier wird vorgeschlagen, die Signatur* **stark unfälschbar** *zu nennen, falls die Nachrichten-Signatur-Paare paarweise verschieden sind.*

Die am Anfang des Kapitels vorgestellten blinden Signaturen sind sogar perfekt blind, d.h. sie erfüllen sogar die zum Beispiel von [CPS94] vorgestellte Definition

von Blindheit, die ohne eine Beschränkung der Rechenkapazität, der Laufzeit oder des Speicherplatzes eines Angreifers auf die Blindheit auskommt:

Definition 2.3 *Ein blindes Signaturschema heißt* **perfekt blind**, *wenn die Protokollansicht des Signierers während des interaktiven Protokolls stochastisch unabhängig von* $(m, \sigma(m))$ *ist.*

Rechnerisch blinde Signaturen sind vor allem aus dem Kontext der so genannten fairen blinden Signaturen bekannt (z.B. [CPS95]), da sich die Fairness-Eigenschaft nicht parallel zu perfekt blinden Signaturen umsetzen lässt. Rechnerisch blinde Signaturen, die nicht aus diesem Zusammenhang stammen, wurden in [JLO97] und [CKW04] vorgestellt. Beide Konstruktionen beruhen auf sicheren Zwei-Parteien-Protokollen, in denen die Ausgabe des Empfängers eine Signatur auf eine zuvor geheim eingegebene Nachricht ist.

2.4 Fragestellungen

Dieser Abschnitt dient dazu, die Ziele der Arbeit klar zu umreißen. Dazu werden wir auf zwei Ebenen Fragestellungen formulieren.

Zunächst kann man sich auf einer grundsätzlichen Ebene fragen, ob die oben angegebene Definition die abstrakten Vorgaben einer blinden Signatur vollständig ausschöpft. Es sei daran erinnert, dass an blinde Signaturen im wesentlichen drei Anforderungen gestellt werden:

1. Der Empfänger möchte die gültige Unterschrift eines Signierers auf ein Dokument erhalten.

2. Jede beliebige dritte Partei soll in der Lage sein, die Unterschrift zu verifizieren.

3. Der Signierer darf nicht in der Lage sein, den Signaturprozess mit Daten in Verbindung zu bringen, die er später erhält.

Dabei sind die ersten beiden Bedingungen gleichzusetzen mit der Forderung nach einer Signatur. Erst in der dritten Eigenschaft wird Bezug auf die Blindheit genommen. In den oben vorgestellten Vorschlägen wird diese Bedingung ausschließlich während des Signaturprozesses umgesetzt. Allerdings ist es grundsätzlich denkbar, dass die Blindheitsbedinung auch während des Verifikationsprozesses umgesetzt wird, etwa in Form eines Zero-Knowledge-Beweises für die Behauptung „Ich kenne eine Signatur“. Die Analyse dieses allgemeineren Ansatzes ist ein Ziel dieser Arbeit.

Auf der zweiten Ebene stellt sich innerhalb des beschriebenen Ansatzes die Frage nach geeigneten Mechanismen, die zur Blindheit einer vorgegebenen Signatur führen. Dabei sind sowohl die perfekte als auch die rechnerische Blindheit in Betracht zu ziehen. Wir haben oben gesehen, dass es sowohl für perfekt blinde als auch für rechnerisch blinde Signaturen Realisierungen gibt. Dies beantwortet jedoch nicht die Frage nach dem Zusammenhang der Begriffe im Falle einer gegebenen Signatur. So sind weder zur Chaum- noch zur Schnorr-Signatur rechnerisch blinde Varianten bekannt. Damit ist es sowohl für rechnerisch als auch für perfekt blinde Signaturen wünschenswert, Rahmenbedingungen für die Umsetzung der Blindheit zu finden, um so den Charakter blinder Signaturen herauszustellen und ggf. den Zusammenhang verschiedener Varianten zu beleuchten. Dies ist ein weiteres Ziel der Arbeit.

Es sei darauf hingewiesen, dass der Schwerpunkt dieser Fragen auf der Blindheit der Signatur liegt. Auch wenn die oben angegebene Definition suggeriert, dass diese Anforderung parallel zur Unfälschbarkeit in einem blinden Signaturschema erreicht werden sollten, sind die benötigten Mechanismen im Falle der Unfälschbarkeit und der Blindheit doch verschieden. Damit ist es sinnvoll, beide Sicherheitsanforderungen getrennt zu betrachten.

Kapitel 3

Interaktive Signaturschemata

Ein Ziel dieser Arbeit ist, blinde digitale Signaturen zu untersuchen. Wie man an den Beispielen aus Kapitel 2 sehen kann, unterscheiden diese sich in einem Punkt wesentlich von digitalen Signaturen:

Bei der Berechnung einer digitalen Signatur werden in erster Linie die Sicherheitsanforderungen des Signierers berücksichtigt. Dementsprechend hat der Empfänger einer Signatur keinen Einfluss auf die Berechnung einer Signatur. Erst in der Verifikation wird ihm Handlungsspielraum eingeräumt, um seine eigenen Interessen zu wahren: er kann die Signatur ablehnen, wenn die Verifikation nicht gelingt. Blinde Signaturen sind hingegen Signaturen, in denen zusätzlich die Anonymität des Empfängers einer Signatur gegenüber dem Signierer gewahrt werden soll.

Um die zusätzlichen Sicherheitsanforderungen des Empfängers in einer blinden Signatur kryptographisch umzusetzen, muss man bei der Konstruktion den zusätzlichen Handlungsbedarf des Empfängers beachten. In den bekannten blinden Signaturen wird dieser dem Empfänger während der Signatur eingeräumt. Die Anonymität des Empfängers ist jedoch nur angreifbar, wenn er die Signatur einer dritten Partei übermittelt, die ebenfalls die Gültigkeit der Signatur feststellen kann. Damit ist ebenso vorstellbar, dass sich der Empfänger stärker in diesen Prozess einbringt, um seine Anonymität zu wahren. In beiden Fällen ist eine Interaktion des Empfängers mit dem Signierer bzw. der dritten Partei unumgänglich. Man kann diese Interaktionen der Teilnehmer somit als grundlegend für jede blinde Signatur ansehen.

Im folgenden Abschnitt 3.1 werden interaktive Signaturen eingeführt, ein Konzept, das diesen Überlegungen Rechnung trägt. Daraus ergibt sich die Notwendigkeit, die Sicherheitsanforderungen der Teilnehmer eines solchen Systems zu untersuchen, was in Abschnitt 3.2 geschieht. Dabei wird hier das Augenmerk zunächst auf der

Unfälschbarkeit der Signatur liegen. Die ausführliche Untersuchung der Blindheitseigenschaft folgt in den nächsten Kapiteln.

3.1 Definition von interaktiven Signaturschemata

Wie bei einfachen Signaturen sind in einem interaktiven Signaturverfahren drei Parteien beteiligt: Ein Signierer $\mathcal{S}$, der Empfänger der Signatur $\mathcal{R}$ und eine dritte Partei, der Verifizierer $\mathcal{V}$. Dabei besitzt der Signierer ein Schlüsselpaar, bestehend aus einem öffentlichen Schlüssel pk und einem geheimen Schlüssel sk, und ist so in der Lage, Datensätze mit seiner Unterschrift zu versehen. Eine interaktive Signatur besteht damit aus drei Schritten: Zunächst erzeugt der Signierer mit dem Schlüsselerzeugungsalgorithmus sein Schlüsselpaar $(\mathsf{pk}, \mathsf{sk})$. Im zweiten Schritt, dem Signaturprotokoll, interagieren der Empfänger $\mathcal{R}$ und der Signierer $\mathcal{S}$ mit dem Ergebnis, dass $\mathcal{R}$ einen von $\mathcal{S}$ unterzeichneten Datensatz besitzt. $\mathcal{R}$ interagiert nun mit einem Verifizierer $\mathcal{V}$, um die Gültigkeit der Signatur dem Verifizierer gegenüber nachzuweisen. Dies tut er im dritten Schritt, dem Verifikationsprotokoll.

Dementsprechend wird in der folgenden Definition ein interaktives Signaturschema als Tripel bestehend aus dem Schlüsselerzeugungsalgorithmus, dem Signatur- und dem Verifikationsprotokoll beschrieben (vgl. Abbildung 3.1). Wie in allen kryptographischen Bausteinen erfasst die Definition einer interaktiven Signatur zunächst grundsätzlich nicht die Sicherheitseigenschaften des Systems. Die einzige Anforderung, die an die Protokolle gestellt werden muss, ist die der Durchführbarkeit: Das System soll das gewünschte Ergebnis liefern, wenn sich alle Parteien korrekt verhalten.

Definition 3.1 *Es sei $k \in \mathbb{N}$ ein Sicherheitsparameter, $\mathcal{M}$ eine Menge und $\mathcal{S}$ (Signierer), $\mathcal{R}$ (Empfänger) sowie $\mathcal{V}$ (Verifizierer) polynomiell beschränkte interaktive Turing-Maschinen. Ferner seien der* **Schlüsselerzeugungsalgorithmus** *G, das* **Signaturprotokoll** *σ_I und das* **Verifikationsprotokoll** *V_I wie in (a) bis (c) beschrieben.*

(a) *G sei ein probabilistischer, polynomieller Algorithmus, der bei Eingabe von 1^k ein Paar $(\mathsf{pk}, \mathsf{sk})$ ausgibt. Wir schreiben $(\mathsf{pk}, \mathsf{sk}) \in G(1^k)$ und nennen pk* **öffentlichen Schlüssel** *sowie sk* **geheimen Schlüssel**.

(b) *σ_I sei ein Protokoll mit polynomieller Rundenzahl zwischen $\mathcal{S}$ mit Eingabe $(\mathsf{pk}, \mathsf{sk})$ und $\mathcal{R}$ mit Eingaben pk und m. Nach der Durchführung von σ_I sei die Ausgabe von $\mathcal{R}$ entweder $\sigma_I(m)$ oder fail. Im ersten Fall sei die Ausgabe von $\mathcal{S}$ $\mathsf{completed}$, sonst $\mathsf{not\ completed}$.*

(c) *V_I sei ein Protokoll mit polynomieller Rundenzahl zwischen $\mathcal{R}$ mit Eingabe $(\mathsf{pk}, m, \sigma_I(m))$ und $\mathcal{V}$ mit Eingabe* pk. *Nach der Durchführung von V_I sei die Ausgabe von $\mathcal{V}$ entweder* okay *oder* fail. *Im ersten Fall sei die Ausgabe von $\mathcal{R}$* completed, *sonst* not completed. *Gibt $\mathcal{V}$* okay *aus, so sprechen wir von $\sigma_I(m)$ als* **gültige Signatur** *und sagen, $\mathcal{V}$* **akzeptiert** *die Signatur.*

Dann heißt $\Sigma_I := (G, \sigma_I, V_I)$ **interaktives Signaturschema** *oder* **interaktive Signatur**, *falls die Protokolle σ_I und V_I für alle $(\mathsf{pk}, \mathsf{sk}) \in G(1^k)$ durchführbar sind: Sind $\mathcal{S}$, $\mathcal{R}$ und $\mathcal{V}$ Teilnehmer, die sich an die Vorgaben der Protokolle halten, so sind die folgenden beiden Bedingungen mit höchstens vernachlässigbarer Wahrscheinlichkeit nicht erfüllt:*

1. *Für alle $(\mathsf{pk}, \mathsf{sk}) \in G(1^k)$ ist im Protokoll σ_I die Ausgabe von $\mathcal{R}$ mit Eingabe (pk, m) eine gültige Signatur.*

2. *Für alle $\sigma_I(m)$, die durch das Protokoll σ_I generiert wurden, ist die Ausgabe von $\mathcal{R}$ mit Eingabe $(m, \sigma_I(m))$ bei einer Durchführung von V_I* completed.

Bemerkung 3.1 *Grundsätzlich sind auch interaktive Signaturen denkbar, in denen die Teilnehmer $\mathcal{S}$, $\mathcal{R}$ oder $\mathcal{V}$ als zusätzliche Eingaben öffentliche oder private Parameter* par *haben können, die jedoch genau wie das Schlüsselpaar nur einmal festgelegt und dann in jedem Signatur- bzw. Verifikationsprotokoll verwendet werden. Insbesondere $\mathcal{R}$ kann unter Umständen die Eingabe eines privaten Parameters in σ_I für die Umsetzung einer blinden interaktiven Signatur nutzen. Falls dies in einem Protokoll der Fall ist, wird dies stets in der ersten Zeile zu erkennen sein, in der sowohl die Teilnehmer als auch ihre Eingaben aufgeführt sind.*

Nach der Einführung von interaktiven Signaturen stellt sich die Frage nach der Existenz solcher Signaturen. Die Definition von interaktiven Signaturen über Protokolle bedeutet im Prinzip, dass wir digitale Signaturen einschließlich aller relevanten Kommunikationsschritte betrachten. Daher ist es plausibel, dass jedes digitale Signaturschema zu mindestens einer interaktiven Signatur führt. Ferner erwartet man, dass Signaturschemata, die aus mehreren Rechenschritten des Signierers bestehen, mehrere triviale Varianten haben können. Diese Überlegungen sollen durch die folgende Bemerkung noch einmal untermauert und verdeutlicht werden.

Darüber hinaus sei an dieser Stelle bereits vermerkt, dass sich blinde Signaturen, wie die in Kapitel 2 vorgestellten, ebenfalls als interaktive Signaturen schreiben lassen. Dies werden wir genauer in Kapitel 6 untersuchen. Da ein Ziel dieser Arbeit ist, eine adäquate Formalisierung blinder Signaturen bereitzustellen, ist diese Beobachtung von besonderer Wichtigkeit.

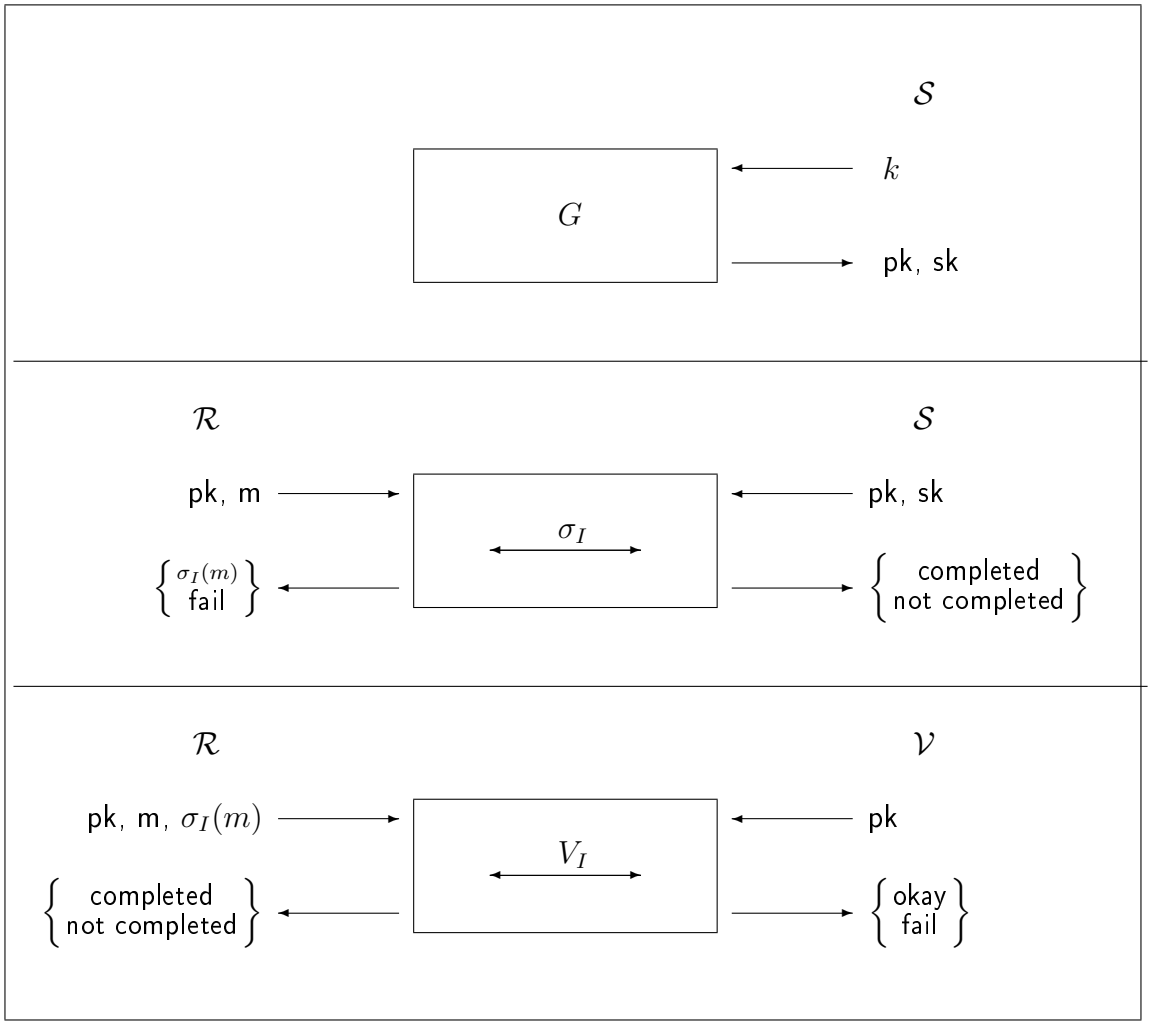


Abbildung 3.1: Interaktive Signatur

Bemerkung 3.2 *Zu jedem Signaturschema $\Sigma = (G, \sigma, V)$ existiert ein interaktives Signaturschema: Man wähle als Schlüsselerzeugungsalgorithmus von Σ_I den Algorithmus G und als Signatur- und Verifikationsprotokoll die in Abbildung 3.2 bzw. 3.3 dargestellten.*
Weiterhin halten wir fest, dass es zu einem vorgegebenen Signaturschema verschiedene interaktive Varianten geben kann. Wir betrachten das folgende Beispiel: Sei (G, σ, V) die Schnorr-Signatur (vgl. Abschnitt 1.1.3.2) und

$$(\mathsf{pk}, \mathsf{sk}) = ((p, q, g, y), x) \in G(1^k)$$

das Schlüsselpaar des Signierers. Sei das Signaturprotokoll σ_I wie in Abbildung 3.4. Hier wurden die durch den Signaturalgorithmus vorgegebenen Schritte auf die beiden teilnehmenden Parteien verteilt, sodass hier eine zweite, leicht veränderte Variante

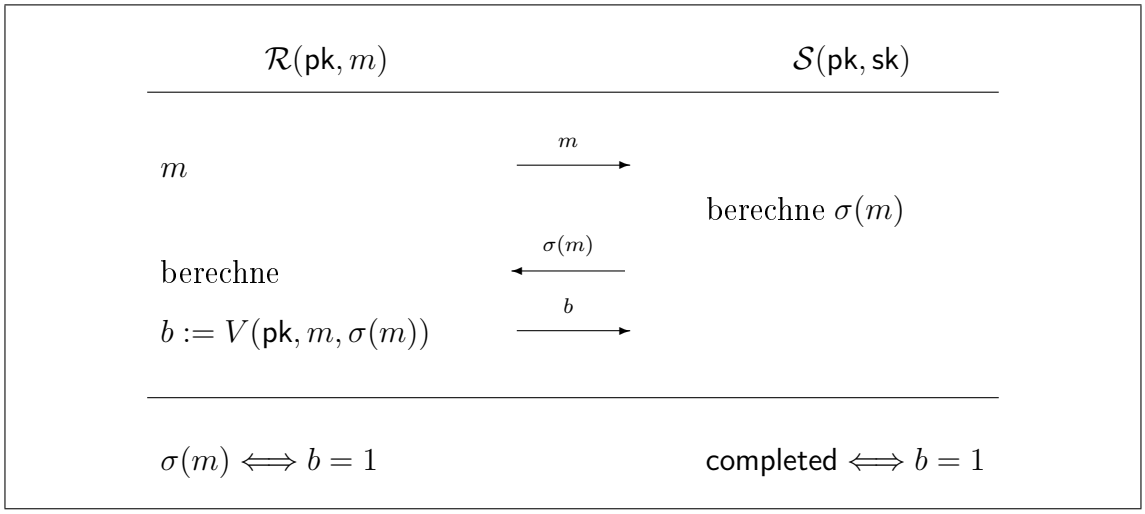


Abbildung 3.2: Signaturprotokoll σ_I

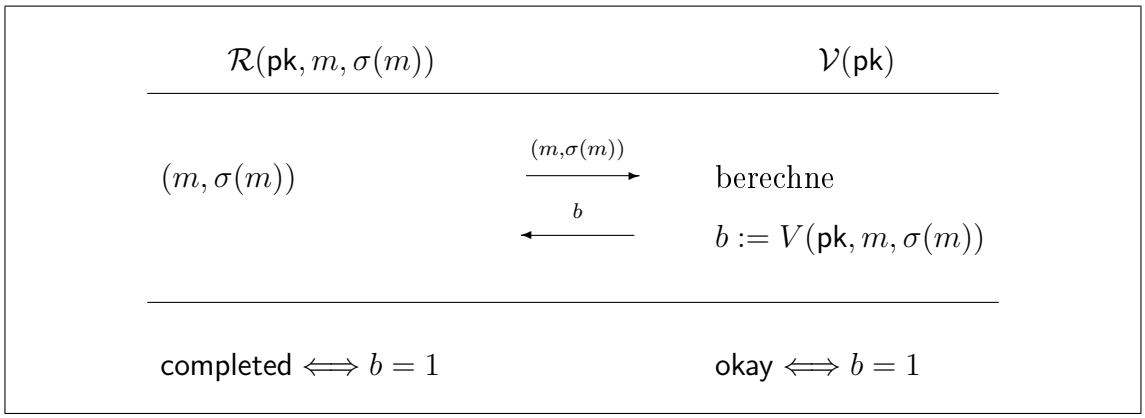


Abbildung 3.3: Verifikationsprotokoll V_I

vorliegt. Das Verifikationsprotokoll bestehe wie oben aus der Übergabe der Signatur an den Verifizierer.

Zum Schluss dieses Abschnitts ist noch eine Bemerkung zu den Abbildungen zu machen: In der letzten Zeile sind die Ausgaben der Teilnehmer aufgeführt. Diese Angaben sind eigentlich kein Teil der Protokolle. In den hier angegebenen Beispielen wurden die Angaben zur Verdeutlichung der Vorgehensweise in einer interaktiven Signatur gemacht, im Weiteren werden sie jedoch nicht mehr als Teil der Protokolle angegeben werden.

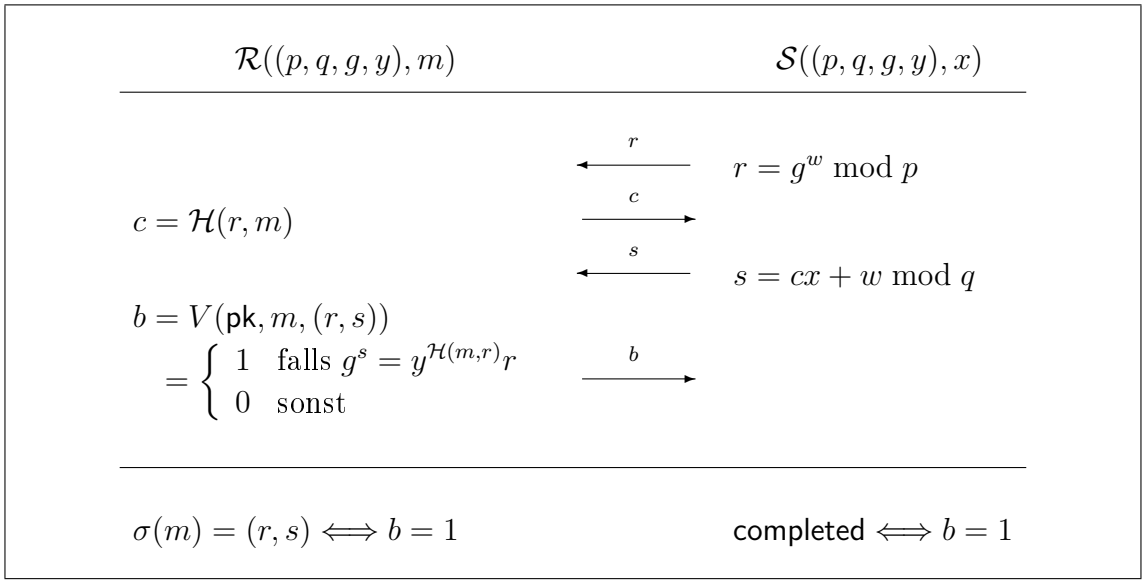


Abbildung 3.4: Signaturprotokoll σ_I der Schnorr-Signatur

3.2 Sicherheit von interaktiven Signaturschemata

Nachdem das Konzept der interaktiven Signaturen eingeführt wurde, ist die Frage nach einem adäquaten Sicherheitsmodell zu stellen. In diesem Abschnitt werden mögliche Angriffe und Erfolgsstufen kurz skizziert und eine formale Definition der Sicherheit einer interaktiven Signatur angegeben. Man beachte, dass sich der Sicherheitsbegriff hier sowohl auf die Interessen des Signierers $\mathcal{S}$ als auch auf die des Verifizierers $\mathcal{V}$ bezieht.

Da blinde digitale Signaturen bereits ein Signaturprotokoll verwenden und darüber hinaus in der Literatur (s. [PS00],[JLO97]) bereits bzgl. ihrer Sicherheitseigenschaften untersucht wurden, können wir uns bei unseren Untersuchungen auf die in Kapitel 2 vorgestellten Ergebnisse stützen.

In der Public-Key-Kryptographie versteht man unter sicheren Bausteine solche, die dem stärksten denkbaren effizienten Angreifer nicht einmal den geringst möglichen Erfolg bieten. Das Ziel ist also, den stärksten Angriff und den geringst mögliche Erfolg in der Situation einer interaktiven Signatur zu finden. Damit können wir dann einen adäquaten Sicherheitsbegriff formulieren.

Wir untersuchen, welche Fälschungen von interaktiven Signaturen grundsätzlich möglich sind. Dazu erklären wir zunächst, was unter einer Fälschung einer interaktiven Signatur zu verstehen ist.

Sei $\Sigma_I = (G, \sigma_I, V_I)$ eine interaktive Signatur und $\mathcal{A}$ ein Angreifer. Eine **Fälschung** (m^*, s^*) wird wie folgt charakterisiert:

(a) Der Angreifer $\mathcal{A}$ hat (m^*, s^*) nicht durch eine Interaktion mit dem Signierer $\mathcal{S}$ erhalten.

(b) Ein ehrlicher Verifizierer $\mathcal{V}$ lehnt (m^*, s^*) mit höchstens vernachlässigbarer Wahrscheinlichkeit ab.

Wir können also im Weiteren davon ausgehen, dass der Verifizierer $\mathcal{V}$ sich stets korrekt verhält.

Die Erfolgstypen eines Angreifers unterscheiden sich nicht von denen, die ein Angreifer auf Signaturen erreichen kann (vgl. Abschnitt 1.1.3):

1. Existentielle Fälschbarkeit: Der Angreifer kann zu einer Nachricht, die er nicht notwendig selbst ausgesucht hat, eine Fälschung generieren.
2. Selektive Fälschbarkeit: Der Angreifer kann zu einer oder mehreren Nachrichten seiner Wahl Fälschungen generieren.
3. Universelle Fälschbarkeit: Der Angreifer kann zu jeder Nachricht seiner Wahl eine Fälschung generieren.
4. Kompromittierung des Schlüssels: Der Angreifer kann den privaten Schlüssel bestimmen.

Damit haben wir die verschiedenen Erfolgsstufen eines Angreifers identifiziert. Den geringsten Erfolg erzielt ein Angreifer, der eine existentielle Fälschung produziert.

Damit können wir uns nun den Angriffstypen zuwenden. Auch hier unterscheiden wir zunächst die Angriffstypen, die wir bereits von den Signaturen kennen:

1. Angriffe ohne bekannte Signaturen,
2. Angriffe mit bekannten Signaturen,
3. Angriffe mit gewählten Nachrichten,
4. adaptive Angriffe mit gewählten Nachrichten.

Da Σ_I aus interaktiven Protokollen besteht, können alle Angriffe, die bekannte Signaturen voraussetzen (Angriffe 2 bis 4), auf unterschiedliche Weise vom Angreifer durchgeführt werden. Hier sind, wie bei den blinden Signaturen, sowohl sequentielle als auch parallele Angriffe möglich. Wie bereits erwähnt wurde, sind parallele Angriffe die stärkeren Angriffe. Damit ist der stärkste Angriff ein paralleler adaptiver Angriff mit gewählten Nachrichten.

Wir fassen die Ergebnisse zusammen: Eine interaktive Signatur ist als sicher anzusehen, wenn es einem effizienten Angreifer $\mathcal{A}$ nicht möglich ist, unter einem parallelen adaptiven Angriff mit gewählten Nachrichten eine existentielle Fälschung zu erzeugen.

Man beachte, dass dieser Begriff gerade dem Konzept der one-more-Fälschung von Pointcheval und Stern entspricht.

Damit kommen wir zu folgender formalen Sicherheitsdefinition für interaktive Signaturen, in der zwei Orakel $\mathcal{O}_\mathcal{S}$ und $\mathcal{O}_\mathcal{V}$ vorgesehen sind, um zu garantieren, dass der Angreifer im Signaturprotokoll bzw. im Verifikationsprotokoll Informationen aus korrekt durchgeführten Protokollen erhält. Dabei übernimmt $\mathcal{O}_\mathcal{S}$ die Rolle des Signierers und $\mathcal{O}_\mathcal{V}$ die Rolle des Verifizierers.

Definition 3.2 *Es sei $k \in \mathbb{N}$ ein Sicherheitsparameter. Ein interaktives Signaturschema $\Sigma_I = (G, \sigma_I, V_I)$ heißt* **unfälschbar**, *falls die Erfolgswahrscheinlichkeit eines effizienten Angreifers, der Zugriff auf zwei Orakel $\mathcal{O}_\mathcal{S}$ und $\mathcal{O}_\mathcal{V}$ hat, im folgenden Spiel gegen einen Signierer $\mathcal{S}$ vernachlässigbar in k ist.*

1. *Der Signierer $\mathcal{S}$ führt G mit Eingabe 1^k aus, um ein Schlüsselpaar $(\mathsf{pk}, \mathsf{sk})$ zu generieren.*
2. *Das Orakel $\mathcal{O}_\mathcal{S}$ erhält als Eingabe $(\mathsf{pk}, \mathsf{sk})$, $\mathcal{O}_\mathcal{V}$ hat als Eingabe pk.*
3. *$\mathcal{A}$ führt adaptiv, parallel und nicht notwendigerweise unabhängig $\mathsf{poly}(k)$-mal die Protokolle σ_I und V_I mit $\mathcal{O}_\mathcal{S}$ bzw. mit $\mathcal{O}_\mathcal{V}$ durch. Dabei hält sich $\mathcal{O}_\mathcal{S}$ an die Schritte von $\mathcal{S}$ in σ_I und $\mathcal{O}_\mathcal{V}$ an die Schritte von $\mathcal{V}$ in V_I. $\mathcal{A}$ entscheidet adaptiv, wann der Angriff aufhört. Sei ℓ die Anzahl der Protokolldurchführungen, in denen $\mathcal{O}_\mathcal{S}$ im Protokoll σ_I als Ausgabe $\mathsf{completed}$ hat, und j die Anzahl der Protokolldurchführungen, in denen $\mathcal{O}_\mathcal{V}$ als Ausgabe okay hat. Dabei müssen die Nachrichten, die $\mathcal{A}$ im Protokoll V_I als Eingabe hat, paarweise verschieden sein.*
4. *$\mathcal{A}$ gewinnt das Spiel, wenn $j > \ell$ ist.*

Analog zur Unfälschbarkeit blinder Signaturen, nennen wir eine interaktive Signatur **stark unfälschbar**, wenn die Nachrichten-Signatur-Paare, die ein Angreifer $\mathcal{A}$ im Verifikationsprotokoll eingibt, paarweise verschieden sind.

Da das Konzept der interaktiven Signatur insbesondere für die Behandlung von blinden Signaturen eingeführt wurde, stellt sich die Frage, ob die Sicherheitseigenschaften von blinden Signaturen erhalten bleiben, wenn man diese als interaktive Signaturen auffasst. Dazu die folgende

Bemerkung 3.3 *Die Sicherheitseigenschaften (unabhängig von dem Angriff und der Erfolgsstufe) von blinden digitalen Signaturen, wie denen, die in Kapitel 2 vorgestellt wurden, bleiben generell erhalten, wenn man diese Signaturen als interaktive Signaturen auffasst (vgl. Kapitel 6): In diesem Fall besteht das Verifikationsprotokoll lediglich aus der Übergabe der in σ_I produzierten Signatur, und der Verifizierer $\mathcal{V}$ gibt genau dann* okay *aus, wenn die Verifikation gelingt. Damit kann ein Angreifer $\mathcal{A}$ eine dritte Partei $\mathcal{V}$ genau dann davon überzeugen, dass eine gefälschte Signatur gültig ist, wenn diese Signatur die Verifikationsgleichung erfüllt. Die Sicherheitseigenschaften der interaktiven Signatur sind in diesem Fall also nicht vom Verifikationsprotokoll abhängig, sondern nur von der Verifikationsgleichung. Dies ist aber gerade die Situation, die man bei den Sicherheitsbetrachtungen einer blinden Situation zu berücksichtigen hat.*

In Bemerkung 3.2 wurde ausgeführt, dass interaktive Signaturen in gewissem Sinne eine Verallgemeinerung von Signaturen darstellen. Es stellt sich hier die Frage, ob auch der vorgestellte Sicherheitsbegriff für interaktive Signaturschemata eine Verallgemeinerung von dem in Kapitel 1.1.3 vorgestellten Sicherheitsbegriff für Signaturen ist. Diese Frage wird in der nächsten Bemerkung beantwortet.

Bemerkung 3.4 *Wir betrachten eine interaktive Signatur Σ_I zu einem Signaturschema $\Sigma = (G, \sigma, V)$ gemäß Bemerkung 3.2. Hier gibt es nur zwei Kommunikationsschritte im Signaturprotokoll. Im ersten Schritt schickt der Empfänger $\mathcal{R}$ die Nachricht an $\mathcal{S}$, und im zweiten erhält er schon die Signatur zurück. Damit wird aber ein paralleler adaptiver Angriff mit gewählten Nachrichten gerade zu einem adaptiven Angriff mit gewählten Nachrichten. Die Sicherheit von interaktiven Signaturen stellt also in diesem Sinne eine Verallgemeinerung der Sicherheit von Signaturen dar.*

3.3 Fazit

Es wurde das Konzept der interaktiven Signaturen eingeführt. Diese stellen im folgenden Sinne eine Verallgemeinerung von digitalen Signaturen dar: Jede digitale

Signatur lässt sich als interaktive Signatur darstellen, indem man die Kommunikationsschritte zwischen den Teilnehmern mit in Betracht zieht. Diese sind in der Definition digitaler Signaturen (vgl. Definition 1.9) nicht erfasst. Offenbar ist die Umkehrung nicht der Fall: Blinde Signaturen lassen sich wohl als interaktive Signaturen auffassen, erfüllen im engeren Sinne jedoch nicht die Definition einer digitalen Signatur.

Ferner wurden die Sicherheitsanforderungen an interaktive Signaturen untersucht und ein Sicherheitsmodell für interaktive Signaturen entwickelt. Es wurde festgestellt, dass die Einbindung der Kommunikationsschritte im Vergleich zu den digitalen Signaturen neue Angriffsmöglichkeiten auf die interaktive Signatur ermöglicht, die für einen adäquaten Sicherheitsbegriff in Betracht gezogen werden müssen. Weiterhin wurde diskutiert, inwiefern die Sicherheit von interaktiven Signaturen mit der von Signaturen und blinden Signaturen zusammenhängt. In beiden Fällen ist das Ergebnis, dass sich die Sicherheitseigenschaften auf die entsprechenden interaktiven Signaturen übertragen.

Wir haben damit einen verallgemeinerten Begriff von digitalen Signaturen und einen verallgemeinerten Sicherheitsbegriff angegeben, der sich in seiner Struktur und bezüglich der Sicherheit als Fundament für die Untersuchung von blinden Signaturschemata eignet. In den nächsten beiden Kapiteln werden wir die Blindheitseigenschaft in interaktiven Signaturen untersuchten.

Kapitel 4

Perfekt blinde interaktive Signaturen

Nachdem im vorherigen Kapitel interaktive Signaturen behandelt wurden, wird im Folgenden der Begriff der perfekten Blindheit interaktiver Signaturen diskutiert. In den in Kapitel 2 beschriebenen blinden Signaturen wird die Blindheit einzig im Signaturprotokoll erreicht. Durch den größeren Handlungsspielraum, der dem Empfänger einer Signatur in interaktiven Signaturen eingeräumt wird, ist es nun vorstellbar, dass sich Blindheit auch über ein geeignetes Verifikationsprotokoll erreichen lässt. Dazu stellen sich folgende Fragen:

1. Ist jede perfekt blinde Signatur auch eine perfekt blinde interaktive Signatur?
2. Muss die Blindheit grundsätzlich im Signaturprotokoll erreicht werden?

Um diese Fragen zu beantworten, wird zunächst in Abschnitt 4.1 der Begriff der perfekten Blindheit erklärt und formal definiert. In Abschnitt 4.2 werden wir sehen, dass sich die erste Frage positiv beantworten lässt, d.h. das Konzept der interaktiven blinden Signatur eignet sich insbesondere, um blinde Signaturen zu beschreiben. Der zweiten Frage gehen wir in Abschnitt 4.3 nach, mit dem Ergebnis, dass Blindheit grundsätzlich auch im Verifikationsprotokoll erreicht werden kann. In beiden Fällen werden wir hinreichende Bedingungen für Signatur- bzw. Verifikationsprotokolle angeben, die zu einer perfekt blinden interaktiven Signatur führen.

Bemerkung 4.1 *Da die Blindheit eine Sicherheitsanforderung des Empfängers $\mathcal{R}$ einer Signatur ist, gehen wir ab jetzt stets davon aus, dass der Empfänger sich korrekt verhält: Ist eine interaktive Signatur $\Sigma_I = (G, \sigma_I, V_I)$ gegeben, so hält sich der Empfänger stets an die von σ_I bzw. V_I vorgegebenen Schritte und er hat als Eingaben in V_I ausschließlich gültige Signaturen, also Signaturen, die ein ehrlicher Verifizierer mit höchstens vernachlässigbarer Wahrscheinlichkeit ablehnt.*

Die Kapazitäten eines betrügerischen Signierers $\mathcal{S}$ unterliegen bei den folgenden Betrachtungen keinerlei Einschränkungen, d.h. auch wenn ein ehrlicher Empfänger das Signatur- bzw. Verifikationsprotokoll mit einem rechnerisch unbeschränkten Signierer bzw. Verifizierer durchführen würde, hätten unsere Überlegungen noch Bestand. Obwohl in der Definition von interaktiven Signaturen alle beteiligten Parteien als effiziente interaktive Turing-Maschinen vorausgesetzt wurden, sind solche Überlegungen sinnvoll, da sie zu einer Abstufung des Blindheitsgrades führen.

4.1 Definition der perfekten Blindheit

Intuitiv erwartet man von einem klassischen perfekt blinden Signaturschema, dass ein Signierer aus der Kenntnis seiner im Signaturvorgang gespeicherten Informationen keinerlei Nutzen ziehen kann: Auch wenn der Signierer durch äußere Umstände eine Vermutung hat, welcher Signaturvorgang zu einer gegebenen Signatur geführt haben könnte, gibt es in seinen Protokollmitschriften keinerlei Anhaltspunkte, durch die er seine Vermutung untermauern kann. Er kann also die zur Signatur passende Protokollmitschrift nur raten. Mit anderen Worten: Man erwartet von einer perfekt blinden Signatur, dass die Protokollmitschrift des Signierers und das Nachrichten-Signatur-Paar stochastisch unabhängig sind (vgl. Kapitel 2).

Geht man vom Konzept der interaktiven Signatur aus, so kommt man zu einem allgemeineren Ansatz: Übernimmt der Signierer die Rolle des Verifizierers oder gibt der Verifizierer seine Protokollmitschrift an den Signierer weiter, so erhält der Signierer gewisse Informationen über die ausgestellte Signatur. Von einem perfekt blinden interaktiven Signaturschema erwartet man, dass der Signierer keinerlei Informationen über die Signatur aus der Protokollansicht des Verifizierers gewinnen kann, also die Protokollansicht des Signierers und die Protokollansicht des Verifizierers stochastisch unabhängig sind.

Bemerkung 4.2 *Bei den vorangegangenen Überlegungen ist zu beachten, dass sowohl das Signatur- als auch das Verifikationsprotokoll bezüglich eines* **festen** *Schlüsselpaars* $(\mathsf{pk}, \mathsf{sk})$ *durchgeführt werden. Das heißt, alle Überlegungen müssen* **unter der Bedingung** *durchgeführt werden, dass schon ein Schlüsselpaar* $(\mathsf{pk}, \mathsf{sk})$ *gewählt wurde.*

In Abschnitt 1.2 wurde der Begriff der gegenseitigen Information zweier Zufallsvektoren X und Y als ein Maß der stochastischen Abhängigkeit von X und Y vorgestellt, wobei X und Y genau dann stochastisch unabhängig sind, wenn $\mathcal{I}[X, Y] = 0$ gilt (vgl. Satz 1.3). Wie in der obigen Bemerkung erwähnt, muss man hier die

Information **unter der Bedingung, dass das Schlüsselpaar** $(\mathsf{pk}, \mathsf{sk})$ **gewählt wurde**, betrachten, d.h. das verwendete Wahrscheinlichkeitsmaß ist die bedingte Wahrscheinlichkeit bezüglich der Bedingung, dass sich das Schlüsselpaar in $(\mathsf{pk}, \mathsf{sk})$ realisiert hat. Da dies alle folgenden Überlegungen betrifft, bezeichnen wir der besseren Lesbarkeit halber im Weiteren die Information **unter der Bedingung, dass der Schlüssel** $(\mathsf{pk}, \mathsf{sk})$ **gewählt wurde**, mit $\mathcal{I}$. Dementsprechend definieren wir die perfekte Blindheit einer interaktiven Signatur wie folgt.

Definition 4.1 *Es sei ein interaktives Signaturschema $\Sigma_I = (G, \sigma_I, V_I)$ mit den Teilnehmern $\mathcal{S}$, $\mathcal{R}$ und $\mathcal{V}$ gemäß Definition 3.1 gegeben. Dann heißt Σ_I* **perfekt blind**, *wenn gilt*

$$\mathcal{I}[view_{\mathcal{S}}(\sigma_I), view_{\mathcal{V}}(V_I)] = 0.$$

Zur Untersuchung der Blindheitseigenschaften beschränken wir uns auf zwei natürliche Varianten:

1. Die Signatur wird im Verifikationsprotokoll im „Klartext“an den Verifizierer übermittelt.

2. Die Signatur wird im Signaturprotokoll erstellt, indem der Empfänger dem Signierer die Nachicht übermittelt, der Signierer seinen Signaturalgorithmus durchführt und die Signatur an den Empfänger zurückschickt.

In diesen Fällen hat die interaktive Signatur eine nützliche Eigenschaft:

Satz 4.1 *Sei $\Sigma_I = (G, \sigma_I, V_I)$ ein interaktives Signaturschema und $\mathsf{key} \in G(1^k)$ ein von dem Signierer gewählter Schlüssel. Ist*

(a) *σ_I gemäß Abbildung 3.2 gegeben, oder*

(b) *ist V_I gemäß Abbildung 3.3 gegeben,*

so bilden die Zufallsvariablen

$$(view_{\mathcal{S}}(\sigma_I), (m, \sigma(m)), view_{\mathcal{V}}(V_I))$$

eine Markovkette.

Beweis. Es seien $U := view_{\mathcal{S}}(\sigma_I)$, $S := (m, \sigma(m))$ und $V := view_{\mathcal{V}}(V_I)$ die im Satz beschriebenen Zufallsvariablen und u, s und v mögliche Realisierungen. Wir stellen fest, dass für einen gegebene Schlüssel key die Verteilung von $view_{\mathcal{S}}(\sigma_I)$ unter

Voraussetzung (a) vollständig durch die Signatur bestimmt ist, sodass U die gleiche Verteilung wie S besitzt. Damit gilt für alle geeigneten u, s und v:

$$\begin{aligned}\mathcal{P}(U=u,S=s,V=v)\mathcal{P}(S=s) &= \mathcal{P}(S=u,S=s,V=v)\mathcal{P}(S=s)\\ &= \begin{cases}\mathcal{P}(S=u,V=v)\mathcal{P}(S=s), & \text{falls } u=s\\ 0, & \text{sonst.}\end{cases}\end{aligned}$$

Andererseits gilt

$$\begin{aligned}\mathcal{P}(S=s,V=v)\mathcal{P}(U=u,S=s) &= \mathcal{P}(S=s,V=v)\mathcal{P}(S=u,S=s)\\ &= \begin{cases}\mathcal{P}(S=u,V=v)\mathcal{P}(S=s), & \text{falls } s=u\\ 0, & \text{sonst.}\end{cases}\end{aligned}$$

Damit haben wir gezeigt, dass

$$\mathcal{P}(U=u,S=s,V=v)\mathcal{P}(S=s)=\mathcal{P}(S=s,V=v)\mathcal{P}(U=u,S=s)$$

gilt und durch Umstellen erhält man die Markoveigenschaft.

Unter der Voraussetzung (b) zeigt man die Behauptung analog. □

Man beachte, dass die Möglichkeiten für einen betrügerischen Signierer $\mathcal{S}$ und einen betrügerischen Verifizierer $\mathcal{V}$ hier stark eingeschränkt sind. Selbst bei einem Zusammenschluss beider Parteien können diese nicht von der Interaktivität der Signatur profitieren: einerseits, da der Verifizierer unter Bedingung (a) keinerlei Handlungsspielraum hat und andererseits, da der Signierer, der im Signaturprotokoll prizipiell Handlungsspielraum besitzt, unter Bedingung (b) noch keinen Zugriff auf die Informationen des Verifizieres bezüglich der aktuellen Nachricht hat. An dieser Überlegung kann man sehen, dass die zufällige Wahl der Nachricht essentiell für den Begriff der perfekten Blindheit ist.

4.2 Perfekte Blindheit des Signaturprotokolls

Wir betrachten das Pendant zu den in Kapitel 2 vorgestellten blinden Signaturen, also interaktive Signaturen, bei denen die Blendung während des Signaturprotokolls erfolgt: Der Signierer erhält im Signaturprotokoll keine Informationen über die Gestalt der eigentlichen Signatur. Wie oben erwähnt, beschreibt man hier die Blindheit durch die Unabhängigkeit der Protokollansicht des Signierers im Signaturprotokoll und des Nachrichten-Signatur-Paares $(m, \sigma_I(m))$.

Definition 4.2 *Es sei ein interaktives Signaturschema $\Sigma_I = (G, \sigma_I, V_I)$ mit den Teilnehmern $\mathcal{S}$, $\mathcal{R}$ und $\mathcal{V}$ gemäß Definition 3.1 gegeben. Dann heißt das Signaturprotokoll σ_I* **perfekt blind**, *wenn gilt*

$$\mathcal{I}[view_{\mathcal{S}}(\sigma_I), (m, \sigma_I(m))] = 0.$$

Man beachte, dass dies die gängige Definition von Blindheit in digitalen Signaturschemata ist (vgl. zum Beispiel [CPS94]). Der nächste Satz zeigt, dass jede interaktive Signatur mit perfekt blindem Signaturprotokoll schon eine perfekt blinde interaktive Signatur darstellt. Mit anderen Worten: Das Konzept der perfekt blinden interaktiven Signaturen ist geeignet, um blinde Signaturen zu beschreiben. Damit sind die in Kapitel 2 vorgestellten Signaturen Beispiele für perfekt blinde interaktive Signaturen. Hier stellt sich die Frage nach Mechanismen, die zu einem perfekt blinden Signaturprotokoll führen. Diese Fragestellung wird in Kapitel 6 für blinde interaktive RSA-Signaturen und blinde interaktive Schnorr-Signaturen bearbeitet.

In Definition 4.2 wird erklärt, was wir unter perfekter Blindheit eines *Signaturprotokolls* verstehen wollen. Definition 4.1 bezieht sich hingegen auf die perfekte Blindheit der kompletten interaktiven Signatur. Damit liegen zunächst zwei unabhängige Begrifflichkeiten vor, deren Zusammenhang im Weiteren beleuchtet wird.

Satz 4.2 *Es sei $\Sigma_I = (G, \sigma_I, V_I)$ ein interaktives Signaturschema, das Voraussetzung (a) aus Satz 4.1 erfüllt. Ist das Signaturprotokoll σ_I perfekt blind, so ist das interaktive Signaturschema Σ_I perfekt blind.*

Beweis. Ist σ_I perfekt blind, so gilt

$$\mathcal{I}[view_{\mathcal{S}}(\sigma_I), (m, \sigma_I(m))] = 0.$$

Nach Satz 4.1 besitzt

$$(view_{\mathcal{S}}(\sigma_I), (m, \sigma_I(m)), view_{\mathcal{V}}(V_I))$$

die Markoveigenschaft, und damit können wir Satz 1.4 anwenden. Somit folgt

$$0 \leq \mathcal{I}[view_{\mathcal{S}}(\sigma_I), view_{\mathcal{V}}(V_I)] \leq \mathcal{I}[view_{\mathcal{S}}(\sigma_I), (m, \sigma_I(m))] = 0.$$

Somit ist $\mathcal{I}[view_{\mathcal{S}}(\sigma_I), view_{\mathcal{V}}(V_I)] = 0$, und es folgt die perfekte Blindheit von Σ_I. □

Damit können wir die Frage, ob jede perfekt blinde Signatur auch gleichzeitig eine perfekt blinde interaktive Signatur ist, positiv beantworten. Unklar ist bislang, ob es grundsätzlich nur diesen Typ von perfekt blinden interaktiven Signaturen geben kann. Dieser Frage gehen wir im nächsten Abschnitt nach.

4.3 Perfekte Blindheit des Verifikationsprotokolls

Analog zu perfekt blinden Signaturprotokollen verlangt man von einem perfekt blinden Verifikationsprotokoll anschaulich, dass in der Protokolldurchführung keinerlei Informationen über das Nachrichten-Signatur-Paar $(m, \sigma_I(m))$ preisgegeben werden. Wir betrachten die Situation genauer: Angenommen, ein Empfänger $\mathcal{R}$ hat mittels des Signaturprotokolls σ_I eine Signatur $\sigma_I(m)$ von dem Signierer $\mathcal{S}$ auf die Nachricht m erhalten. Dann besitzt der Signierer die Protokollmitschrift des Protokolls σ_I mit $\mathcal{R}$, die er speichern kann. Da der Signierer die Blindheit angreifen möchte, müssen wir davon ausgehen, dass er alle ihm zugänglichen Informationen sammelt. Also hat der Signierer seine Protokollmitschrift in seiner Datenbank. Im günstigsten Fall für den Signierer enthält diese Protokollmitschrift das Nachrichten-Signatur-Paar $(m, \sigma_I(m))$. Möchte der Empfänger $\mathcal{R}$ seine Signatur einsetzen, so führt er das Verifikationsprotokoll mit einem Verifizierer $\mathcal{V}$ aus. Man beachte, dass ggf. auch $\mathcal{S}$ selbst dieser Verifizierer sein kann. Anderenfalls erhält der Signierer die Protokollmitschrift von $\mathcal{V}$. Damit ist der Signierer einerseits im Besitz seiner eigenen Protokollmitschrift des Signaturprotokolls und damit ggf. im Besitz von $(m, \sigma_I(m))$ und andererseits im Besitz der Protokollmitschrift des Verifikationsprotokolls. In dieser Situation verlangen wir, dass der Signierer nicht in der Lage sein soll, zu beurteilen, ob seine Angaben zusammenpassen. Dementsprechend halten wir fest:

Definition 4.3 *Es sei ein interaktives Signaturschema $\Sigma_I = (G, \sigma_I, V_I)$ mit den Teilnehmern $\mathcal{S}$, $\mathcal{R}$ und $\mathcal{V}$ gemäß Definition 3.1 gegeben. Dann heißt das Verifikationsprotokoll V_I* **perfekt blind**, *wenn gilt*

$$\mathcal{I}[(m, \sigma_I(m))), view_{\mathcal{V}}(V_I)] = 0.$$

Wie im letzem Abschnitt erklärt diese Definition den Begriff der perfekten Blindheit im *Verifikationsprotokoll*. Durch den folgenden Satz wird der Zusammenhang zu perfekt blinden *interaktiven Signaturen* beleuchtet.

Satz 4.3 *Es sei $\Sigma_I = (G, \sigma_I, V_I)$ ein interaktives Signaturschema, das Bedingung (b) aus Satz 4.1 erfüllt. Ist das Verifikationsprotokoll V_I perfekt blind, so ist die interaktive Signatur Σ_I perfekt blind.*

Beweis. Der Beweis kann analog zum Beweis von Satz 4.2 geführt werden. □

Der Signierer ist also im Falle eines perfekt blinden Verifikationsprotokolls nicht in der Lage, seine eigene Protokollmitschrift des Signaturprotokolls der Protokollmitschrift eines Verifizierers zuzuordnen. Damit lassen sich perfekt blinde interaktive

Signaturen gemäß Satz 4.1 grundsätzlich sowohl durch perfekt blinde Verifikationsprotokolle als auch durch perfekt blinde Signaturprotokolle realisieren. Mit anderen Worten: Betrachtet man nicht das übliche Konzept der perfekt blinden Signaturen, sondern erweitert dieses zu dem Konzept der perfekt blinden interaktiven Signaturen, hat man eine zusätzliche Möglichkeit, die perfekte Blindheit zu erreichen, nämlich indem man das Verifikationsprotokoll blendet.

Hier schließt sich einerseits die Frage nach der Existenz von perfekt blinden Verifikationsprotokollen und andererseits die Frage nach Bedingungen für die Blindheit eines Verifikationsprotokolls an. Diesen Fragen werden wir in Kapitel 7 nachgehen.

4.4 Fazit

Nachdem im letzten Kapitel das Konzept der interaktiven Signaturen eingeführt wurde, konnten wir in diesem Kapitel zeigen, dass perfekt blinde interaktive Signaturen eine Verallgemeinerung von perfekt blinden Signaturen darstellen. Wir konnten zwei hinreichende Bedingungen für die perfekte Blindheit einer interaktiven Signatur angeben: einerseits die perfekte Blindheit des Signaturprotokolls und andererseits die perfekte Blindheit des Verifikationsprotokolls. Damit können die eingangs gestellten Fragen beantwortet werden:

1. Jede perfekt blinde Signatur ist auch eine perfekt blinde interaktive Signatur.
2. Blindheit kann (unter gewissen Voraussetzungen) auch im Verifikationsprotokoll erreicht werden.

An diese Ergebnisse schließt sich die Frage nach geeigneten kryptographischen Bausteinen für die Konstruktion von perfekt blinden Signatur- bzw. Verifikationsprotokollen sowie die Frage nach der Existenz von perfekt blinden Verifikationsprotokollen an. Diese Fragestellungen werden in Kapitel 6 und 7 bearbeitet.

Kapitel 5

Rechnerisch blinde interaktive Signaturen

Die perfekte Blindheit, d.h. die stochastische Unabhängigkeit der Protokollansichten von Signaturen, wird im Allgemeinen durch die Verwendung von Zufallszahlen erzeugt, die an einer späteren Stelle im Protokollverlauf wieder entfernt werden. Das Protokolldesign dieser Protokolle setzt für die perfekte Blindheit perfekte Zufallszahlengeneratoren voraus, die in der Praxis schwer realisierbar sind. Daraus ergibt sich die Notwendigkeit eines schwächeren Blindheitsbegriffs, der diesem Umstand Rechnung trägt. Man beachte, dass diese Fragestellung erhebliche Parallelen zu der Problematik aufweist, praxisnahe Verschlüsselungsmethoden zu entwerfen, die zwar ein ausreichendes Sicherheitsniveau besitzen, jedoch keine perfekte Sicherheit aufweisen.

Wie bereits erwähnt wurde, gibt es einen solchen Blindheitsbegriff für blinde Signaturen (vgl. Kapitel 2). Im letzten Kapitel haben wir gesehen, dass sich der perfekte Blindheitsbegriff auf interaktive Signaturen übertragen lässt. Nun werden wir in diesem Kapitel sehen, dass dies für den rechnerischen Blindheitsbegriff ebenfalls gelingt. Auch hier stellt sich die Frage nach Eigenschaften der einzelnen Protokolle in einer rechnerisch blinden Signatur. Es zeigt sich wie in Kapitel 4, dass sich rechnerisch blinde Signaturen als rechnerisch blinde interaktive Signaturen auffassen lassen, und dass die rechnerische Blindheit nicht über das Signaturprotokoll erreicht werden muss.

Es sei noch einmal daran erinnert, dass wir in allen folgenden Überlegungen von einem Empfänger $\mathcal{R}$ ausgehen, der sich korrekt verhält. In diesem Kapitel wird, anders als im vorangegangenen, der Angreifer als rechnerisch beschränkt angenommen.

5.1 Definition der rechnerischen Blindheit

Wie in Abschnitt 3.2 ist unser Ziel, den stärksten Angriff und den geringsten Erfolg zu identifizieren. Da die Blindheit die Sicherheitsinteressen des Empfängers einer Signatur gegenüber einer Koalition eines betrügenden Signierers und eines betrügenden Verifizierers darstellt, übernimmt der Angreifer in den folgenden Überlegungen die Rolle des Signierers bzw. des Verifizierers. Das Ziel eines Angriffs ist dabei, eine korrekte Zuordnung der Protokollmitschriften des Signierers im Signaturprotokoll und der Protokollmitschriften eines Verifizierers im Verifikationsprotokoll zu finden. Wir sehen eine interaktive Signatur als rechnerisch blind an, wenn ein effizienter Angreifer nach dem stärksten Angriff nur mit vernachlässigbarer Wahrscheinlichkeit einen solchen Erfolg erzielen kann.

Grundsätzlich sind wie in Abschnitt 3.2 zwei Angriffe möglich: Der sequentielle und der parallele Angriff. Wir werden uns in der Definition der rechnerischen Blindheit zunächst auf den stärkeren der beiden, den parallelen Angriff konzentrieren, wobei trotzdem in der Angriffsphase auch ein sequentieller Angriff möglich ist.

Wie oben beschrieben, ist das Ziel des Angreifers $\mathcal{A}$, für zwei Protokollmitschriften des Signierers im Signaturprotokoll und zwei Protokollmitschriften des Verifiziers im Verifikationsprotokoll herauszufinden, welche zusammengehören, also zu der gleichen Signatur passen. Damit muss sich der Angreifer zunächst in der Rolle des Signierers und dann in der Rolle des Verifizierers am Spiel beteiligen, um die relevanten Protokollmitschriften zu erhalten. Es ist klar, dass der Angriff nicht parallel auf beide Protokollteile durchgeführt werden kann: In diesem Fall kann der Angreifer seine Aufgabe auf jeden Fall erfüllen. Dies wird man in dem im Folgenden vorgestellten Spiel 5.1 wiederfinden.

Nun ist noch die Frage zu stellen, wieviel Handlungsspielraum dem Angreifer $\mathcal{A}$ im Spiel eingeräumt werden sollen. Hier gibt es im wesentlichen zwei verschiedene Möglichkeiten:

- Der Angreifer hat keine Möglichkeit, auf die Wahl der Nachrichten einzuwirken.
- Der Angreifer kann die Nachrichten bestimmen, die er signieren möchte.

Offensichtlich führt letzteres zum stärkeren Blindheitsbegriff, sodass wir dem Angreifer diese Möglichkeit in der Tat einräumen.

Wir definieren also die rechnerische Blindheit eines interaktiven Signaturschemas $\Sigma_I = (G, \sigma_I, V_I)$, indem wir die Erfolgswahrscheinlichkeiten im folgenden Spiel eines

Angreifers $\mathcal{A}$ mit Zugriff auf ein Orakel $\mathcal{O}$ gegen einen Empfänger $\mathcal{R}$ betrachten. Das Ziel von $\mathcal{A}$ in diesem Spiel ist das Folgende: Mithilfe des Orakels generiert er zu zwei selbstgewählten Nachrichten zwei Protokollmitschriften des Signaturprotokolls und zwei Protokollmitschriften des Verifikationsprotokolls. Anschließend versucht er herauszufinden, welche der Mitschriften in dem Sinne zusammengehören, dass sie durch das gleiche Nachrichten-Signatur-Paar erzeugt wurden. Dies wird in Spiel 5.1 so umgesetzt, dass das Orakel ein Bit b erhält, das im Prinzip beschreibt, ob die Empfänger die Rollen tauschen oder nicht. Da der Angreifer die Nachrichten selbst generiert, muss allerdings dafür gesorgt werden, dass er die einzelnen Protokollmitschriften nicht den Nachrichten zuordnen kann. Zu diesem Zweck wird in Spiel 5.1 ein Hilfsbit c gewählt. Dieses ist für den Angreifer im Spiel jedoch nicht von Bedeutung: Er ist einzig und alleine damit befasst, das Bit b zu bestimmen. Das Spiel ist schematisch in Abbildung 5.1 dargestellt.

Man beachte, dass der Empfänger $\mathcal{R}$, gegen den der Angreifer in dem beschriebenen Spiel antritt, ein abstrakter Empfänger ist, der dafür zuständig ist, einerseits das zu ratende Bit b zu wählen und andererseits die Zusatzeingaben für die konkreten, im Protokoll von $\mathcal{O}$ vertretenen Empfänger $\mathcal{R}_0$ und $\mathcal{R}_1$ zu wählen. Damit kann man sich $\mathcal{R}$ als Partei vorstellen, der die Empfängerinteressen vertritt, jedoch als solcher nicht in Erscheinung tritt.

Wie in Kapitel 3 bereits erwähnt wurde, sind einem Empfänger $\mathcal{R}$ grundsätzlich zusätzliche Eingaben par erlaubt. Da diese für die Blindheit von Nutzen sein können, werden sie im folgenden Spiel mit aufgeführt. Allerdings gehen wir im Weiteren stets davon aus, dass $\mathcal{R}$ die Eingabe par nur in einem der beiden Protokolle σ_I und V_I verwendet. Dies ist mit der Notation $\overline{\mathsf{par}}$ angedeutet: Hat $\mathcal{R}$ in σ_I die Zusatzeingabe par, so hat er in V_I keine Eingabe, d.h. $\overline{\mathsf{par}}$ ist in diesem Fall nicht mit einem Wert belegt. Hat umgekehrt $\mathcal{R}$ in σ_I keine Zusatzeingabe, d.h. ist par nicht mit einem Wert belegt, so ist die Zusatzeingabe von $\mathcal{R}$ in V_I mit $\overline{\mathsf{par}}$ gekennzeichnet.

Spiel 5.1

1. *Der Empfänger $\mathcal{R}$ wählt ein Hilfsbit $c \in_R \{0,1\}$ und ein Bit $b \in_R \{0,1\}$. Ferner berechnet er $\hat{c} := b + c \bmod 2$ und bestimmt ggf. seine Zusatzeingaben par_0 und par_1. Das Orakel $\mathcal{O}$ erhält als Eingabe $c, \hat{c}$ und das Bit b sowie par_0 und par_1.*

2. *Der Angreifer $\mathcal{A}$ erzeugt durch Eingabe des Sicherheitsparameters 1^k in den Schlüsselerzeugungsalgorithmus G ein Schlüsselpaar:*

$$(\mathsf{pk}, \mathsf{sk}) \in G(1^k).$$

3. *$\mathcal{A}$ erzeugt zwei Nachrichten m_0 und m_1, die insbesondere von* pk *oder* sk *abhängen können.*

4. *$\mathcal{A}$ sendet (m_0, m_1) und* pk *an das Orakel.*

5. *$\mathcal{A}$ führt parallel und nicht zwingend unabhängig voneinander das interaktive Protokoll σ_I mit dem Orakel $\mathcal{O}$ aus. Dabei übernimmt das Orakel die Rolle von zwei ehrlichen Empfängern $\mathcal{R}_c$ und $\mathcal{R}_{1-c}$. Die Eingabe von $\mathcal{A}$ in beiden Protokollen ist $(1^k, \mathsf{pk}, \mathsf{sk}, m_0, m_1)$, die Eingabe von $\mathcal{O}$ in der Rolle von $\mathcal{R}_c$ ist $(\mathsf{pk}, \mathsf{par}_c, m_c)$, die von $\mathcal{O}$ in der Rolle von $\mathcal{R}_{1-c}$ ist $(\mathsf{pk}, \mathsf{par}_{1-c}, m_{1-c})$.*

6. *Erhält das Orakel $\mathcal{O}$ als Ausgabe Signaturen $\sigma_I(m_0)$ und $\sigma_I(m_1)$ auf m_0 und m_1, so schickt $\mathcal{O}$* okay *an den Angreifer. Im Folgenden sei $s_i = (m_i, \sigma_I(m_i))$ für $i = 0, 1$.*

7. *$\mathcal{A}$ führt parallel und nicht zwingend unabhängig voneinander das interaktive Protokoll V_I mit dem Orakel $\mathcal{O}$ aus. Dabei übernimmt das Orakel die Rolle von zwei ehrlichen Empfängern $\mathcal{R}_{\hat{c}}$ und $\mathcal{R}_{1-\hat{c}}$. Die Eingabe von $\mathcal{A}$ in beiden Protokollen sind $(1^k, \mathsf{pk}, \mathsf{sk})$ und die Protokollmitschriften aus den Signaturprotokollen. Die Eingabe von $\mathcal{O}$ in der Rolle von $\mathcal{R}_{\hat{c}}$ ist $(\mathsf{pk}, \overline{\mathsf{par}}_{\hat{c}}, s_{\hat{c}})$, die von $\mathcal{O}$ in der Rolle von $\mathcal{R}_{1-\hat{c}}$ ist $(\mathsf{pk}, \overline{\mathsf{par}}_{1-\hat{c}}, s_{1-\hat{c}})$.*

8. *$\mathcal{A}$ gibt ein Bit b aus.*

9. *Der Angreifer $\mathcal{A}$ gewinnt das Spiel, wenn $b' = b$ ist.*

Das Spiel kann verschiedene Angriffphasen beinhalten, in denen der Angreifer die Möglichkeit besitzt, mit verschiedenen Empfängern sowohl das Signatur- als auch das Verifikationsprotokoll durchzuführen. Dazu hat der Angreifer Zugriff auf zwei weitere Orakel $\mathcal{O}_0$ und $\mathcal{O}_1$, die für den Empfänger $\mathcal{R}_0$ mit Zusatzeingabe par_0 bzw. für $\mathcal{R}_1$ mit Zusatzeingabe par_1 stehen. Er ist somit innerhalb dieser Phasen in der Lage, Protokollmitschriften und Nachrichten-Signatur-Paare zu erzeugen, von denen er weiß, dass sie zusammengehören. Eine solche Phase ist vor der Wahl der Nachrichten m_0 und m_1 erlaubt, aber auch nach der Beendigung von Schritt 6 sowie nach Schritt 7. Die Protokolldurchführungen der Angriffsphase können dabei parallel und adaptiv zu beliebig von $\mathcal{A}$ gewählten Nachrichten durchgeführt werden. Als einzige Einschränkung machen wir hier, dass die Anzahl der Empfänger, mit denen $\mathcal{A}$ in den Angriffsphasen kommuniziert, polynomiell in k ist. Zusammenfassend sind also $\mathcal{A}$ parallel zu der Interaktion mit $\mathcal{O}$ Kommunikationsphasen mit verschiedenen Empfängern erlaubt, solange die Anzahl der Empfänger polynomiell in k bleibt. Man beachte, dass die beiden beschriebenen Orakel $\mathcal{O}_0$ und $\mathcal{O}_1$ zusammenfallen, wenn

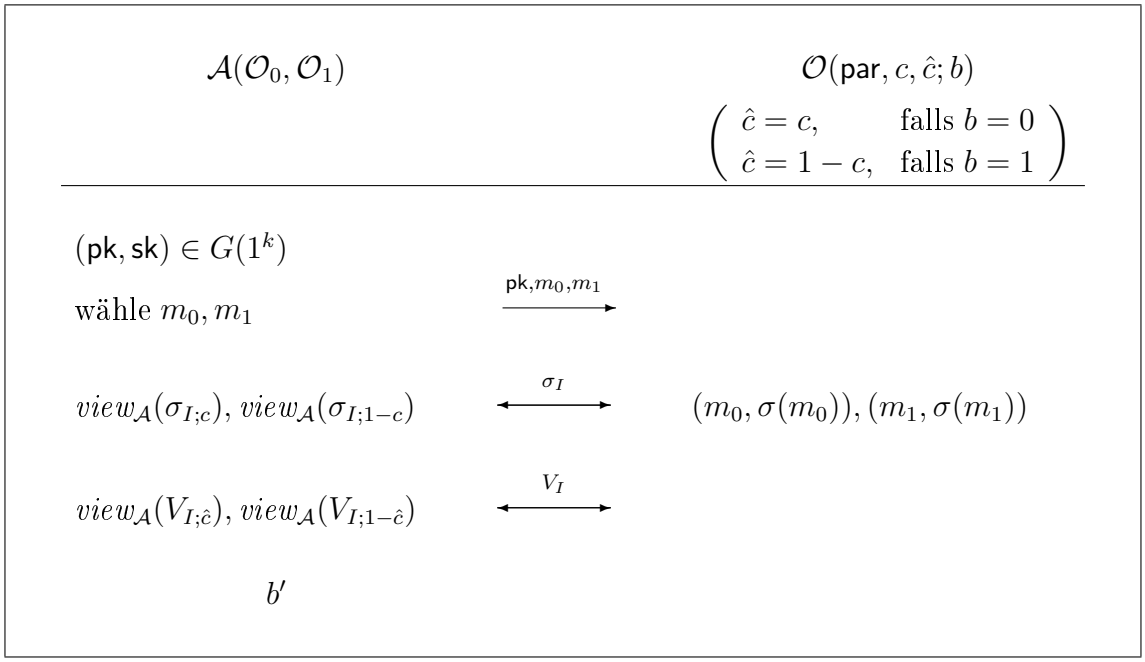


Abbildung 5.1: Spiel 5.1

die interaktive Signatur keine Zusatzeingaben des Empfängers vorsieht.

Man beachte, dass der Angreifer das Spiel mit Wahrscheinlichkeit 1/2 gewinnt, wenn er die gewonnenen Informationen ignoriert und rät. Man muss hier also fordern, dass ihm die Information, die er gewinnt, nur einen vernachlässigbaren Vorteil gegenüber dem Raten gewährt.

Definition 5.1 *Es sei $k \in \mathbb{N}$ ein Sicherheitsparameter. Ein interaktives Signaturschema $\Sigma_I = (G, \sigma_I, V_I)$ heißt* **rechnerisch blind** *(im Sicherheitsparameter k), falls es für jeden effizienten Angreifer $\mathcal{A}$ mit Zugriff auf zwei wie oben beschriebene Orakel $\mathcal{O}_0$ und $\mathcal{O}_1$ eine in k vernachlässigbare Funktion ν gibt, sodass für die Erfolgswahrscheinlichkeit $\varepsilon(k)$ von $\mathcal{A}$ in Spiel 5.1 gilt:*

$$1/2 - \varepsilon(k) = \nu(k).$$

Bemerkung 5.1 *Man beachte, dass es zu jedem effizienten Angreifer $\mathcal{A}$, dessen Verlustwahrscheinlichkeit $1 - \varepsilon(k)$ in Spiel 5.1 nicht vernachlässigbar von 1/2 abweicht (der also mit „großer“ Wahrscheinlichkeit das Spiel verliert), einen effizienten Angreifer $\mathcal{A}^*$ gibt, der Spiel 5.1 mit nicht vernachlässigbar von 1/2 abweichender Wahrscheinlichkeit gewinnt: $\mathcal{A}^*$ lässt $\mathcal{A}$ seinen Angriff durchführen und gibt dann statt b' das Bit $1 - b'$ aus. Aus diesem Grunde kann man statt der oben aufgeführten Bedingung an die Erfolgswahrscheinlichkeit auch fordern, dass es für jeden*

effizienten Angreifer eine in k vernachlässigbare Funktion ν gibt, sodass

$$\varepsilon(k) - 1/2 = \nu(k)$$

gilt. Diese Bemerkung gilt allgemein für Spiele des hier aufgeführten Typs, also insbesondere für Spiel 5.2 und Spiel 5.3.

Wie im letzten Kapitel stellt sich die Frage, nach einem Modell für eine Blendung während des Signatur- bzw. während des Verifikationsprotokolls. Diese werden wir in den folgenden Abschnitten beleuchten.

5.2 Rechnerische Blindheit des Signaturprotokolls

Die folgende Definition wurde analog zu der im letzten Abschnitt entwickelt: Der stärkste Angriff ist ein paralleler adaptiver Angriff mit gewählten Nachrichten, der Erfolg des Angreifers besteht darin, zwei Protokollmitschriften des Signierers im Signaturprotokoll und zwei Nachrichten-Signatur-Paare zuzuordnen. Dementsprechend definieren wir das folgende Spiel, das der Angreifer $\mathcal{A}$ mit Zugriff auf ein Orakel $\mathcal{O}$ gegen einen Empfänger $\mathcal{R}$, der wie oben als Interessensvertreter aller Empfänger zu sehen ist, gewinnen möchte.

Spiel 5.2

1. *Der Empfänger $\mathcal{R}$ wählt ein Bit $b \in_R \{0,1\}$ und ggf. die Zusatzeingaben par_0 und par_1. Das Orakel $\mathcal{O}$ erhält als Eingabe das Bit b sowie par_0 und par_1.*

2. *Der Angreifer $\mathcal{A}$ erzeugt durch Eingabe des Sicherheitsparameters 1^k in den Schlüsselerzeugungsalgorithmus G ein Schlüsselpaar:*

$$(\mathsf{pk}, \mathsf{sk}) \in G(1^k).$$

3. *$\mathcal{A}$ erzeugt zwei Nachrichten m_0 und m_1, die insbesondere von pk oder sk abhängen können.*

4. *$\mathcal{A}$ sendet (m_0, m_1) und pk an das Orakel.*

5. *$\mathcal{A}$ führt parallel und nicht zwingend unabhängig voneinander das interaktive Protokoll σ_I mit dem Orakel $\mathcal{O}$ aus. Dabei übernimmt das Orakel $\mathcal{O}$ die Rolle von zwei ehrlichen Empfängern $\mathcal{R}_b$ und $\mathcal{R}_{1-b}$. Die Eingabe von $\mathcal{A}$ in beiden Protokollen ist $(1^k, \mathsf{pk}, \mathsf{sk}, m_0, m_1)$, die Eingabe von $\mathcal{O}$ in der Rolle von $\mathcal{R}_b$ ist $(\mathsf{pk}, \mathsf{par}_b, m_b)$, die von $\mathcal{O}$ in der Rolle von $\mathcal{R}_{1-b}$ ist $(\mathsf{pk}, \mathsf{par}_{1-b}, m_{1-b})$.*

6. *Gibt $\mathcal{O}$ in beiden Protokolldurchführungen gültige Signaturen aus, so erhält der Angreifer von $\mathcal{O}$ die Nachrichten-Signatur-Paare*

$$(m_0, \sigma_I(m_0)) \text{ und } (m_1, \sigma_I(m_1)).$$

7. *$\mathcal{A}$ gibt ein Bit b' aus.*

8. *Der Angreifer $\mathcal{A}$ gewinnt das Spiel, wenn $b = b'$ gilt.*

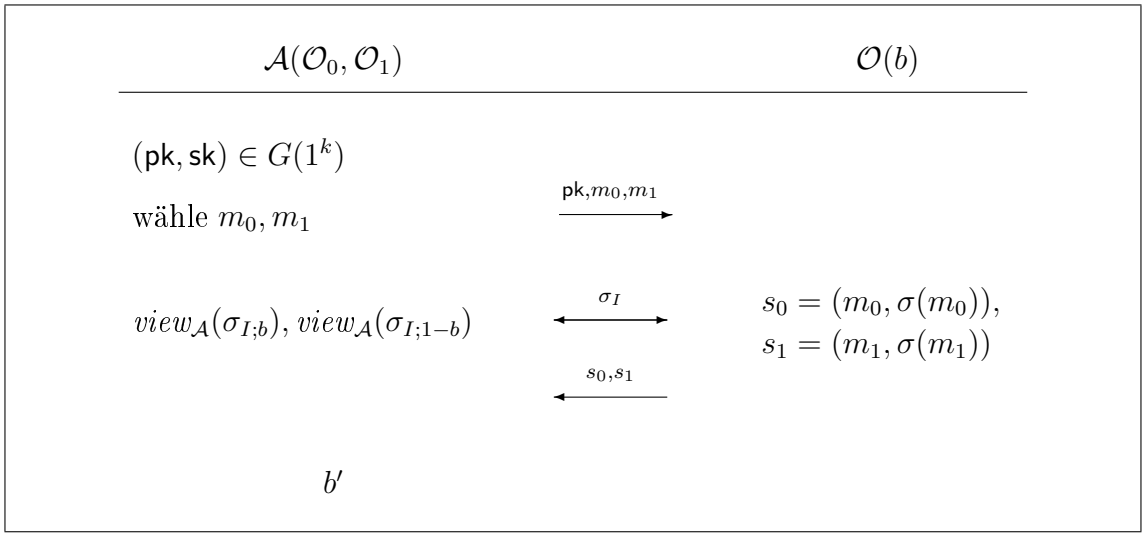


Abbildung 5.2: Spiel 5.2

Man beachte, dass der Angreifer die Signatur in Schritt 6 ggf. schon früher berechnen kann. Da es nicht sein Ziel ist, eine gültige Signatur zu produzieren, sondern seine Protokollmitschriften den Nachrichten-Signatur-Paaren zuzuordnen, muss der Angreifer lediglich irgendwann während seines Angriffs in den Besitz der Signaturen zu m_0 und m_1 kommen. Um dies sicherzustellen, übermittelt das Orakel $\mathcal{O}$ dem Angreifer die Signaturen $\sigma_I(m_0)$ und $\sigma_I(m_1)$.

Wie oben erlauben wir dem Angreifer an bestimmten Punkten des Spiels adaptive, parallele und voneinander abhängige Durchführungen des Protokolls σ_I mit zwei weiteren Orakeln $\mathcal{O}_0$ und $\mathcal{O}_1$, wobei wie oben $\mathcal{O}_0$ für den Empfänger mit Zusatzeingabe par_0 und $\mathcal{O}_1$ für den Empfänger mit Zusatzeingabe par_1 steht. Dabei darf die Anzahl der Interaktionen höchstens polynomiell in k sein. Im folgenden Spiel sind solche Interaktionen vor der Wahl der Nachrichten m_0 und m_1 sowie nach dem Erhalt der beiden Nachrichten-Signatur-Paare möglich. Wie oben wird in den Angriffsphasen ein einziges Orakel verwendet, wenn in σ_I keine Zusatzeingaben vorgesehen sind.

Definition 5.2 *Es sei $k \in \mathbb{N}$ ein Sicherheitsparameter und $\Sigma_I = (G, \sigma_I, V_I)$ eine interaktive Signatur. Das interaktive Signaturprotokoll σ_I von Σ_I heißt* **rechnerisch blind** *(im Sicherheitsparameter k), falls es für jeden effizienten Angreifer $\mathcal{A}$ mit Zugriff auf zwei wie oben beschriebene Orakel $\mathcal{O}_0$ und $\mathcal{O}_1$ eine in k vernachlässigbare Funktion ν gibt, sodass für die Erfolgswahrscheinlichkeit $\varepsilon(k)$ von $\mathcal{A}$ in Spiel 5.2 gilt:*

$$1/2 - \varepsilon(k) = \nu(k).$$

Bemerkung 5.2 *Die Blindheit des Signaturprotokolls entspricht im wesentlichen der Blindheit von Signaturen, wie sie in Kapitel 2 eingeführt wurde. Man beachte, dass dem Empfänger in dem oben vorgestellten Spiel Zusatzeingaben erlaubt sind.*

Vergleichbar zu der Situation bei der perfekten Blindheit folgt aus der rechnerischen Blindheit des Signaturprotokolls schon die rechnerische Blindheit der gesamten interaktiven Signatur. Wir erhalten also auch hier die Aussage, dass rechnerisch blinde Signaturen auch rechnerisch blinde interaktive Signaturen sind.

Satz 5.1 *Ist das Signaturprotokoll σ_I einer interaktiven Signatur $\Sigma_I = (G, \sigma_I, V_I)$ rechnerisch blind, so ist auch die interaktive Signatur Σ_I rechnerisch blind.*

Beweis. Sei $k \in \mathbb{N}$ ein Sicherheitsparameter, $\Sigma_I = (G, \sigma_I, V_I)$ ein interaktives Signaturschema und σ_I rechnerisch blind. Wir nehmen an, dass Σ_I nicht rechnerisch blind ist. Dann gibt es einen effizienten Angreifer, der Spiel 5.1 mit nicht vernachlässigbar von $1/2$ abweichender Wahrscheinlichkeit von $\varepsilon(k)$ gegen $\mathcal{R}$ gewinnt. Wir konstruieren einen effizienten Angreifer $\mathcal{A}^*$, der mit hinreichend großer Wahrscheinlichkeit Spiel 5.2 gewinnt (siehe auch Abbildung 5.3):

1. Der Empfänger $\mathcal{R}$ wählt ein Bit $b \in_R \{0, 1\}$ und ggf. seine Zusatzeingaben par_0 und par_1. Das Orakel $\mathcal{O}$ erhält als Eingabe b sowie par_0 und par_1.

2. $\mathcal{A}^*$ startet $\mathcal{A}$.

3. $\mathcal{A}$ generiert ein Schlüsselpaar $(\mathsf{pk}, \mathsf{sk}) \in G(1^k)$ und Nachrichten (m_0, m_1). Er sendet (m_0, m_1) und pk an $\mathcal{A}^*$.

4. $\mathcal{A}^*$ gibt (m_0, m_1) und pk an das Orakel und führt das Signaturprotokoll σ_I parallel und nicht zwingend unabhängig voneinander bzgl. $\mathcal{R}_b$ und $\mathcal{R}_{1-b}$ mit $\mathcal{O}$ durch. Dabei gibt er alle Werte an $\mathcal{A}$ weiter und verwendet dessen Ergebnisse als Antworten für das Orakel. Im letzten Schritt erhält $\mathcal{A}^*$ die Signaturen $\sigma_I(m_0)$ und $\sigma_I(m_1)$ von $\mathcal{O}$, die er nicht weitergibt.

5. Nun übernimmt $\mathcal{A}^*$ aus Sicht von $\mathcal{A}$ die Rolle des Orakels und führt den Schritt 7 aus Spiel 5.1 mit $\mathcal{A}$ durch. Dabei wählt er stets das Bit $\hat{c} = 0$.

6. $\mathcal{A}$ gibt ein Bit b' aus.

7. Die Ausgabe von $\mathcal{A}^*$ ist ebenfalls b'.

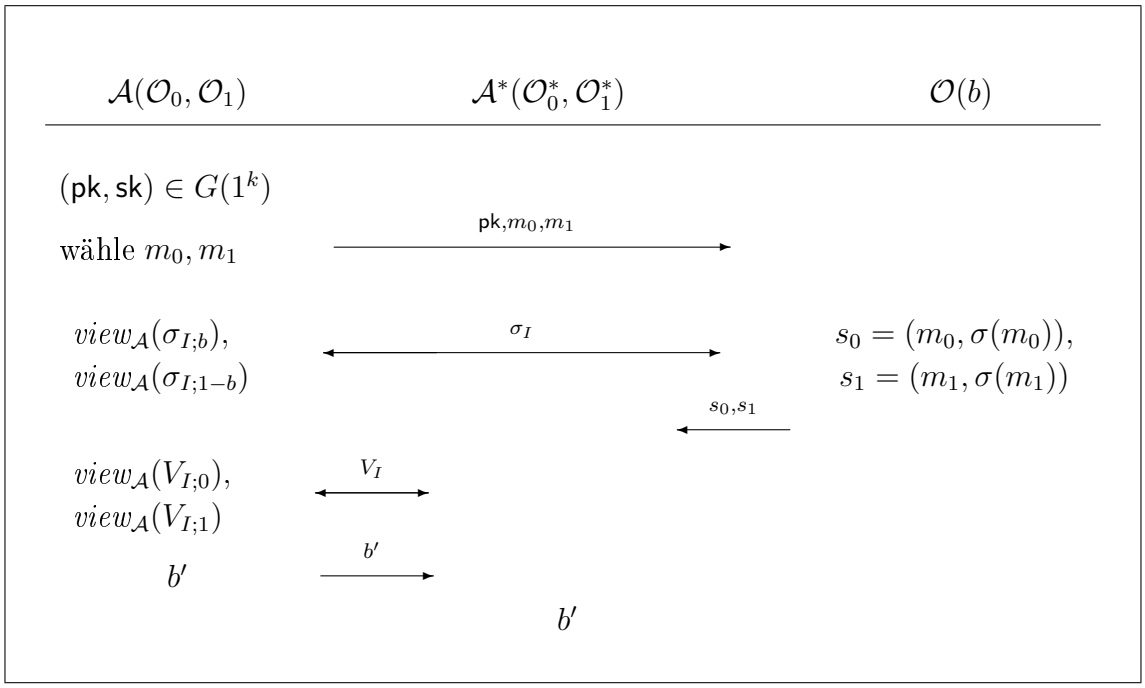


Abbildung 5.3: Beweis zu Satz 5.1

Man beachte, dass die Eingabe der Zusatzinformationen par_0 und par_1 kein Problem für die Durchführung des Verifikationsprotokolls darstellt: Verwendet das Orakel $\mathcal{O}$ die Werte par_0 und par_1 im Protokol σ_I, so benötigt $\mathcal{A}^*$ diese Information für die Protokolldurchführung von V_I nicht. Verwendet $\mathcal{O}$ keine Zusatzinformationen, so wählt sich $\mathcal{A}^*$ bereits bevor er $\mathcal{A}$ startet beliebige Zusatzinformationen $\overline{\mathsf{par}}_0$ und $\overline{\mathsf{par}}_1$, falls diese im Protokoll V_I gefordert werden.

Verlangt $\mathcal{A}$ in einer der möglichen Angriffsphasen Zugriff auf die Orakel $\mathcal{O}_0$ und $\mathcal{O}_1$, wie im Kontext von 5.1 beschrieben, so kann $\mathcal{A}^*$ die erforderlichen Informationen aus dem Zugriff auf seine eigenen Orakel $\mathcal{O}_0^*$ und $\mathcal{O}_1^*$ gemäß dem oben dargestellten Spiel erhalten.

Insgesamt ist $\mathcal{A}^*$ effizient, da $\mathcal{A}$ effizient ist. Ferner gelingt der Angriff von $\mathcal{A}^*$, wenn $\mathcal{A}$ das richtige Bit $b' = b$ ausgibt: Ist $b' = 0$, so weiß $\mathcal{A}^*$, dass die Protokollmitschriften nicht getauscht wurden, d.h. es gilt $b = 0$. Ist $b' = 1$, so weiß $\mathcal{A}^*$, dass die Protokollmitschriften getauscht wurden. In diesem Fall muss auch $b = 1$ gewesen

sein. $\mathcal{A}$ gelingt dies mit Wahrscheinlichkeit $\varepsilon(k)$, sodass $\mathcal{A}^*$ mit einer Wahrscheinlichkeit von mindestens $\varepsilon(k)$ Spiel 5.2 gewinnt.
Somit folgt die Behauptung. □

5.3 Rechnerische Blindheit des Verifikationsprotokolls

In Abschnitt 4.3 wurde die Erwartung, dass ein Angreifer (in der Rolle des Verifizierers) in einem blinden Verifikationsprotokoll Nachrichten-Signatur-Paare und seine eigenen Protokollansichten nicht zuordnen kann, motiviert. Mit dieser Überlegung formulieren wir die rechnerische Blindheit des Verifikationsprotokolls analog zu den letzten beiden Abschnitten: Der stärkste Angriff ist ein paralleler adaptiver Angriff mit gewählten Nachrichten-Signatur-Paaren, und der Erfolg des Angreifers besteht darin, zwei Protokollmitschriften des Verifizierers im Verifikationsprotokoll und zwei Nachrichten-Signatur-Paare zuzuordnen. Dementsprechend erzeugt der Angreifer $\mathcal{A}$ in dem folgenden Spiel, das er gegen ein Orakel $\mathcal{O}$ gewinnen möchte, zunächst zwei Nachrichten-Signatur-Paare. Danach führt er dann das Protokoll V_I mit dem Orakel $\mathcal{O}$ aus, sodass er zwei Protokollmitschriften des Verifizierers erhält.

Spiel 5.3

1. *$\mathcal{R}$ wählt ein Bit $b \in_R \{0,1\}$ und ggf. die Zusatzeingaben $\overline{\mathsf{par}}_0$ und $\overline{\mathsf{par}}_1$. Das Orakel $\mathcal{O}$ erhält als Eingabe b sowie $\overline{\mathsf{par}}_0$ und $\overline{\mathsf{par}}_1$.*

2. *$\mathcal{A}$ erzeugt durch Eingabe des Sicherheitsparameters 1^k in den Schlüsselerzeugungsalgorithmus G ein Schlüsselpaar:*

$$(\mathsf{pk}, \mathsf{sk}) \in G(1^k).$$

3. *$\mathcal{A}$ erzeugt zwei Nachrichten m_0 und m_1, die insbesondere von pk oder sk abhängen können.*

4. *Ferner erzeugt $\mathcal{A}$ gültige Signaturen auf die Nachrichten m_0 und m_1. Im Folgenden sei $s_i := (m_i, \sigma_I(m_i))$, $i = 0, 1$.*

5. *$\mathcal{A}$ sendet (s_0, s_1) und pk an das Orakel.*

6. *$\mathcal{A}$ führt parallel und nicht zwingend unabhängig voneinander das interaktive Protokoll V_I mit dem Orakel $\mathcal{O}$ aus. Dabei übernimmt das Orakel die Rolle von zwei ehrlichen Empfängern $\mathcal{R}_b$ und $\mathcal{R}_{1-b}$. Die Eingabe von $\mathcal{A}$ in beiden*

Protokollen ist $(1^k, \mathsf{pk}, \mathsf{sk}, s_0, s_1)$ *und die Protokollmitschriften aus den Signaturprotokollen. Die Eingabe von* $\mathcal{O}$ *in der Rolle von* $\mathcal{R}_b$ *ist* $(\mathsf{pk}, \overline{\mathsf{par}}_b, s_b)$*, die von* $\mathcal{O}$ *in der Rolle von* $\mathcal{R}_{1-b}$ *ist* $(\mathsf{pk}, \overline{\mathsf{par}}_{1-b}, s_{1-b})$*.*

7. $\mathcal{A}$ *gibt ein Bit* b' *aus.*
8. *Der Angreifer* $\mathcal{A}$ *gewinnt das Spiel, wenn* $b' = b$*.*

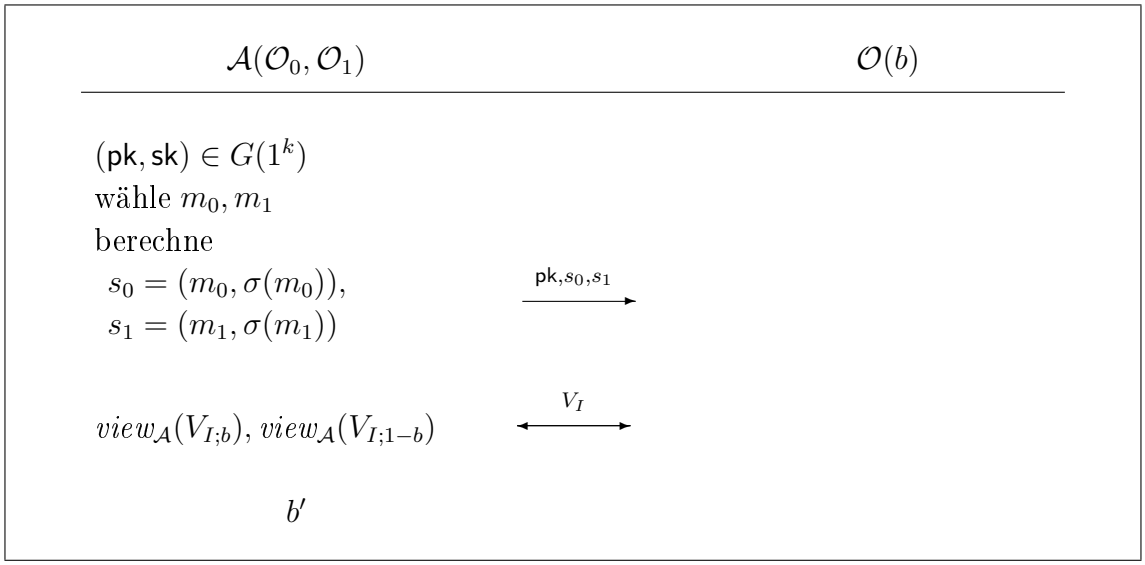


Abbildung 5.4: Spiel 5.3

Man beachte, dass $\mathcal{A}$ nach Schritt 7 nur die Eingaben der Empfänger und seine eigenen Protokollansichten der durchgeführten Verifikationsprotokolle kennt, d.h. er muss seine Wahl von b' aufgrund dieser Informationen treffen.

Im Fall von Spiel 5.3 sind $\mathcal{A}$ Angriffsphasen vor der Wahl der Nachrichten-Signatur-Paare und nach Schritt 6 erlaubt. Dazu hat er auch in diesem Fall Zugriff auf zwei Orakel $\mathcal{O}_0$ und $\mathcal{O}_1$, die unter der Eingabe von $\overline{\mathsf{par}}_0$ und $\overline{\mathsf{par}}_1$ die Empfänger $\mathcal{R}_0$ und $\mathcal{R}_1$ repräsentieren. Dabei ist wie oben nur eine Anzahl von Anfragen erlaubt, die polynomiell in k ist. Ferner wird nur ein einziges Orakel verwendet, wenn V_I keine Zusatzeingaben von $\mathcal{R}$ vorsieht.

Definition 5.3 *Es sei* $k \in \mathbb{N}$ *ein Sicherheitsparameter und* $\Sigma_I = (G, \sigma_I, V_I)$ *eine interaktive Signatur. Das interaktive Verifikationsprotokoll* V_I *heißt* **rechnerisch blind** *(im Sicherheitsparameter* k*), falls es für jeden effizienten Angreifer* $\mathcal{A}$ *mit Zugriff auf zwei wie oben beschriebene Orakel* $\mathcal{O}_0$ *und* $\mathcal{O}_1$ *eine in* k *vernachlässigbare Funktion* ν *gibt, sodass für die Erfolgswahrscheinlichkeit* $\varepsilon(k)$ *von* $\mathcal{A}$ *in Spiel 5.2 gilt:*

$$1/2 - \varepsilon(k) = \nu(k).$$

Es zeigt sich auch im Falle der rechnerisch blinden interaktiven Signaturen, dass die Blindheit nicht zwingend im Signaturprotokoll erreicht werden muss, da wie im vorigen Abschnitt aus der rechnerischen Blindheit des Verifikationsprotokolls schon die rechnerische Blindheit der gesamten interaktiven Signatur folgt:

Satz 5.2 *Ist das Verifikationsprotokoll V_I einer interaktiven Signatur $\Sigma_I = (G, \sigma_I, V_I)$ rechnerisch blind, so ist auch die interaktive Signatur Σ_I rechnerisch blind.*

Beweis. Sei k ein Sicherheitsparameter, $\Sigma_I = (G, \sigma_I, V_I)$ ein interaktives Signaturschema und V_I rechnerisch blind. Wir nehmen an, dass Σ_I nicht rechnerisch blind ist. Dann gibt es einen effizienten Angreifer, der Spiel 5.1 mit einer Wahrscheinlichkeit von $\varepsilon(k)$, die nicht vernachlässigbar von $1/2$ abweicht, gewinnt. Wir konstruieren einen effizienten Angreifer $\mathcal{A}^*$, der mit hinreichend großer Wahrscheinlichkeit Spiel 5.3 gewinnt:

1. Der Empfänger $\mathcal{R}$ wählt ein Bit $b \in_R \{0,1\}$ und ggf. seine Zusatzeingaben $\overline{\mathsf{par}}_0$ und $\overline{\mathsf{par}}_1$. Das Orakel $\mathcal{O}$ erhält als Eingabe b sowie $\overline{\mathsf{par}}_0$ und $\overline{\mathsf{par}}_1$.
2. $\mathcal{A}^*$ startet $\mathcal{A}$.
3. $\mathcal{A}$ generiert ein Schlüsselpaar $(\mathsf{pk}, \mathsf{sk}) \in G(1^k)$ und Nachrichten (m_0, m_1). Er sendet (m_0, m_1) und pk an $\mathcal{A}^*$.
4. $\mathcal{A}^*$ führt gemäß Spiel 5.1 anstelle des Orakels das Signaturprotokoll σ_I mit $\mathcal{A}$ durch. Dazu wählt er $c = 0$. Als Ergebnis erhält $\mathcal{A}^*$ Signaturen $\sigma_I(m_0)$ und $\sigma_I(m_1)$.
5. $\mathcal{A}^*$ gibt (s_0, s_1) und pk an das Orakel und führt das Verifikationsprotokoll V_I parallel und nicht zwingend unabhängig voneinander bzgl. $\mathcal{R}_b$ und $\mathcal{R}_{1-b}$ mit $\mathcal{O}$ durch. Dabei gibt er alle Werte an $\mathcal{A}$ weiter und verwendet dessen Ergebnisse als Antworten für das Orakel.
6. Nach Beendigung des Protokolls gibt $\mathcal{A}$ ein Bit b' aus.
7. Die Ausgabe von $\mathcal{A}^*$ ist ebenfalls b'.

Man beachte, dass wie oben die Eingabe der Zusatzinformationen par_0 und par_1 unproblematisch ist: Verlangt das Protokoll σ_I die Zusatzinformationen, so wählt $\mathcal{A}^*$ beliebige Werte par_0 und par_1 bevor er $\mathcal{A}$ startet. Anderenfalls hat das Orakel von $\mathcal{R}$ die entsprechenden Informationen bereits erhalten.

Mit dem gleichen Vorgehen gelingt es $\mathcal{A}^*$, auf Anfragen von $\mathcal{A}$ zu reagieren, wenn dieser in der Angriffsphase Zugriff auf die Orakel $\mathcal{O}_0$ und $\mathcal{O}_1$ verlangt, da $\mathcal{A}^*$ Zugriff

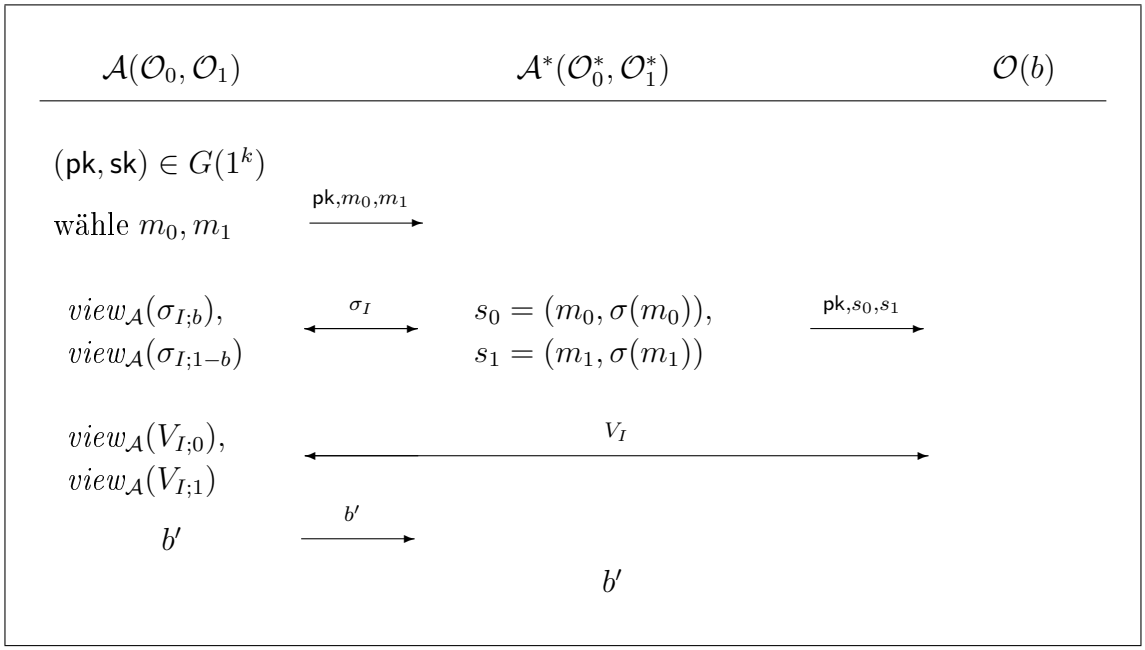


Abbildung 5.5: Beweis zu Satz 5.2

auf seine eigenen Orakel $\mathcal{O}_0^*$ und $\mathcal{O}_1^*$ bezüglich des Verifikationsprotokolls hat.

Insgesamt ist $\mathcal{A}^*$ effizient, da $\mathcal{A}$ effizient ist. Ferner gelingt der Angriff von $\mathcal{A}^*$ dann, wenn der Angriff von $\mathcal{A}$ auf die ganze Signatur gelingt: Gibt $\mathcal{A}$ das Bit $b' = 0$ aus, so weiß $\mathcal{A}$, dass die Protokollmitschriften nicht vertauscht wurden, und gibt ebenfalls $b' = 0$ aus. Ist die Ausgabe von $\mathcal{A}$ das Bit $b' = 1$, so wurden die Protokollmischriften vertauscht, und auch $\mathcal{A}^*$ gibt $b' = 1$ aus. $\mathcal{A}$ gelingt dies mit Wahrscheinlichkeit $\varepsilon(k)$, sodass $\mathcal{A}^*$ mit einer Wahrscheinlichkeit von mindestens $\varepsilon(k)$ Spiel gewinnt.

Insgesamt folgt die Behauptung. □

Das hier vorgestellte Modell für Blindheit liefert den stärksten Begriff im Kontext der rechnerischen Blindheit. Im nächsten Abschnitt wird ein etwas schwächerer Blindheitsbegriff diskutiert.

5.4 Rechnerische Blindheit unter sequentiellen Angriffen

Zum Abschluss dieses Kapitels werden sequentielle Angriffe untersucht. Auch wenn es zunächst scheint, als ob dieser Begriff durch die Forderung nach sequentiellen An-

griffen in allen vorangegangenen Definitionen abgedeckt würde, lässt sich hier doch ein etwas schwächerer Begriff definieren.

Es stellt sich zunächst die Frage, wie ein Spiel, das ausschließlich sequentielle Angriffe modelliert, im Kontext der rechnerischen Blindheit aussieht. Von einer unter einem sequentiellen Angriff rechnerisch blinden interaktiven Signatur Σ_I erwartet man, dass ein Angreifer (bzw. Signierer), der eine Signatur nach der anderen erstellt und diese der Reihe nach in der Rolle des Verifizierers wiedersieht, die Protokollansichten nicht zuordnen kann. In diesem Szenario kann der Angreifer seine Aufgabe aber in jedem Fall lösen, da sich das Verifikationsprotokoll direkt dem Signaturprotokoll anschließt. Somit ist es wenig sinnvoll, sequentielle Angriffe gleichzeitig auf beide Protokolle zu betrachten. Statt dessen betrachten wir zunächst sequentielle Angriffe auf die beiden einzelnen Protokolle Σ_I und V_I, um einen adäquaten Blindheitsbegriff für Σ_I zu formulieren. Man beachte, dass in allen folgenden Überlegungen auch in den Angriffsphasen keine parallelen Ausführungen der Protokolle gestattet sind.

Wir beginnen mit der rechnerischen Blindheit des Signaturprotokolls unter sequentiellen Angriffen. Im Signaturprotokoll besteht die Notwendigkeit, dass der Angreifer zwei Protokolldurchläufe ausführt, um in den Besitz von zwei Nachrichten-Signatur-Paaren zu kommen. Somit ergibt sich hier keine Möglichkeit, den Angreifer in dem die Sicherheit definierenden Spiel zu der sequentiellen Durchführung der Protokolle zu zwingen. Dementsprechend unterschiedet sich das folgende Spiel nur in einem einzigen Punkt von Spiel 5.2, nämlich in der *Forderung* nach sequentiellen Protokolldurchführungen an den Angreifer.

Spiel 5.4 *Es werden die Schritte 1 bis 4 und die Schritte 6 bis 8 von Spiel 5.2 durchgeführt. In Schritt 5 fordern wir, dass die Protokolldurchführungen sequentiell erfolgen.*

Dann ist es sinnvoll, σ_I als rechnerisch blind unter einem sequentiellen Angriff anzusehen, wenn es keinen effizienten Angreifer gibt, der im Spiel 5.2 einen zu großen Vorteil durch die Protokolldurchführung erhalten hat. Dabei hat $\mathcal{A}$ an den gleichen Stellen wie in Spiel 5.2 Zugriff auf ein Orakel $\mathcal{O}_{0,1}$. Dieses Orakel übernimmt die Rolle der beiden Orakel $\mathcal{O}_0$ und $\mathcal{O}_1$ aus Definition 5.2, d.h. es führt auf Anfrage sowohl Protokolle zu der Zusatzeingabe $\overline{\mathsf{par}}_0$ als auch zu der Zusatzeingabe $\overline{\mathsf{par}}_1$ aus. Allerdings lässt $\mathcal{O}_{0,1}$ immer nur ein Protokoll zur gleichen Zeit zu.

Definition 5.4 *Es sei $k \in \mathbb{N}$ ein Sicherheitsparameter und $\Sigma_I = (G, \sigma_I, V_I)$ eine interaktive Signatur. Das interaktive Signaturprotokoll σ_I heißt* **rechnerisch blind unter einem sequentiellen Angriff** *(im Sicherheitsparameter k), falls es für jeden*

effizienten Algorithmus $\mathcal{A}$ mit Zugriff auf ein Orakel $\mathcal{O}_{0,1}$ eine in k vernachlässigbare Funktion ν gibt, sodass für die Erfolgswahrscheinlichkeit von $\mathcal{A}$ in Spiel 5.4 gilt:

$$1/2 - \varepsilon(k) = \nu(k).$$

Für sequentiellen Angriffe auf V_I geben wir ein etwas anderes Modell vor: Hier ist der Angreifer in der Lage, sich Nachrichten-Signatur-Paare vorzugeben, sodass es genügt, wenn er einem Protokolldurchlauf das korrekte Nachrichten-Signatur-Paar zuordnet. Damit können wir eine, von anderen Protokolldurchführungen unabhängige, sequentielle Durchführung des Verifikationsprotokolls erzwingen.

Spiel 5.5 *Es werden die Schritte 2 bis 5 und die Schritte 7 bis 9 von Spiel 5.3 durchgeführt. Schritt 1 und Schritt 6 werden durch folgende Schritte ersetzt:*

1'. $\mathcal{R}$ wählt ein Bit $b \in_R \{0,1\}$ und ggf. die Zusatzeingabe $\overline{\mathsf{par}}$. Das Orakel $\mathcal{O}$ erhält als Eingabe b sowie $\overline{\mathsf{par}}$.

6'. $\mathcal{A}$ führt das Protokoll V_I mit dem Orakel $\mathcal{O}$ durch. Dabei übernimmt das Orakel die Rolle des ehrlichen Empfängers $\mathcal{R}_b$ mit den Eingaben $(\mathsf{pk}, \overline{\mathsf{par}}, s_b)$. Die Eingaben von $\mathcal{A}$ sind $(1^k, \mathsf{pk}, \mathsf{sk}, s_0, s_1)$.

Mit Hilfe dieses Spiels formulieren wir die rechnerische Blindheit von V_I unter einem sequentiellen Angriff. Dabei hat $\mathcal{A}$ an den gleichen Stellen wie in Spiel 5.3 Zugriff auf ein Orakel $\mathcal{O}_{\overline{\mathsf{par}}}$. Dieses Orakel übernimmt die Rolle der eines ehrlichen Empfängers $\mathcal{R}$ mit Zusatzeingaben $\overline{\mathsf{par}}$, d.h. es führt auf Anfrage Protokolle zu der Zusatzeingabe $\overline{\mathsf{par}}$ durch, wobei die Protokolldurchläufe unabhängig voneinander sind.

Definition 5.5 *Es sei $k \in \mathbb{N}$ ein Sicherheitsparameter, ν eine vernachlässigbare Funktion. Das interaktive Signaturprotokoll σ_I heißt* **rechnerisch blind unter einem sequentiellen Angriff** *(im Sicherheitsparameter k), falls es für jeden effizienten Algorithmus $\mathcal{A}$ mit Zugriff auf ein Orakel $\mathcal{O}_{\overline{\mathsf{par}}}$ eine in k vernachlässigbare Funktion ν gibt, sodass für die Erfolgswahrscheinlichkeit von $\mathcal{A}$ in Spiel 5.4 gilt:*

$$1/2 - \varepsilon(k) = \nu(k).$$

Eine sinnvolle Forderung für den entsprechenden Blindheitsbegriff bzgl. Σ_I ist sicher, dass kein betrügender Signierer in der Lage sein sollte, beide Spiele unabhängig voneinander zu gewinnen, da er in diesem Fall seine Protokollansichten von σ_I und V_I zuordnen könnte. Dementsprechend formulieren wir die

Definition 5.6 *Die interaktive Signatur $\Sigma_I = (G, \sigma_I, V_I)$ heißt* **rechnerisch blind unter einem sequentiellen Angriff** *(im Sicherheitsparameter k), falls eins der Protokolle σ_I oder V_I rechnerisch blind unter einem sequentiellen Angriff ist.*

Offenbar besteht ein analoger Zusammenhang zwischen den so definierten Begriffen, wie in den vorangegangenen Abschnitten:

Satz 5.3 *Sei $\Sigma_I = (G, \sigma_I, V_I)$ eine interaktive Signatur.*

(a) *Ist σ_I rechnerisch blind unter einem sequentiellen Angriff, so ist Σ_I rechnerisch blind unter einem sequentiellen Angriff.*

(b) *Ist V_I rechnerisch blind unter einem sequentiellen Angriff, so ist Σ_I rechnerisch blind unter einem sequentiellen Angriff.*

Beweis. Die Behauptung folgt direkt aus den Definitionen 5.4, 5.5 und 5.6. □

Bemerkung 5.3 *Offensichtlich ist die rechnerische Blindheit unter sequentiellen Angriffen für beide Protokolle jeweils ein schwächerer Begriff als die rechnerische Blindheit unter parallelen Angriffen.*

5.5 Fazit

Wir haben in diesem Kapitel den rechnerische Blindheitsbegriff unter verschiedenen Angriffstypen sowohl für interaktive Signaturen als auch für das Signatur- und Verifikationsprotokoll definiert und diskutiert. Wir haben gesehen, dass wir bzgl. der rechnerischen Blindheit analoge Ergebnisse zu den Ergebnissen aus Kapitel 4 erhalten:

- Jede rechnerisch blinde Signatur ist auch eine rechnerisch blinde interaktive Signatur.
- Jedes rechnerisch blinde Verifikationsprotokoll erzeugt ebenfalls eine rechnerisch blinde interaktive Signatur.

Ferner konnte ein Sicherheitsmodell für sequentielle Angriffe angegeben werden, in dem die gleichen Zusammenhänge bestehen. Damit haben wir jeweils zwei hinreichende Bedingungen sowohl für rechnerisch blinde interaktive Signaturen als auch für unter einem sequentiellen Angriff rechnerisch blinde interaktive Signaturen gefunden.

Auch für die Konstruktion von rechnerisch blinden interaktiven Signaturen stellt sich die Frage nach der Beschaffenheit der Kryptobausteine. Ferner wurde bisher noch keine Aussage über die Existenz von rechnerisch blinden Signaturen als schwächere Variante der perfekt blinden Signaturen getroffen. Für die Diskussion dieser

Fragen sei auf die folgenden Kapitel verwiesen.

Damit können die bisherigen Ergebnisse wie folgt zusammengefasst werden: Der Begriff der blinden interaktiven Signatur wurde eingeführt. Dieser erfasst die herkömmlichen blinden Signaturen, die interaktive Signaturen mit blindem Signaturprotokoll sind. Aber es wird auch die Klasse der Signaturen mit blindem Verifikationsprotokoll abgedeckt. Als hinreichende Bedingung für blinde interaktive Signaturen konnte sowohl für perfekte als auch für rechnerische Blindheit lokalisiert werden, dass die Verifikationsprotokolle, genau wie die Signaturprotokolle, beliebige Nachrichten-Signatur-Paare geeignet vor dem Signierer oder dem Verifizierer verbergen müssen.

Teil II

Konstruktion interaktiver Signaturen

Kapitel 6

Blindheit im Signaturprotokoll

Die im vorigen Teil der Arbeit vorgestellten Bedingungen für blinde interaktive Signaturen sind naturgemäß sehr allgemein gehalten. Es stellt sich einerseits die Frage nach der Existenz solcher interaktiven Signaturen und andererseits die Frage, zu welchen Bedingungen an interaktive Signaturen die Überlegungen aus den vorangegangenen Kapiteln im Fall einer konkreten interaktiven Signatur führen. Diesen Fragen werden wir nun nachgehen:

Wir werden im Folgenden zwei verschiedene Konstruktionsprinzipien für blinde interaktive Signaturen beleuchten. In diesem Kapitel wird die Blendung während des Signaturprotokolls untersucht, während im nächsten Kapitel eine interaktive Signatur mit blindem Verifikationsprotokoll konstruiert wird.

Das Konstruktionsprinzip, das in diesem Kapitel vorgestellt wird, wird auf drei Ebenen untersucht:

Ebene 1: Auf der ersten Ebene wird, ausgehend von einem allgemein gegebenen Signaturschema, ein Konstruktionsprinzip für interaktive Signaturen angegeben. Auf dieser Ebene ist es nicht möglich, die Bedingungen aus den Kapiteln 4 und 5 in eine konkretere Form zu bringen. Zu diesem Zweck begeben wir uns auf die zweite Ebene.

Ebene 2: Hier untersuchen wir nicht mehr Konstruktionen, die auf allgemein gegebenen Signaturschemata aufbauen, sondern wir legen uns auf zwei Signaturen fest. Dementsprechend werden auf dieser Ebene die folgenden Ziele verfolgt:

1. Es werden Bedingungen angegeben, unter denen die interaktive RSA-Signatur (perfekt oder rechnerisch) blind ist.

2. Ebenso werden Bedingungen angegeben, unter denen die interaktive Schnorr-

Signatur (perfekt oder rechnerisch) blind ist.

Um diese Ziele zu erreichen, werden in den Abschnitten 6.1 und 6.4 zunächst die interaktive RSA-Signatur und die interaktive Schnorr-Signatur beschrieben, bevor in den Abschnitten 6.2 und 6.5 die Blindheit der beiden Signaturen untersucht wird. Für die interaktive RSA-Signatur werden wir notwendige und hinreichende Bedingungen sowohl für ein perfekt blindes als auch für ein rechnerisch blindes Signaturprotokoll angeben. Für die Schnorr-Signatur wird dies nur für die perfekte Blindheit möglich sein, für die rechnerische Blindheit erhalten wir eine notwendige Bedingung. Damit kommen wir zur dritten Ebene.

Ebene 3: Hier werden für beide Signaturen spezielle Varianten untersucht. Auf dieser Ebene werden zwei Punkte demonstriert:

1. Einerseits wird gezeigt, dass die abstrakten Überlegungen aus den vorherigen Kapiteln auf bekannte blinde Signaturen angewendet werden können. Es wird insbesondere gezeigt, dass diese bekannten Realisierungen keineswegs zwingend sind.

2. Andererseits wird gezeigt, dass insbesondere die Überlegungen bzgl. der rechnerischen Sicherheit verwendet werden können, um einen praxisnahen Blindheitsbegriff zu etablieren.

Die untersuchten interaktiven Signaturen werden in den Abschnitten 6.3 und 6.6 vorgestellt und mit Hilfe der Ergebnisse aus den Abschnitten 6.2 und 6.5 auf ihre Blindheitseigenschaften hin untersucht. An dieser Stelle werden auch die Sicherheitseigenschaften dieser Signaturen diskutiert.

Wir beginnen mit der ersten Ebene. Dazu betrachten wir das Signaturprotokoll. Nach der Definition einer interaktiven Signatur sind als Signaturprotokoll beliebige Zwei-Parteien-Protokolle erlaubt, in denen $\mathcal{R}$ am Ende des Protokolls eine Signatur auf seine Nachricht m erhält. Ein naheliegender Ansatz, um eine blinde interaktive Signatur zu konstruieren, ist der folgende: Der Empfänger $\mathcal{R}$ läßt nicht die Nachricht m sondern eine von m verschiedene Nachricht $\overline{m}$ von $\mathcal{S}$ signieren. Er erhält von dem Signierer eine Signatur $\sigma(\overline{m})$ auf $\overline{m}$ und berechnet nun eine Signatur $\sigma(m)$ auf die Nachricht m.

Bevor wir uns den konkreten Signaturen zuwenden, widmen wir uns der Frage, unter welchen Voraussetzungen man mit dem oben beschriebenen Vorgehen eine interaktive Signatur erhält:

Im Folgenden sei ein Signaturschema $\Sigma = (G, \sigma, V)$ mit Nachrichtenraum $\mathcal{M}$ gegeben. Es sei $S(m)$ die Menge aller gültigen Signaturen auf eine Nachricht $m \in \mathcal{M}$. Wie bereits erwähnt, gibt es zu jedem Signaturschema verschiedene interaktive Varianten. In den in Kapitel 3 vorgestellten Varianten war der Empfänger weitgehend nicht mit eigenen Rechenschritten beteiligt, seine Einflussmöglichkeiten auf das Aussehen der Signatur waren somit prinzipiell stark eingeschränkt. Mit der Einführung der **Blendungsfunktionen** erhält der Empfänger einen größeren Handlungsspielraum. Dazu seien die Mengen $\mathcal{Z}$, Par und ferner für alle $z \in \mathcal{Z}$ und $\mathsf{par} \in \mathsf{Par}$ zwei effizient berechenbare Abbildungen

$$\varphi_{\mathsf{par},z} : \mathcal{M} \to \mathcal{M} \tag{6.1}$$

und

$$\psi_{\mathsf{par},z} : S(\varphi_{\mathsf{par},z}(m)) \to S(m) \tag{6.2}$$

gegeben, wobei $\mathcal{Z}$ eine Menge von Zahlentupeln (wir sprechen auch von Tupeln von **Blendungsvariablen**) sei, die zur Blendung von Nachrichten $m \in \mathcal{M}$ und der zugehörigen Signaturen $\sigma(m)$ geeignet sind. Par sei eine Menge, die sowohl öffentlich zugängliche Parameter als auch geheime Parameter von $\mathcal{R}$ enthält. Dazu können der öffentliche Schlüssel von $\mathcal{S}$ oder anderer Teilnehmer gehören, aber auch ein geheimer Schlüssel von $\mathcal{R}$, zum Beispiel für eine symmetrische Verschlüsselung. Grundsätzlich soll hier die Möglichkeit, dass keine Blendungsvariablen oder öffentliche Parameter verwendet werden, nicht ausgeschlossen werden, sodass $\mathcal{Z} = \emptyset$ oder $\mathsf{Par} = \emptyset$ erlaubt ist. Ferner gehen wir, wie schon in Kapitel 5, davon aus, dass die Parameter par nur im Signaturprotokoll verwendet werden.

In den in Abbildung (6.1) und (6.2) vorgestellten interaktiven Protokollen wird die oben beschriebene Konstruktion umgesetzt. Dazu gehen wir davon aus, dass der Signierer bereits den Algorithmus G ausgeführt und ein Schlüsselpaar $(\mathsf{pk}, \mathsf{sk}) \in G(1^k)$ erzeugt hat.

Der folgende Satz zeigt, dass wir mit dieser allgemeinen Konstruktion tatsächlich eine interaktive Signatur erhalten.

Satz 6.1 *Es sei $\Sigma = (G, \sigma, V)$ ein Signaturschema und $\varphi_{\mathsf{par},z}$ sowie $\psi_{\mathsf{par},z}$ für $\mathsf{par} \in \mathsf{Par}$ und $z \in \mathcal{Z}$ Abbildungen gemäß (6.1) und (6.2). Gilt für alle $m \in \mathcal{M}$*

$$\psi_{\mathsf{par},z}(\sigma(\varphi_{\mathsf{par},z}(m))) \in S(m), \tag{6.3}$$

so ist $\Sigma_I = (G, \sigma_I, V_I)$ mit σ_I und V_I gemäß Abbildung (6.1) bzw. (6.2) eine interaktive Signatur.

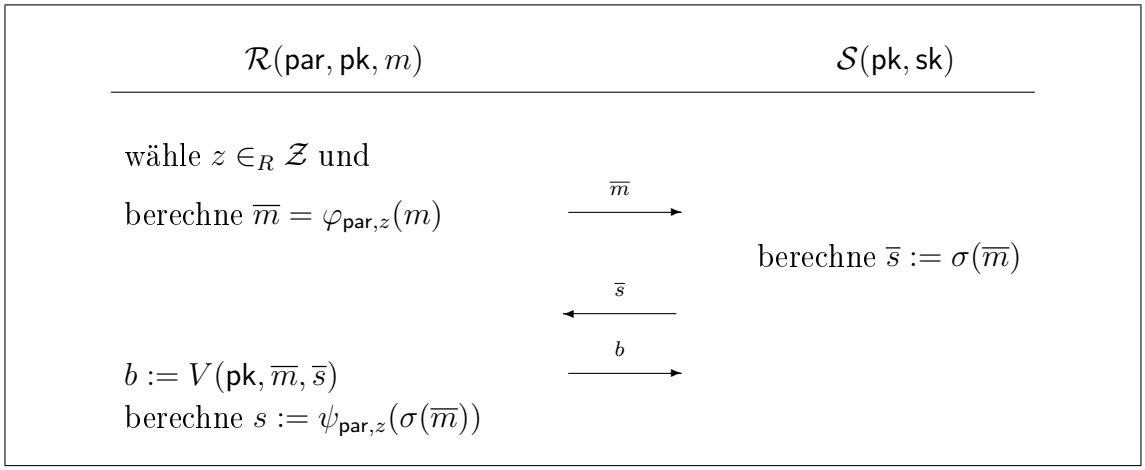


Abbildung 6.1: Signaturprotokoll σ_I

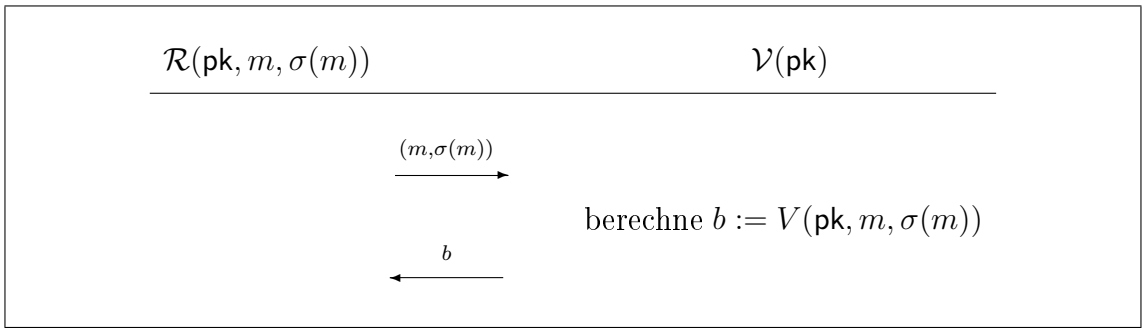


Abbildung 6.2: Verifikationsprotokoll V_I

Beweis. Zu zeigen ist die Durchführbarkeit der Protokolle, d.h. bei korrektem Verhalten aller Teilnehmer ist die Ausgabe von $\mathcal{V}$ im Verifikationsprotokoll V_I okay. Wir betrachten zunächst das Signaturprotokoll. Verhalten sich $\mathcal{R}$ und $\mathcal{S}$ korrekt, so gilt wegen der Durchführbarkeit von Σ:

$$V(\mathsf{pk}, \overline{m}, \overline{s}) = 1.$$

Damit ist die Ausgabe von $\mathcal{S}$ $\mathsf{completed}$ und $\mathcal{R}$ gibt s aus. Wegen Gleichung (6.3) gilt $s \in S(m)$. Somit ist $s = \sigma(m)$ und es gilt $V(\mathsf{pk}, m, s) = 1$. Damit gibt $\mathcal{V}$ im Verifikationsprotokoll bei Eingabe von $(m, \sigma(m))$ okay und $\mathcal{R}$ $\mathsf{completed}$ aus. □

Offensichtlich sind einerseits $\varphi_{\mathsf{par},z}$ und $\psi_{\mathsf{par},z}$ von dem konkreten Signaturschema Σ abhängig, das verwendet wird. Andererseits wird die Blindheit dieser Konstruktion von $\varphi_{\mathsf{par},z}$ und $\psi_{\mathsf{par},z}$ abhängen. In dem folgenden Abschnitten wird diese Konstruktion exemplarisch für zwei Signaturschemata untersucht, für die RSA- und die Schnorr-Signatur. Wir werden zunächst sehen, dass der hier vorgestellte Ansatz zu

einer Verallgemeinerung der perfekt blinden Signatur von Chaum und zu einer übergeordneten Darstellung von verschiedenen Typen von blinden Schnorr-Signaturen führt. Weiterhin werden zu beiden Signaturen rechnerisch blinde Varianten vorgestellt, die ebenfalls auf dem Konzept der Blendungsfunktionen beruhen.

6.1 Die interaktive RSA-Signatur

Wir betrachten die RSA-Signatur: Sei $\Sigma = (G, \sigma, V)$ mit G, σ und V wie in Abschnitt 1.1.3.1 beschrieben und $\mathcal{M} = \mathbb{Z}_n$. Sei dazu $(\mathsf{pk}, \mathsf{sk}) = ((n, e), d) \in G(1^k)$ als Ausgabe des Schlüsselerzeugungsalgorithmus' das Schlüsselpaar von $\mathcal{S}$.

Seien $\varphi_{\mathsf{par},z}$ und $\psi_{\mathsf{par},z}$ für $\mathsf{par} \in \mathsf{Par}$ und $z \in \mathcal{Z}$ gemäß (6.1) bzw. (6.2) gegeben. Die oben angegebenen Protokolle σ_I und V_I nehmen hier die in Abbildung 6.3 bzw. Abbildung 6.4 dargestellte Gestalt an.

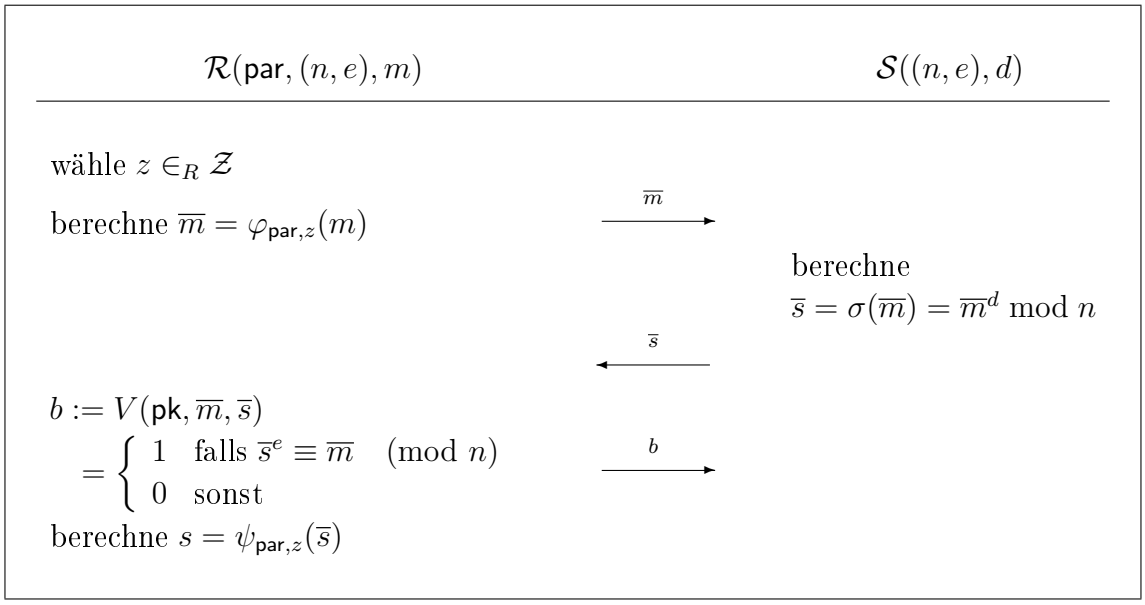


Abbildung 6.3: Signaturprotokoll σ_I

Wie wir oben gesehen haben, müssen geeignete Blendungsfunktionen $\varphi_{\mathsf{par},z}$ und $\psi_{\mathsf{par},z}$ zu gegebenen $\mathsf{par} \in \mathsf{Par}$ und $z \in \mathcal{Z}$ für alle $m \in \mathcal{M}$ die Bedingung

$$\psi_{\mathsf{par},z}(\sigma(\varphi_{\mathsf{par},z}(m))) \in S(m)$$

erfüllen, damit wir mit (G, σ_I, V_I) eine interaktive Signatur erhalten. Bei der RSA-Signatur zum Schlüsselpaar $((e, n), d)$ ist $S(m) = \{m^d \bmod n\}$, d.h. für gegebene $\mathsf{par} \in \mathsf{Par}$ und $z \in \mathcal{Z}$ lautet die Bedingung in diesem Fall: Für alle $m \in \mathbb{Z}_n$ gilt

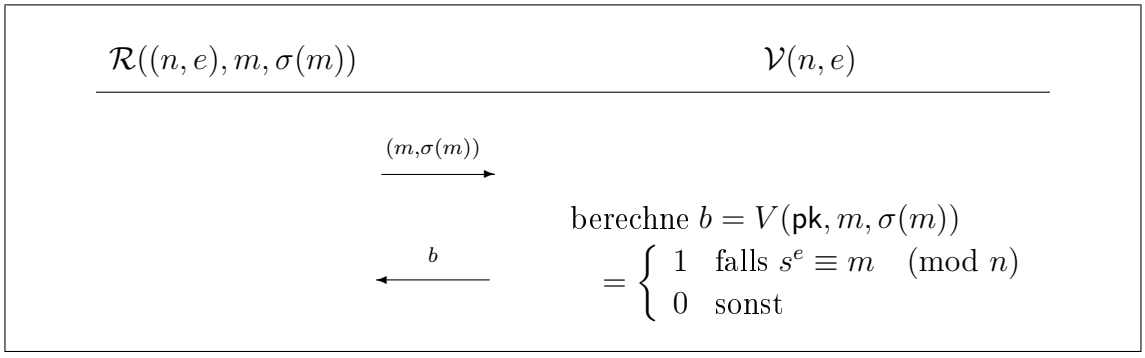


Abbildung 6.4: Verifikationsprotokoll V_I

$$\psi_{\mathsf{par},z}(\varphi^d_{\mathsf{par},z}(m))) = m^d \bmod n. \tag{6.4}$$

Dieses Ergebnis wird noch einmal in dem folgenden Satz zusammengefasst.

Satz 6.2 *Sei $\Sigma = (G, \sigma, V)$ die RSA-Signatur und die Protokolle σ_I und V_I gemäß Abbildungen 6.3 und 6.4 sowie für alle $z \in \mathcal{Z}$ und $\mathsf{par} \in \mathsf{Par}$ effizient berechenbare Abbildungen $\varphi_{\mathsf{par},z}$ und $\psi_{\mathsf{par},z}$ gemäß (6.1) und (6.2) gegeben. Dann gilt:*
Ist Gleichung (6.4) erfüllt, so ist $\Sigma_I = (G, \sigma_I, V_I)$ eine interaktive Signatur.

Beweis. Die Behauptung folgt unmittelbar aus Satz 6.1 und den obigen Überlegungen. □

Damit haben wir eine interaktive Signatur auf der Basis der RSA-Signatur konstruiert. Im nächsten Abschnitt wird die Blindheit dieser interaktiven Signatur untersucht.

6.2 Analyse der Blindheit

In diesem Abschnitt werden die Blindheitseigenschaften der gerade vorgestellten interaktiven Signatur auf Basis der RSA-Signatur analysiert. Dazu unterscheiden wir perfekt blinde interaktive RSA-Signaturen und rechnerisch blinde RSA-Signaturen. In beiden Fällen werden wir notwendige und hinreichende Bedingungen für ein blindes Signaturprotokoll angeben.

6.2.1 Perfekt blinde interaktive RSA-Signaturen

Ziel dieses Abschnitts ist es, notwendige und hinreichende Bedingungen für die perfekte Blindheit des Signaturprotokolls in der vorgestellten interaktiven RSA-

Signatur anzugeben. Dazu sei an die Definition eines perfekt blinden Signaturschemas erinnert: σ_I ist genau dann perfekt blind, wenn die Protokollansicht des Signierers und das Nachrichten-Signatur-Paar stochastisch unabhängig sind. Der folgende Satz zeigt, dass in einer interaktiven Signatur nur die Blendung der Nachricht einen Beitrag zur Unabhängigkeit leisten muss, nicht aber die Blendung der Signatur selbst.

Satz 6.3 *Es sei das interaktive Signaturschema Σ_I aus Satz 6.2 zur RSA-Signatur gegeben. Unter der Voraussetzung, dass die Protokolle von Σ_I von einem ehrlichen Teilnehmer $\mathcal{R}$ durchgeführt werden, gilt:*

(a) *σ_I ist genau dann perfekt blind, wenn die* Zufallsvariable *m stochastisch unabhängig von der Zufallsvariablen $\overline{m}$ ist.*

(b) *Ist m stochastisch unabhängig von $\overline{m}$, so ist Σ_I perfekt blind.*

Beweis. Wir müssen nur die Aussage aus Teil (a) zeigen, da Teil (b) mit Satz 4.2 direkt aus Teil (a) folgt. Dazu beachte man, dass die allgemeine Konstruktion gerade Voraussetzung (a) aus Satz 4.1 erfüllt.

Seien zunächst m und $\overline{m}$ stochastisch unabhängig. Wir zeigen, dass die stochastische Unabhängigkeit von $(m, \sigma(m))$ und $(\overline{m}, \overline{s})$ folgt. Wegen $\sigma(m) = m^d$ und $\overline{s} = \overline{m}^d$ sind $\sigma(m)$ und $\overline{s}$ deterministische Abbildungen von m bzw. $\overline{m}$, sodass wir Gleichung (1.6) aus Lemma 1.2 anwenden können. Somit gilt $H(m, \sigma(m)) = H(m)$ sowie $H(\overline{m}, \overline{s}) = H(\overline{m})$ und ferner $H((m, \sigma(m)), (\overline{m}, \overline{s})) = H(m, \overline{m}) = H(m) + H(\overline{m})$, wobei das letzte Gleichheitszeichen wegen der stochastischen Unabhängigkeit von m und $\overline{m}$ gilt. Damit folgt insgesamt

$$\begin{aligned} \mathcal{I}[(m, \sigma(m)), (\overline{m}, \overline{s})] &= H(m, \sigma(m)) + H(\overline{m}, \overline{s}) - H((m, \sigma(m)), (\overline{m}, \overline{s})) \\ &= H(m) + H(\overline{m}) - (H(m) + H(\overline{m})) \\ &= 0. \end{aligned}$$

Damit sind $(m, \sigma(m))$ und $\mathit{view}_{\mathcal{S}}(\sigma_I)$ unabhängig, d.h. σ_I ist ein perfekt blindes Signaturprotokoll.

Sei nun σ perfekt blind. Dann gilt

$$H(m, \sigma(m)) + H(\overline{m}, \overline{s}) = H((m, \sigma(m)), (\overline{m}, \overline{s})),$$

und mit Lemma 1.2 folgt wie oben

$$\mathcal{I}[m, \overline{m}] = H(m) + H(\overline{m}) - H(m, \overline{m}) = 0. \qquad \square$$

Damit haben wir das Wesen der Blendungsfunktionen beschrieben: Um mit der angegebenen Konstruktion perfekt blinde interaktive Signaturen zu erhalten, muss man dafür sorgen, dass der Signierer einen Wert signiert, der stochastisch unabhängig von der Nachricht ist. Die Blendungsfunktion $\varphi_{\mathsf{par},z}$ muss also eine stochastische Unabhängigkeit von m und dem Bild von m erzeugen. Dies wird im Allgemeinen nur möglich sein, indem der zufällig gewählte Wert z verwendet wird. Der Parameter par kann diesbezüglich nur von untergeordneter Bedeutung sein.

6.2.2 Rechnerisch blinde interaktive RSA-Signaturen

In diesem Abschnitt werden wir die rechnerische Blindheit der vorgestellten interaktiven RSA-Signatur untersuchen. Unter Verwendung des in Abschnitt 1.1.2 vorgestellten Sicherheitsbegriffs für Kryptosysteme ist ein zu Satz 6.3 vergleichbares Ergebnis zu erzielen, indem man die Blendungsfunktion als Chiffre mit Schlüsselraum Par auffasst und die Menge $\mathcal{Z} = \emptyset$ setzt, also am Anfang des Protokolls keine Zufallszahl wählt. Damit hängen die Funktionen $\varphi_{\mathsf{par},z}$ und $\psi_{\mathsf{par},z}$ nicht mehr von z ab, d.h. es ist $\varphi_{\mathsf{par},z} = \varphi_{\mathsf{par}}$ und $\psi_{\mathsf{par},z} = \psi_{\mathsf{par}}$. Der nächste Satz zeigt, dass die rechnerische Sicherheit der Chiffre unter diesen Bedingungen hinreichend und notwendig für die rechnerische Blindheit der Signatur ist.

Satz 6.4 *Es sei das interaktive Signaturschema Σ_I aus Satz 6.2 zur RSA-Signatur mit $\mathcal{Z} = \emptyset$ gegeben. Ist zusätzlich zu den dort angegebenen Bedingungen*

$$\varphi_{\mathsf{par},z} \in \mathcal{F} := \{\varphi_{\mathsf{par}} : \mathbb{Z}_n \to \mathbb{Z}_n \mid \mathsf{par} \in \mathsf{Par}\},$$

wobei $\mathcal{F}$ eine Chiffre ist, so gilt unter der Voraussetzung, dass die Protokolle von Σ_I von einem ehrlichen Teilnehmer $\mathcal{R}$ durchgeführt werden:

(a) *σ_I ist genau dann rechnerisch blind, wenn $\mathcal{F}$ rechnerisch sicher unter einem Angriff auf zwei Schlüssel gemäß Definition 1.8 ist.*

(b) *Ist $\mathcal{F}$ rechnerisch sicher unter einem Angriff auf zwei Schlüssel gemäß Definition 1.8, so ist Σ_I rechnerisch blind.*

Beweis. Wir zeigen zuerst die Behauptung aus Teil (a), die Aussage von Teil (b) folgt direkt aus Teil (a) und Satz 5.1.

Sei zunächst σ_I rechnerisch blind. Wir nehmen an, dass es einen effizienten Algorithmus $\mathcal{A}$ gibt, der das Spiel aus Definition 1.8 gegen einen Empfänger $\mathcal{R}$ mit Wahrscheinlichkeit $\varepsilon(k)$ gewinnt, wobei $1/2 - \varepsilon(k)$ nicht vernachlässigbar im Sicherheitsparameter k ist, und konstruieren einen effizienten Algorithmus $\mathcal{A}^*$, der die

rechnerische Blindheit ebenfalls mit Erfolgswahrscheinlichkeit $\varepsilon(k)$ angreifen kann. $\mathcal{A}^*$ agiert in Spiel 5.2 wie folgt:

1. Der Empfänger $\mathcal{R}$ wählt ein Bit $b \in_R \{0,1\}$ und als Zusatzeingabe die Schlüssel par_0 und par_1 für $\mathcal{F}$. Das Orakel erhält als Eingabe b sowie par_0 und par_1.
2. $\mathcal{A}^*$ führt $G(1^k)$ aus und erhält so das Schlüsselpaar $(\mathsf{pk}, \mathsf{sk})$.
3. $\mathcal{A}^*$ startet $\mathcal{A}$, der Nachrichten (m_0, m_1) erzeugt.
4. $\mathcal{A}^*$ führt zusammen mit $\mathcal{O}$ Schritt 4 bis Schritt 6 von Spiel 5.2 aus. Dazu hält sich $\mathcal{A}^*$ an das Protokoll σ_I. Als Ergebnis erhält $\mathcal{A}^*$ die Protokollmitschriften der durchgeführten Protokolle, d.h. insbesondere $\overline{m}_b = \varphi_{\mathsf{par}_b}(m_b)$ und $\overline{m}_{1-b} = \varphi_{\mathsf{par}_{1-b}}(m_{1-b})$.
5. $\mathcal{A}^*$ gibt $\overline{m}_b$ und $\overline{m}_{1-b}$ an $\mathcal{A}$ weiter und $\mathcal{A}$ hat als Ausgabe ein Bit b'.
6. Die Ausgabe von $\mathcal{A}^*$ ist b'.

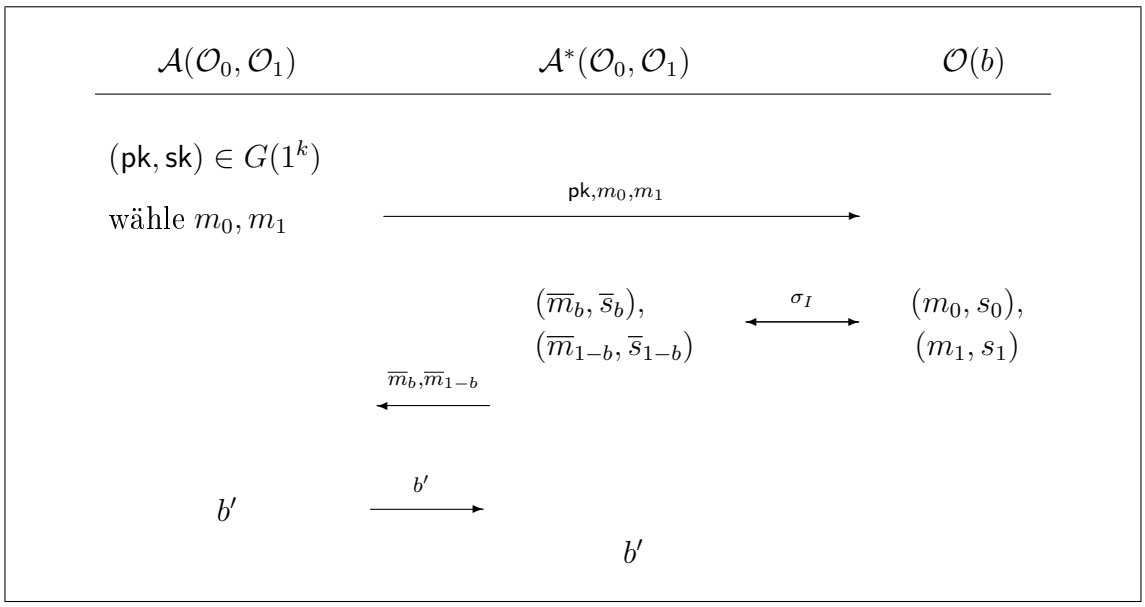


Abbildung 6.5: Beweis zu Satz 6.4

Verlangt $\mathcal{A}$ während des Spiels Zugriff auf seine Orakel $\mathcal{O}_0$ und $\mathcal{O}_1$, so kann $\mathcal{A}^*$ seine eigenen Orakel $\mathcal{O}_0^*$ und $\mathcal{O}_1^*$, wie in Abschnitt 5.2 beschrieben, verwenden, um die Anfragen gemäß dem oben beschriebenen Vorgehen zu beantworten.

Da A ein effizienter Algorithmus ist, gilt dasselbe offensichtlich auch für A^*. Ferner gelingt der Angriff von A^* genau dann, wenn A das Bit $b' = b$ ausgibt. Dem

Algorithmus A gelingt dies mit Wahrscheinlichkeit $\varepsilon(k)$, sodass auch A^* mit Wahrscheinlichkeit $\varepsilon(k)$ Erfolg hat.

Sei nun $\mathcal{F}$ rechnerisch sicher. Wir nehmen an, dass σ_I nicht rechnerisch blind ist, und konstruieren mit Hilfe eines effizienten Angreifers $\mathcal{A}$ auf die Blindheit der Signatur mit Erfolgswahrscheinlichkeit $\varepsilon(k)$ einen effizienten Angreifer $\mathcal{A}^*$, der ein Spiel gemäß Satz 1.1 gegen das dort beschriebene Orakel $\mathcal{O}$ ebenfalls mit $\varepsilon(k)$ gewinnt. Da σ_I nicht rechnerisch blind ist, ist $1/2 - \varepsilon(k)$ nicht vernachlässigbar im Sicherheitsparameter k. In dem im Folgenden beschriebenen Angriff nimmt $\mathcal{A}^*$ die Rolle des Orakels gegenüber $\mathcal{A}$ ein.

Zuvor stellen wir fest, dass $\mathcal{A}$ auch folgendes Spiel gegen ein leicht abgeändertes Orakel gewinnt, wenn er Spiel 5.2 gewinnt: Seien die Schritte 1 bis 5 wie in Spiel 5.2. In Schritt 6 schickt $\mathcal{A}$ nicht die Nachrichten-Signatur-Paare (m_0, s_0) und (m_1, s_1) an das Orakel zurück, sondern nur okay oder fail. Ist die Ausgabe fail, wird das Spiel abgebrochen, sonst gibt $\mathcal{A}$ ein Bit b' aus.

Diese Aussage ist richtig, da das Orakel in der RSA-Signatur ausschließlich Signaturen $\overline{s}_b$ und $\overline{s}_{1-b}$ akzeptiert, für die gilt

$$\overline{s}_b^e \equiv \overline{m}_b \pmod{n} \quad \text{und} \quad \overline{s}_{1-b}^e \equiv \overline{m}_{1-b} \pmod{n}.$$

Damit ist klar, dass die Signaturen tatsächlich zu den Nachrichten $\overline{m}_b$ bzw. $\overline{m}_{1-b}$ gehören, d.h. $\mathcal{A}$ hat keinerlei Möglichkeiten, in diesem Schritt auf die Nachricht einzuwirken. Da sich das Orakel dem Protokoll gemäß verhält, wird die Blendung korrekt von den Nachrichten-Signatur-Paaren entfernt, sodass der Angreifer die Signaturen auf m_0 und m_1 erhält. Diese kennt er aber ohnehin, der letzte Schritt ist also in diesem Kontext überflüssig.

Dementsprechend formulieren wir den Angreifer $\mathcal{A}^*$ mithilfe eines Angreifers $\mathcal{A}$, der das gerade beschriebene Spiel gewinnt. $\mathcal{A}^*$ geht in dem Spiel aus Definition 1.8 wie folgt vor:

1. Es werden ein Bit $b \in_R \{0, 1\}$ und Schlüssel $\mathsf{par}_0, \mathsf{par}_1 \in \mathsf{Par}$ gewählt. Diese Werte werden geheimgehalten und das Orakel $\mathcal{O}$ erhält sie als Eingabe.

2. $\mathcal{A}^*$ startet $\mathcal{A}$.

3. $\mathcal{A}$ wählt im ersten Schritt sowohl den Signaturschlüssel $(\mathsf{pk}, \mathsf{sk})$ als auch zwei Nachrichten m_0 und m_1, die nun auf seinem Ausgabe-Kommunikationsband stehen. Damit erhält $\mathcal{A}^*$ die Nachrichten m_0 und m_1.

4. $\mathcal{A}^*$ sendet (m_0, m_1) an $\mathcal{O}$.

5. $\mathcal{O}$ berechnet $\overline{m}_b := \varphi_{\mathsf{par}_b}(m_b)$ und $\overline{m}_{1-b} := \varphi_{\mathsf{par}_{1-b}}(m_{1-b})$ und sendet diese Werte zurück zu $\mathcal{A}^*$.

6. $\mathcal{A}^*$ führt (in der Rolle des Orakels aus Sicht von $\mathcal{A}$) die restlichen Schritte des Signaturalgorithmus' σ_I mit $\mathcal{A}$ durch und erhält die gültigen Signaturen $\sigma(\overline{m}_b) = \overline{s}_b$ und $\sigma(\overline{m}_{1-b}) = \overline{s}_{1-b}$.

7. $\mathcal{A}^*$ sendet okay zu $\mathcal{A}$, der ein Bit b' ausgibt.

8. Die Ausgabe von $\mathcal{A}^*$ ist b'.

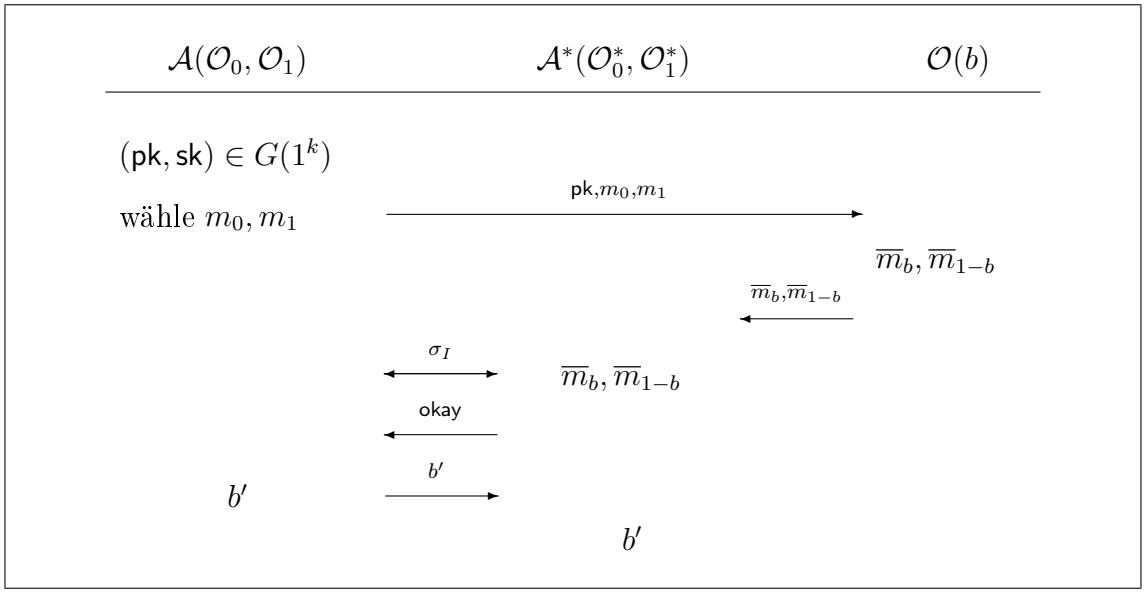


Abbildung 6.6: Beweis zu Satz 6.4

Verlangt $\mathcal{A}$ während des Spiels Zugriff auf seine Orakel $\mathcal{O}_0$ und $\mathcal{O}_1$, so kann $\mathcal{A}^*$ seine eigenen Orakel $\mathcal{O}_0^*$ und $\mathcal{O}_1^*$ verwenden, um korrekte Anworten auf die Anfragen gemäß dem oben beschriebenen Vorgehen zu erzeugen.

Da $\mathcal{A}$ ein effizienter Algorithmus ist, gilt dasselbe offensichtlich auch für $\mathcal{A}^*$. Ferner gelingt der Angriff von $\mathcal{A}^*$, wenn $\mathcal{A}$ das Bit $b' = b$ ausgibt. Dem Algorithmus $\mathcal{A}$ gelingt dies mit Wahrscheinlichkeit $\varepsilon(k)$, sodass auch $\mathcal{A}^*$ mit Wahrscheinlichkeit $\varepsilon(k)$ Erfolg hat. Ferner ist nach Voraussetzungn $1/2 - \varepsilon(k)$ nicht vernachlässigbar in k, sodass die Behauptung folgt. □

Die Aussage des Satzes zeigt einerseits eine Möglichkeit auf, rechnerisch blinde interaktive RSA-Signaturen zu konstruieren. Andererseits wird hier ein Instrumentarium bereitgestellt, um praktische Umsetzungen von perfekt blinden Signaturen bezüglich ihres Blindheitsgrades zu analysieren. Darauf werden wir im nächsten Abschnitt noch genauer eingehen.

Insgesamt haben wir bisher in diesem Kapitel sowohl für perfekt als auch für rechnerisch blinde Signaturen nach der vorgeschlagenen Konstruktion hinreichende und notwendige Bedingungen angegeben. Diese werden wir verwenden, um die Blindheit des im nächsten Abschnitt vorgestellten Beispiels einer interaktiven RSA-Signatur zu untersuchen.

6.3 Ein Beispiel

Nachdem wir in den letzten Abschnitten untersucht haben, unter welchen Bedingungen sich der am Anfang des Kapitels vorgestellte allgemeine Ansatz für die Konstruktion von interaktiven Signaturen im Fall der RSA-Signatur als blinde Signatur umsetzen lässt, betrachten wir in diesem Abschnitt die interaktive Signatur aus Satz 6.2 für eine gegebene Blendungsfunktion $\varphi_{\mathsf{par},z}$ mit $\mathsf{par} \in \mathsf{Par}$ und $z \in \mathcal{Z}$. Dazu werden wir zunächst zeigen, dass die Forderung nach einer interaktiven Signatur dazu führt, dass die zweite Blendungsfunktion $\psi_{\mathsf{par},z}$ eindeutig durch die Wahl von $\varphi_{\mathsf{par},z}$ bestimmt ist. Die Wahl der Funktion $\varphi_{\mathsf{par},z}$ ist motiviert durch die Homomorphieeigenschaft des RSA: Die Bedingung aus Satz 6.2 legt nahe, dass sich lineare Funktionen besonders gut als Blendungsfunktionen eignen.

Satz 6.5 (Die interaktive RSA-Signatur mit linearer Blendung) *Es sei* $\Sigma = (G, \sigma, V)$ *die RSA-Signatur. Für* $z \in \mathcal{Z}$ *und* $\mathsf{par} \in \mathsf{par}$ *sei die Blendungsfunktion* $\varphi_{\mathsf{par},z} : \mathbb{Z}_n \longrightarrow \mathbb{Z}_n$ *als*

$$\varphi_{\mathsf{par},z}(m) = \rho^e(\mathsf{par}, z) \cdot m \bmod n \tag{6.5}$$

für alle $m \in \mathbb{Z}_n$ *gegeben. Dabei sei* $\rho : \mathsf{Par} \times \mathcal{Z} \to \mathbb{Z}_n^*$ *eine effizient berechenbare Abbildung. Im Folgenden bezeichnen wir für alle* $\mathsf{par} \in \mathsf{Par}$ *mit* $(\rho(\mathsf{par}))^{-1}$ *die multiplikative Inverse von* $\rho(\mathsf{par}) \in \mathbb{Z}_n^*$. *Unter diesen Voraussetzungen gilt:*
$\Sigma_I = (G, \sigma_I, V_I)$ *mit* σ_I *und* V_I *gemäß Abbildung 6.3 bzw. 6.4 ist genau dann ein interaktives Signaturschema, wenn* $\psi_{\mathsf{par},z} : \mathbb{Z}_n \to \mathbb{Z}_n$ *als*

$$\psi_{\mathsf{par},z}(\overline{s}) = (\rho(\mathsf{par}, z))^{-1} \cdot \overline{s} \tag{6.6}$$

für alle $\overline{s} \in \mathbb{Z}_n$ *gegeben ist.*

Beweis. Sei zunächst Gleichung (6.6) erfüllt. Dann gilt für alle $m \in \mathbb{Z}_n$ die folgende Gleichungskette modulo n:

$$\begin{aligned} \psi_{\mathsf{par},z}(\varphi^d_{\mathsf{par},z}(m)) &= \psi_{\mathsf{par},z}(\rho(\mathsf{par},z) \cdot m^d) \\ &= (\rho(\mathsf{par},z))^{-1} \cdot \rho(\mathsf{par},z) \cdot m^d \\ &= m^d. \end{aligned}$$

Also ist Gleichung (6.3) erfüllt. Damit sind die Voraussetzungen von Satz 6.1 erfüllt, und die Behauptung folgt.

Sei nun $\Sigma_I = (G, \sigma_I, V_I)$ eine interaktive Signatur und $m \in \mathbb{Z}_n$ eine beliebige Nachricht. Aus der Durchführbarkeit des Protokolls σ_I folgt, dass bei korrektem Verhalten von $\mathcal{R}$ und $\mathcal{S}$ die Ausgabe von $\mathcal{R}$ eine gültige Signatur $\sigma(m) \in S(m) = \{m^d \bmod n\}$ auf die Nachricht m ist. Damit gilt modulo n die Gleichung

$$\psi^e_{\mathsf{par},z}(\varphi^d(\mathsf{par}, z; m)) = s^e = m,$$

und mit der Definition von $\varphi_{\mathsf{par},z}$ folgt modulo n die Gleichung

$$\psi_{\mathsf{par},z}(\rho(\mathsf{par},z) \cdot m^d) = m^d.$$

Mit $\overline{s} := \rho(\mathsf{par},z) \cdot m^d \in \mathbb{Z}_n$ folgt die Behauptung. □

Somit erhält man als Spezialfall der interaktiven Signatur aus Satz 6.2 die in Satz 6.5 vorgestellte interaktive Signatur. Bevor wir uns der Untersuchung der Sicherheit dieses Vorschlags zuwenden, werden wir zunächst anhand der Ergebnisse aus den vorangegangenen Abschnitten die Blindheitseigenschaften untersuchen. Es ist naheliegend, dass wir hier konkretere Angaben über die für perfekte oder rechnerische Blindheit notwendige Beschaffenheit der Blendungsfunktion $\varphi_{\mathsf{par},z}$ treffen können. Zuvor sei jedoch darauf hingewiesen, dass zwei Spezialfälle der interaktiven RSA-Signatur mit linearer Blendung bereits bekannt sind:

Bemerkung 6.1 *Die in Satz 6.5 angegebene interaktive Signatur fasst RSA-Signaturen von unterschiedlich starker Blindheit zusammen: Wählt man $\rho(\mathsf{par},z) \equiv 1$, so erhält man die ungeblendete RSA-Signatur, es werden lediglich die Kommunikationsschritte der beteiligten Parteien in der interaktiven Variante mit erfasst. Ist $\rho(\mathsf{par},z) = z$ und $z \in \mathcal{Z} = \mathbb{Z}_n^*$, so erhält man die von Chaum vorgeschlagene Variante ohne Hash-Funktion.*

Damit kommen wir zu einer genauen Formulierung der für die perfekte Blindheit benötigten Bedingungen an die vorgeschlagene interaktive Signatur:

Satz 6.6 *Es sei $\Sigma_I = (G, \sigma_I, V_I)$ die interaktive Signatur aus Satz 6.5. Ist die Zufallsvariable $\rho(\mathsf{par}, z)$ stochastisch unabhängig von m und gleichverteilt auf $\mathbb{Z}_n^*$, so ist Σ_I perfekt blind.*

Beweis. Nach Lemma 1.1 folgt aus der Definition von $\varphi(\mathsf{par}, z; \cdot)$ und aus der stochastischen Unabhängigkeit von $\rho(\mathsf{par}, z)$ und m die stochastische Unabhängigkeit von m und $\varphi(\mathsf{par}, z; m)$. □

Man beachte, dass die Wahl des Parameters par nur einmal für jedes System getroffen wird, sodass par nichts zur Unabhängigkeit von m und $\rho(\mathsf{par}, z)$ beiträgt. Hier ist einzig die adäquate Wahl von z entscheidend. Nach Satz 6.6 muss die Wahl von z aber nicht unbedingt gemäß einer Gleichverteilung geschehen, wenn die Funktion ρ derart ist, dass $\rho(\mathsf{par}, z)$ gleichverteilt ist. Dies bedeutet insbesondere, dass man beliebig verteilte Zufallszahlen für die Blendung verwenden kann, wenn man $\rho(\mathsf{par}, z)$ als Transformationen zu einer Gleichverteilung wählt. Solche Transformationen sind aus der Wahrscheinlichkeitsrechnung bekannt.

Ferner liefert Satz 6.5 offenbar auch für gleichverteilte $z \in \mathbb{Z}_n$ eine Vielzahl von perfekt blinden Varianten der interaktiven RSA-Signatur: Zum Beispiel kann man ggf. die Abbildung $\rho(\mathsf{par}, z) = e + z \mod n$ oder $\rho(\mathsf{par}, z) = m + z$ setzen, da der Parameter par für $\mathcal{R}$ frei wählbar ist. Eine andere Möglichkeit ist, dass der Empfänger den Parameter par einmal für alle Signaturprotokolle zufällig aus $\mathbb{Z}_n^*$ auswählt und eine Blendungsfunktion $\rho(\mathsf{par}, z) = \mathsf{par} \cdot z$ verwendet. Diese Varianten liefern nach Satz 6.5 noch immer eine perfekt blinde interaktive Signatur. Ebenso folgt aus Satz 6.6 sofort die bereits bekannte perfekte Blindheit der Signatur von Chaum.

Folgerung 6.1 *Die blinde interaktive RSA-Signatur nach Chaum ohne Hash-Funktion ist perfekt blind.*

Beweis. Die blinde interaktive RSA-Signatur ohne Hash-Funktion nach Chaum ist ein Spezialfall der Signatur Σ_I aus Satz 6.5: Wir wählen $\mathsf{Par} = \emptyset$ und $\mathcal{Z} = \mathbb{Z}_n^*$ sowie $\rho(\mathsf{par}, z) = z$. Das Protokoll gibt vor, dass z unabhängig von m gemäß einer Gleichverteilung aus $\mathbb{Z}_n^*$ ausgewählt wird, sodass die Behauptung nach Satz 6.6 gilt. □

Damit schließen wir die Untersuchung der Möglichkeiten, perfekte Blindheit zu erhalten, ab und wenden uns der rechnerischen Blindheit der interaktiven RSA-Signatur mit linearer Blendung zu. Hier können wir das Ergebnis aus Satz 6.4 direkt auf diese interaktive Signatur anwenden. Da wir in Satz 6.4 Blendungsfunktionen betrachtet haben, die sich als Chiffre mit Schlüsselraum Par auffassen lassen, gehen wir auch hier davon aus, dass $\mathcal{Z} = \emptyset$ ist.

Satz 6.7 *Es sei $\Sigma_I = (G, \sigma_I, V_I)$ die interaktive Signatur aus Satz 6.5 mit $\mathcal{Z} = \emptyset$ und $\rho : \mathsf{Par} \longrightarrow \mathbb{Z}_n^*$. Besitzt die Menge*

$$\mathcal{F} := \{\varphi_{\mathsf{par}} : \mathbb{Z}_n \to \mathbb{Z}_n \mid \varphi_{\mathsf{par}}(m) = \rho^e(\mathsf{par}) \cdot m \pmod{n};\ \mathsf{par} \in \mathsf{Par}\},$$

wobei e der mit G erzeugte öffentliche Schlüssel von $\mathcal{S}$ ist, die Eigenschaften einer unter einem Angriff auf zwei Schlüssel rechnerisch sicheren symmetrischen Chiffre gemäß Definition 1.8, so ist Σ_I rechnerisch blind.

Beweis. Die Behauptung folgt unmittelbar aus den Sätzen 6.5 und 6.4. □

Dieses Ergebnis lässt sich wie folgt interpretieren:

Bemerkung 6.2 *Die rechnerische Sicherheit von $\mathcal{F}$ hängt wesentlich von $\rho(\mathsf{par})$ ab. Im Allgemeinen wird man sich ρ als Pseudozufallszahlengenerator mit seed par vorstellen können. Damit stellt dieser Satz insbesondere einen Zusammenhang zwischen der Güte der Zufallszahlen und der Blindheit her: Ist der verwendete Pseudozufallszahlengenerator derart, dass er die polynomielle Ununterscheidbarkeit der Nachrichten und ihrer geblendeten Versionen im Sinne von Definition 1.8 sichert, so ist Σ_I rechnerisch blind. Insbesondere kann man die Wahl von z in der perfekt blinden Chaum-Signatur durch einen Pseudozufallszahlengenerator ersetzen. Eine solche Implementierung dieser Signatur in der Praxis gewährleistet eine rechnerisch blinde Signatur, falls der Pseudozufallszahlengenerator Zahlen generiert, die zu Blendungsfunktionen mit den geforderten Eigenschaften führen. Kurz gesagt liefert dieser Satz einen formalen Beweis für ein Ergebnis, das intuitiv klar ist: Der Grad der Blindheit hängt von der Güte des verwendeten Pseudozufallszahlengenerator ab. Darüber hinaus haben wir Bedingungen an den Zufallszahlengenerator formuliert, die einen gewissen Blindheitsgrad gewährleisten.*

Damit haben wir die Blindheitseigenschaften der verallgemeinerten RSA-Signatur sowohl in Bezug auf die perfekte als auch auf die rechnerische Blindheit untersucht und wenden uns nun den Sicherheitseigenschaften zu.

Betrachtet man die Sicherheit der durch Satz 6.5 gegebenen interaktiven Signatur, so stellt man zunächst fest, dass sich die Eigenschaften der RSA-Signatur auf die interaktive Signatur übertragen: Die interaktive RSA ist existentiell fälschbar unter einem Angriff ohne bekannte Signaturen und universell fälschbar unter einem Angriff mit gewählten Nachrichten. Wie im klassischen Fall kann aber durch die Verwendung einer Hash-Funktion ein ausreichendes Maß an Sicherheit erreicht werden. Da die Blendungsfunktionen in den Bereich des Empfängers einer Signatur

fallen und von ihm frei gewählt werden können, besitzt die verallgemeinerte interaktive RSA-Signatur mit Hash-Funktion für alle Blendungsfunktionen das gleiche Sicherheitsniveau. Dieses wurde für die blinde RSA-Signatur von Chaum in der Literatur bereits mit Hilfe des Random-Oracle-Modells untersucht, sodass wir uns hier dieser Frage nicht mehr widmen. Bezüglich der Blindheit der verallgemeinerten interaktiven RSA-Signatur mit Hash-Funktion ist zu bemerken, dass sich die gleichen Argumente wie oben auf das Bild der Hash-Funktion anwenden lassen, sodass der Blindheitsgrad sich auf das Bild der Hash-Funktion überträgt. Diese Ergebnisse sind noch einmal in der folgenden Bemerkung zusammengefasst.

Bemerkung 6.3 *Wir betrachten die interaktive RSA-Signatur Σ_I mit linearer Blendung.*

(a) *Man kann in der interaktiven Signatur genauso wie in der nicht interaktiven Variante eine Hash-Funktion $\mathcal{H} : \mathbb{Z} \to \mathbb{Z}_n$ verwenden: Dazu erhält der Signierer im Signaturprotokoll statt $\overline{m}$ den Wert $\overline{c} := \varphi_{\mathsf{par},z}(\mathcal{H}(m))$, und in der Verifikationsgleichung wird statt m das Bild der Hash-Funktion $\mathcal{H}(m)$ verwendet.*

(b) *Die Aussage von Satz 6.5 ist auch in diesem Fall richtig, wie man leicht verifiziert.*

(c) *Die verallgemeinerte interaktive RSA-Signatur mit Hash-Funktion besitzt bezüglich Unfälschbarkeit den gleichen Sicherheitsgrad wie die blinde RSA-Signatur von Chaum (vgl. Abschnitt 2.1).*

(d) *Unter den Voraussetzungen von Satz 6.6 bzw. Satz 6.7 erhält man zu jeder Nachricht $m \in \mathbb{Z}$ eine perfekt bzw. rechnerisch blinde Signatur auf den Hash-Wert $\mathcal{H}(m)$. Insbesondere gilt: Ersetzt man die Hash-Funktion durch ein Zufallsorakel, d.h. durch eine Abbildung, die ihre Bilder gemäß einer Gleichverteilung aus ihrem Bildbereich wählt, jedoch zu jedem Urbild stets das gleiche Bild liefert, so ist die interaktive RSA-Signatur mit Hash-Funktion perfekt blind.*

Insgesamt haben wir somit eine Vielzahl von Möglichkeiten aufgezeigt, die zu einer perfekt oder rechnerisch blinden interaktive RSA-Signatur führen. Insbesondere wurde festgestellt, dass sich der von Chaum vorgestellte Ansatz deutlich verallgemeinern lässt. Ferner wurden die Rahmenbedingungen für die Konstruktion von perfekt oder rechnerisch blinden interaktiven Signaturen genau beschrieben und somit die wesentlichen Eigenschaften der Blindheit herausgearbeitet.

6.4 Die interaktive Schnorr-Signatur

Zu keiner anderen Signatur gibt es so viele Varianten wie zur Schnorr-Signatur. Dies ist einerseits der Tatsache zu verdanken, dass dies eine recht effiziente Signatur ist, die sich gut in verschiedensten Anwendungen einsetzen lässt. Ein weiterer Grund ist der mögliche Einsatz auf beliebigen Gruppen, wie z.B. den durch elliptische Kurven erzeugte Gruppen. Die verschiedenen Varianten kann man nach ihrem unterschiedlichen Blindheitsgrad klassifizieren, so gibt es z.B. die schwach blinden Schnorr-Signaturen oder die perfekt blinden Schnorr-Signaturen. Wir werden in den folgenden Abschnitten sehen, dass all diese Varianten auf der gleichen interaktiven Signatur beruhen. In diesem Abschnitt wird zunächst untersucht, welche Gestalt die am Anfang des Kapitels vorgestellte Konstruktion im Falle der Schnorr-Signatur annimmt.

Um deutlich zu machen, auf welche Weise die angegebene Konstruktion zu einer interaktiven Schnorr-Signatur führt, betrachten wir zunächst eine vereinfachte Variante, die Schnorr-Signatur ohne Hash-Funktion für Nachrichten $m \in \mathcal{M} = \mathbb{Z}_q$. Der Schlüsselerzeugungsalgorithmus sei wie in der Schnorr-Signatur, d.h. bei Eingabe von 1^k erzeugt G ein Schlüsselpaar

$$(\mathsf{pk}, \mathsf{sk}) = ((p, q, g, y), x),$$

wobei p eine Primzahl der Länge k, q eine Primzahlen mit $q|p-1$, $g \in \mathbb{Z}_p$ der Ordnung q sowie $x \in_R \mathbb{Z}_q$ und $y = g^x \bmod p$ ist. Der Signatur- bzw. Verifikationsalgorithmus sei wie folgt:

Signaturalgorithmus. Bei Eingabe von m und dem oben angegebenen Schlüsselpaar besteht der Signaturalgorithmus aus folgenden Schritten:

1. Es wird ein $w \in_R \mathbb{Z}_q$ gewählt.
2. Man berechnet $r = g^w$ und $s = m \cdot x + w \bmod q$.
3. Die Ausgabe ist $\sigma(m) := (r, s)$.

Verifikation. V hat bei Eingabe von $(m, (r, s))$ und (p, q, g, y) die Ausgabe 1, falls die Gleichung

$$g^s = y^m \cdot r$$

gilt, und 0 sonst.

Bemerkung 6.4 *Man beachte, dass diese Signatur, im Gegensatz zu der Version mit Hash-Funktion, existentiell fälschbar unter einem Angriff ohne Nachrichten*

ist. Wir betrachten die Signatur trotzdem, da die Blendungsmechanismen in beiden Signaturen auf die gleiche Art und Weise implementiert werden können, die Hash-Funktion zunächst also für die Blendung der Signatur nicht ausschlaggebend ist. Wir werden allerdings im Weiteren Vorbereitungen treffen, die es ermöglichen, die Signatur mit Hash-Funktion zu verwenden.

Um die für unsere Konstruktion notwendigen Blendungsfunktionen näher zu beschreiben, bestimmen wir zunächst zu einer gegebenen Nachricht $m \in \mathbb{Z}_q$ die Menge $S(m)$ aller gültigen Signaturen auf m: Nach dem Signatur- bzw. Verifikationsalgorithmus gilt

$$\begin{aligned} S(m) &= \{(r,s) \mid r = g^w \bmod p,\ s = x \cdot m + w \bmod q,\ w \in \mathbb{Z}_q\} \\ &= \{(r,s) \mid r \in \mathbb{Z}_p,\ g^s = y^m \cdot r \bmod p\}. \end{aligned}$$

Seien für $\mathsf{par} \in \mathsf{Par}$ und $z \in \mathcal{Z}$ die effizient berechenbaren Funktionen $\varphi_{\mathsf{par},z}$ und $\psi_{\mathsf{par},z}$ gemäß (6.1) bzw. (6.2) gegeben. Dann nehmen die Protokolle σ_I und V_I aus den Abbildungen 6.1 und 6.2 die in Abbildung 6.7 bzw. 6.8 angegebene Gestalt an.

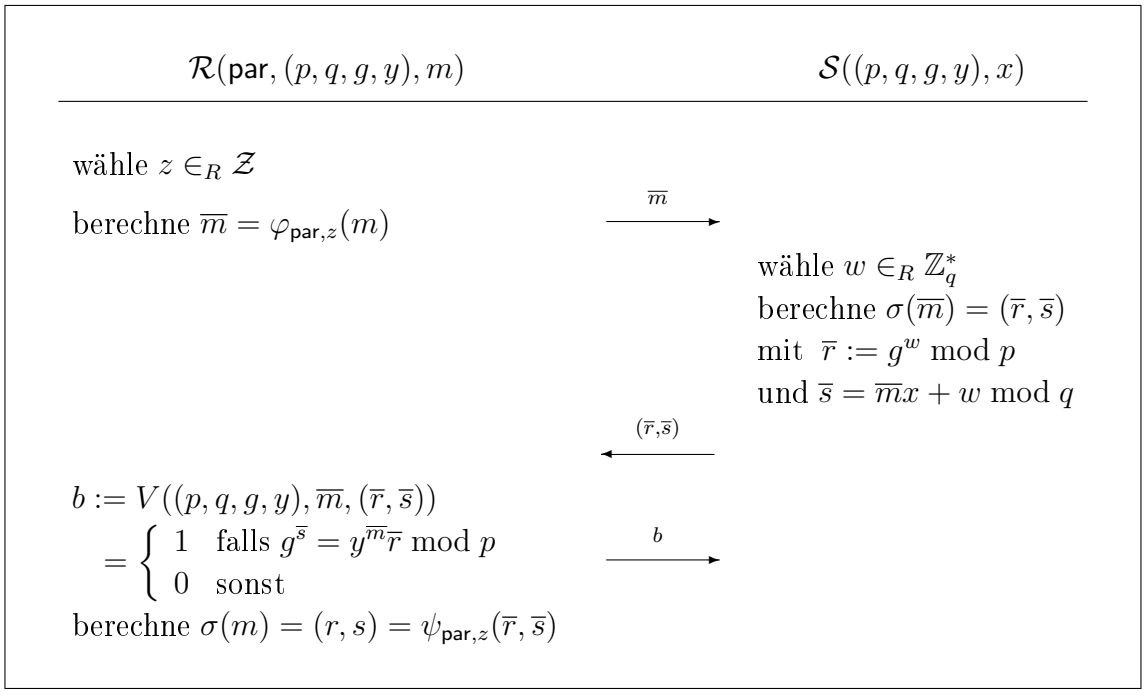


Abbildung 6.7: Signaturprotokoll σ_I

Nach Satz 6.1 ist $\Sigma_I = (G, \sigma_I, V_I)$ mit σ_I und V_I gemäß Abbildung 6.7 bzw. 6.8 eine interaktive Signatur, falls die Blendungsfunktionen $\varphi_{\mathsf{par},z}$ und $\psi_{\mathsf{par},z}$ für gegebene

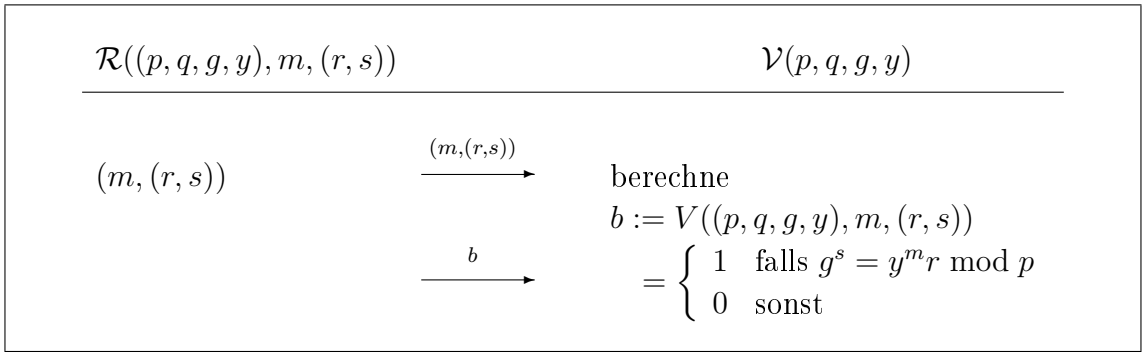


Abbildung 6.8: Verifikationsprotokoll V_I

$\mathsf{par} \in \mathsf{Par}$ und $z \in \mathcal{Z}$ und für alle $m \in \mathbb{Z}_q$ die Bedingung

$$\psi_{\mathsf{par},z}(\sigma(\varphi_{\mathsf{par},z}(m))) \in \{(r,s) \mid r \in \mathbb{Z}_p^*,\ g^s = y^m \cdot r \bmod p\} \tag{6.7}$$

erfüllen.

Dieses Ergebnis ist in dem folgenden Satz festgehalten.

Satz 6.8 *Es sei $\Sigma = (G, \sigma, V)$ die Schnorr-Signatur ohne Hash-Funktion. Ferner seien die Protokolle σ_I und V_I gemäß Abbildungen 6.7 und 6.8 sowie für alle $z \in \mathcal{Z}$ und $\mathsf{par} \in \mathsf{Par}$ effizient berechenbare Abbildungen $\varphi_{\mathsf{par},z}$ und $\psi_{\mathsf{par},z}$ gemäß (6.1) und (6.2) gegeben. Dann gilt:*
Ist Bedingung (6.7) erfüllt, so ist $\Sigma_I = (G, \sigma_I, V_I)$ eine interaktive Signatur.

Beweis. Die Behauptung folgt unmittelbar aus Satz 6.1 und den obigen Überlegungen. □

In den folgenden Betrachtungen beschränken wir uns auf Blendungsfunktionen $\psi_{\mathsf{par},z}$, die als

$$\psi_{\mathsf{par},z}(\overline{r}, \overline{s}) = (\psi_{1,\mathsf{par},z}(\overline{r}), \psi_{2,\mathsf{par},z}(\overline{s})) \tag{6.8}$$

für $\psi_{1,\mathsf{par},z} : \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ und $\psi_{2,\mathsf{par},z} : \mathbb{Z}_q \to \mathbb{Z}_q$ gegeben sind.

Dies tun wir aus den folgenden zwei Gründen:

1. Ein Ziel dieses Kapitels ist es, Bedingungen an $\varphi_{\mathsf{par},z}$ und $\psi_{\mathsf{par},z}$ anzugeben, unter denen die Blindheit der interaktiven Signatur gewährleistet ist. Dies ist für Blendungsfunktionen gemäß (6.7) jedoch nur schwer zu erreichen, da diese allgemeine Bedingung unterschiedlichste Blendungen gestattet.

2. Wie in Bemerkung 6.4 bereits festgehalten wurde, ist das Sicherheitsniveau der Schnorr-Signatur ohne Hash-Funktion nicht ausreichend. Schlussendlich muss also eine Variante der Signatur betrachtet werden, in der die Verwendung einer Hash-Funktion grundsätzlich möglich ist. Um den Angriff ohne Nachrichten auf die Schnorr-Signatur auszuschließen, ist es jedoch nötig, die Hash-Funktion nicht nur auf die zu unterzeichnende Nachricht m anzuwenden, sondern den Hash-Wert der Konkatenation von m und des Signaturparameters r zu berechnen. Bezogen auf eine Blendung bedeutet dies: Ein Empfänger muss den Wert r schon kennen, bevor er den Hash-Wert auf die Nachricht berechnet, und damit insbesondere, bevor er den Wert s kennt. Diese Überlegung motiviert die „getrennte“ Blendung von $\overline{r}$ und $\overline{s}$.

Wir halten fest, dass die Bedingung 6.7 für Funktionen $\psi_{\mathsf{par},z}$ gemäß (6.8) die folgende Gestalt annimmt: Für alle $m \in \mathbb{Z}_q$ und $(\overline{r}, \overline{s}) \in S\left(\varphi_{\mathsf{par},z}(m)\right)$ gilt

$$(\psi_{1,\mathsf{par},z}(\overline{r}), \psi_{2,\mathsf{par},z}(\overline{s})) \in \{(r,s) \mid r \in \mathbb{Z}_p^*,\ g^s = y^m \cdot r \bmod p\}. \tag{6.9}$$

Der zweite der oben genannten Gründe hat eine weitere Konsequenz: In dem in Abbildung 6.7 vorgestellten Signaturprotokoll muss die Reihenfolge der Kommunikationsschritte verändert werden, um eine interaktive Signatur zu erhalten, in der eine Hash-Funktion eingesetzt werden kann. Aus diesem Grund wenden wir uns einer zweiten Variante von interaktiven Schnorr-Signaturen zu, die in Abbildung 6.9 zu sehen ist. Auch wenn in dem dort vorgestellten Signaturprotokoll (noch) keine Hash-Funktion verwendet wird, erfüllt es doch die folgende Anforderung: Der Empfänger kann einen Teil der Signatur, nämlich r, schon berechnen, bevor der Signierer die zu signierende Nachricht erhält. Auf diese Weise kann man durch Ersetzen der Nachricht m durch $c := \mathcal{H}(r, m)$, wobei $\mathcal{H} : \mathbb{Z}_p \times \{0,1\}^* \to \mathbb{Z}_q$ eine Hash-Funktion ist, leicht eine interaktive Variante der Schnorr-Signatur mit Hash-Funktion angeben.

Da die Kommunikationsschritte gegenüber der am Anfang des Kapitels vorgestellten Konstruktion verändert wurden, können wir Satz 6.1 nicht anwenden, um zu zeigen, dass wir unter der Bedingung 6.9 eine interaktive Signatur erhalten. Dies ist aber trotzdem richtig, wie der nächste Satz zeigt.

Satz 6.9 *Sei $\Sigma = (G, \sigma, V)$ die Schnorr-Signatur ohne Hash-Funktion und zu $\mathsf{par} \in \mathsf{Par}$ sowie $z \in \mathcal{Z}$ Abbildungen $\varphi(\mathsf{par}, z; \cdot)$ gemäß (6.1) und $\psi(\mathsf{par}, z; \cdot)$ gemäß (6.9) gegeben. Dann ist $\Sigma_I = (G, \sigma_I, V_I)$ eine interaktive Signatur.*

Beweis. Wir zeigen, dass die Ausgabe von $\mathcal{V}$ im Verifikationsprotokoll okay ist, falls alle Teilnehmer den Protokollen σ_I und V_I folgen. Wegen der Durchführbarkeit von

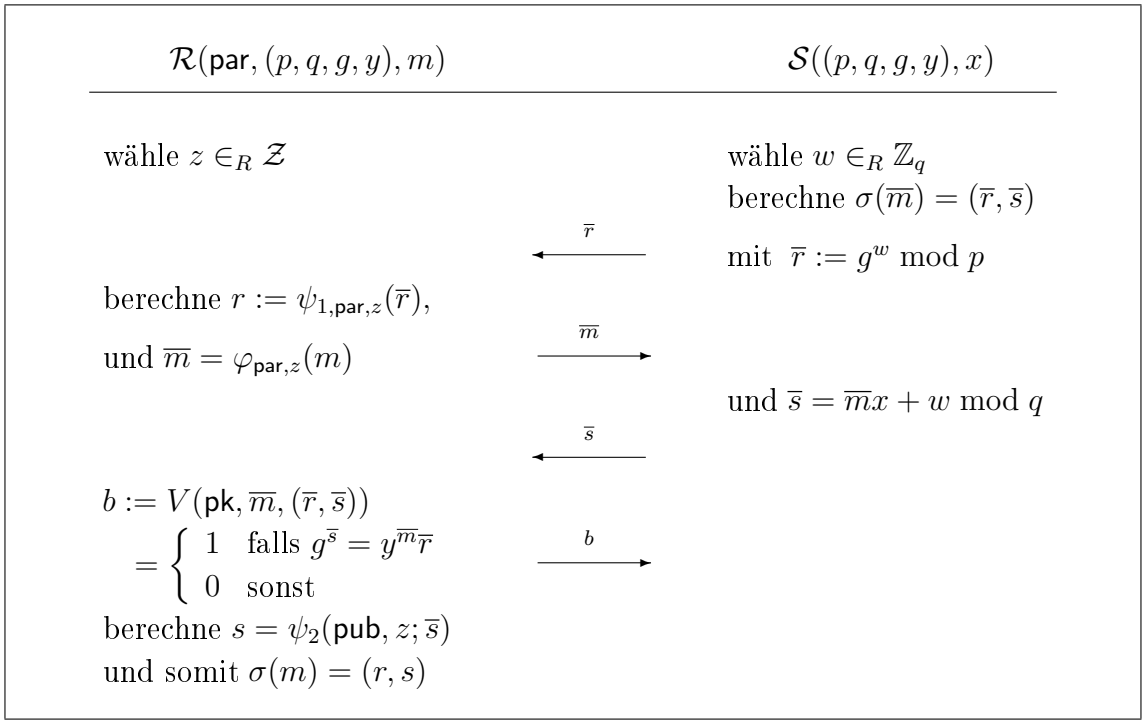


Abbildung 6.9: Signaturprotokoll σ_I

Σ gilt

$$V(\mathsf{pk},\overline{m},(\overline{r},\overline{s})) = 1,$$

sodass $\mathcal{S}$ im Signaturprotokoll okay ausgibt und $\mathcal{R}$ die Ausgabe $\sigma(m) = (r,s)$ hat. Wegen Gleichung (6.9) gilt

$$V(\mathsf{pk},m,(r,s)) = 1,$$

sodass $\mathcal{V}$ im Verifikationsprotokoll okay ausgibt. □

Somit können wir zwei verschiedene interaktive Schnorr-Signaturen aus der vereinfachten Variante der Schnorr-Signatur konstruieren. In den weiteren Überlegungen beziehen wir uns aus den oben genannten Gründen stets auf die letzte Variante.

6.5 Analyse der Blindheit

Das Ziel dieses Abschnitts ist es, in Anhängigkeit von den Blendungsfunktionen konkrete Bedingungen für die Blindheit der durch Satz 6.9 gegebenen interaktiven Schnorr-Signatur anzugeben. Wie bei der Untersuchung der RSA-Signatur unterscheiden wir auch hier Bedingungen, die zur perfekten, und solche, die zur rechnerischen Blindheit führen.

6.5.1 Perfekt blinde interaktive Schnorr-Signaturen

Wir betrachten die Fragestellung nach Bedingungen an die Blendungsfunktionen $\varphi_{\mathsf{par},z}$ und $\psi_{\mathsf{par},z}$, um perfekte Blindheit in der interaktiven Schnorr-Signatur aus Satz 6.9 zu erreichen. Im Gegensatz zu der RSA-Signatur genügt es hier nicht, die stochastische Unabhängigkeit von m und $\overline{m}$ zu fordern. Intuitiv ist dies darin begründet, dass die Signatur aus zwei Komponenten r und s besteht, wobei s durch r und m eindeutig bestimmt ist. Damit wird auch die Blendung von r eine gewisse Rolle spielen. Es stellt sich heraus, dass die Blendung von r sogar die gleiche Rolle spielt wie die von m. Genauer gesagt ist die Blendung von m in der RSA-Signatur im Falle der Schnorr-Signatur mit der Blendung des Paares (m, r) gleichzusetzen. Dieses Ergebnis wird im nächsten Satz formal festgehalten.

Satz 6.10 *Es sei Σ_I das interaktive Signaturschema zur Schnorr-Signatur aus Satz 6.9 gegeben. Unter der Voraussetzung, dass die Protokolle von Σ_I von einem ehrlichen Teilnehmer $\mathcal{R}$ durchgeführt werden, gilt:*

(a) *σ_I ist genau dann perfekt blind, wenn die Zufallsgrößen (m, r) und $(\overline{m}, \overline{r})$ stochastisch unabhängig sind.*

(b) *Sind die Zufallsgrößen (m, r) und $(\overline{m}, \overline{r})$ stochastisch unabhängig, so ist Σ_I perfekt blind.*

Beweis. Wir beginnen mit dem Beweis zu Teil (a).

Seien zunächst (m, r) und $(\overline{m}, \overline{r})$ stochastisch unabhängig. Wir zeigen, dass die Protokollansicht des Signierers

$$view_{\mathcal{S}}(\sigma_I) = (\overline{r}, \overline{m}, \overline{s})$$

und (m, r, s) stochastisch unabhängig sind, falls $\mathcal{R}$ den Protokollen folgt und $\mathcal{V}$ im Protokoll V_I die Ausgabe okay hat. In diesem Fall gilt:

$$g^{\overline{s}} = y^{\overline{m}}\overline{r},$$

also

$$\overline{s} = \overline{m} \log_g y + log_g \overline{r}$$

und genauso

$$s = \mathcal{H}(r, m) \log_g y + \log_g r.$$

Damit sind die Voraussetzungen von Lemma 1.2 erfüllt und somit gelten die Gleichungen $H(\overline{r}, \overline{m}, \overline{s}) = H(\overline{r}, \overline{m})$, $H(m, r, s) = H(m, r)$ und

$$H(\overline{r}, \overline{m}, \overline{s}, m, r, s) = H(\overline{r}, \overline{m}, r, m) = H(\overline{r}, \overline{m}) + H(r, m),$$

wobei das letzte Gleichheitszeichen wegen der Unabhängigkeit von (m, r) und $(\overline{m}, \overline{r})$ gilt. Zusammen folgt

$$\begin{aligned} \mathcal{I}[view_{\mathcal{S}}(\sigma_I), (m, \sigma(m))] &= \mathcal{I}[(\overline{r}, \overline{m}, \overline{s}), (m, r, s)] \\ &= H(\overline{r}, \overline{m}, \overline{s}) + H(m, r, s) - H((\overline{r}, \overline{m}, \overline{s}), (m, r, s)) \\ &= H(\overline{r}, \overline{m}) + H(m, r) - H(\overline{r}, \overline{m}, m, r) = 0, \end{aligned}$$

d.h. σ_I ist perfekt blind.

Sei nun umgekehrt σ_I perfekt blind. Dann gilt unter den oben genannten Voraussetzungen an $\mathcal{R}$ und $\mathcal{V}$:

$$H(\overline{r}, \overline{m}, \overline{s}) + H(m, r, s) = H((\overline{r}, \overline{m}, \overline{s}), (m, r, s))$$

und mit Lemma 1.2 folgt die Behauptung.

Die Aussage von Teil (b) ist eine Folgerung aus (a) und Satz 4.2, wobei zu beachten ist, dass die allgemeine Konstruktion Voraussetzung (a) aus Satz 4.1 erfüllt. □

Damit haben wir eine hinreichende und notwendige Bedingung für die perfekte Blindheit des Signaturprotokolls einer interaktiven Schnorr-Signatur gefunden und wenden uns nun der rechnerischen Blindheit zu.

6.5.2 Rechnerisch blinde interaktive Schnorr-Signaturen

Auch hier fassen wir die Blendungsfunktion als symmetrische Chiffre auf, um die Beschreibung der rechnerischen Blindheit in Schnorr-Signaturen umzusetzen. Im Gegensatz zur RSA-Signatur kann man hier keine so umfassende Aussage machen, sondern nur eine notwendige Bedinung für die rechnerische Blindheit des Signaturprotokolls angeben.

Satz 6.11 *Es sei das interaktive Signaturschema Σ_I aus Satz 6.9 mit $\mathcal{Z} = \emptyset$ gegeben. Ist zusätzlich zu den dort angegebene Bedingungen*

$$\varphi_{\mathsf{par}} \in \mathcal{F} := \{\varphi_{\mathsf{par}} : \mathbb{Z}_q \to \mathbb{Z}_q \mid \mathsf{par} \in \mathsf{Par}\}$$

und $\mathcal{F}$ eine Chiffre, so gilt unter der Voraussetzung, dass die Protokolle von Σ_I von einem ehrlichen Teilnehmer $\mathcal{R}$ durchgeführt werden:
Ist σ_I rechnerisch blind unter einem Angriff auf zwei Schlüssel gemäß Definition 1.8, so ist $\mathcal{F}$ rechnerisch sicher unter einem Angriff mit gewählten Nachrichten.

Beweis. Der Beweis ist analog zu dem der entsprechenden Aussage aus Satz 6.4. □

Der Satz zeigt, dass eine rechnerisch sichere Blendung von m zwar notwendig, aber keineswegs hinreichend für die Blindheit des Signaturprotokolls ist. Dies liegt gerade an dem bei der perfekten Blindheit schon festgestellten Einfluss der Blendung des Parameters r: Hier kommt eine zweite Blendungsfunktion ins Spiel, die eigentlich parallel angegriffen werden muss. Andererseits kann er in Spiel 5.2 die Werte $\overline{r}_b$ und r_b durchaus zuordnen, da sie beide der Protokolldurchführung mit dem gleichen Empfänger $\mathcal{R}_b$ entstammen. Dies alles führt dazu, dass sich die Eigenschaften hier nicht mehr in eine so schöne Form bringen lassen, wie dies bei der RSA-Signatur der Fall war.

6.6 Ein Beispiel

Wie schon bei der RSA-Signatur wollen wir zum Schluss konkrete Blendungsfunktionen $\varphi_{\mathsf{par},z}$ und $\psi_{\mathsf{par},z}$ betrachten. Hier haben wir, im Gegensatz zur RSA-Signatur, drei Blendungsfunktionen, nämlich eine für die Nachricht m, eine für den Signaturparameter r und eine für den zweiten Signaturparameter s. Dabei ist zu beachten, dass die Blendungen für m und s modulo q und die Blendung von r modulo p durchgeführt werden, da die geblendeten Werte wiederum eine Signatur darstellen sollten.

Die Aussage des folgenden Satzes macht noch einmal den Einfluss des Parameters r auf das Blendungsverhalten der interaktiven Signatur deutlich: In einer interaktiven Signatur legt die Blendung von m und die von einem der beiden Signaturparameter die Blendung des dritten Parameters eindeutig fest. Insbesondere ergibt sich die Blendung von s, wenn man sich die von m und r vorgibt. Wie bei der interaktiven RSA-Signatur ist die Wahl der Blendungsfunktion für m bei den folgenden Überlegungen durch Bedingung 6.9 motiviert.

Man beachte, dass der folgende Satz eine interaktive Schnorr-Signatur liefert, die verschiedene mögliche Blendungen zusammenfasst. Dies werden wir später noch genauer diskutieren.

Satz 6.12 Die interaktive Schnorr-Signatur mit linearer Blendung *Es sei* $\Sigma = (G, \sigma, V)$ *die Schnorr-Signatur ohne Hash-Funktion. Für alle* $\mathsf{par} \in \mathsf{Par}$ *und* $z \in \mathcal{Z}$ *seien effizient berechenbare Abbildungen* $\rho_1 : \mathsf{Par} \times \mathcal{Z} \to \mathbb{Z}_q^*$, $\rho_2 : \mathsf{Par} \times \mathcal{Z} \to \mathbb{Z}_q$ *und* $\rho_3 : \mathsf{Par} \times \mathcal{Z} \to \mathbb{Z}_q$ *und für* $z \in \mathcal{Z}$ *und* $\mathsf{par} \in \mathsf{Par}$ *sei die Blendungsfunktion*

$\varphi_{\mathsf{par},z} : \mathbb{Z}_q \longrightarrow \mathbb{Z}_q$ *als*

$$\varphi_{\mathsf{par},z}(m) = \rho_1(\mathsf{par}, z) \cdot m + \rho_2(\mathsf{par}, z) \bmod q \tag{6.10}$$

für alle $m \in \mathbb{Z}_q$ *gegeben. Ferner sei die Blendungsfunktion* $\psi(\mathsf{pub}, z) : \mathbb{Z}_p \times \mathbb{Z}_q \to \mathbb{Z}_p \times \mathbb{Z}_q$ *als*

$$\psi_{\mathsf{par},z}(\overline{r}, \overline{s}) = (\psi_{1,\mathsf{par},z}(\overline{r}), \psi_{2,\mathsf{par},z}(\overline{s}))$$

für $\psi_{1,\mathsf{par},z} : \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ *und* $\psi_{2,\mathsf{par},z} : \mathbb{Z}_q \to \mathbb{Z}_q$ *vorausgesetzt. Weiterhin sei* σ_I *wie in Abbildung 6.7 oder 6.9 und* V_I *wie in Abbildung 6.8 beschrieben. Im Folgenden bezeichne* $(\rho_1(\mathsf{par}, z))^{-1}$ *die multiplikative Inverse von* $\rho_1(\mathsf{par}, z)$ *modulo* q.

Unter diesen Voraussetzungen gilt:

(a) *Sind für* $\mathsf{par} \in \mathsf{Par}$ *und* $z \in \mathcal{Z}$ *die Abbildungen* $\psi_{1,\mathsf{par},z}$ *und* $\psi_{2,\mathsf{par},z}$ *für alle* $\overline{r} \in \mathbb{Z}_p^*$ *und* $\overline{s} \in \mathbb{Z}_q$ *als*

$$\psi_{1,\mathsf{pub},\mathsf{z}}(\overline{r}) = \left(y^{\rho_2(\mathsf{par},z)}\overline{r}\right)^{(\rho_1(\mathsf{par},z))^{-1}} g^{\rho_3(\mathsf{par},z)} \bmod p \tag{6.11}$$

und

$$\psi_{2,\mathsf{par},z}(\overline{s}) = (\rho_1(\mathsf{par}, z))^{-1}\overline{s} + \rho_3(\mathsf{par}, z) \bmod q \tag{6.12}$$

gegeben, so ist $\Sigma_I = (G, \sigma_I, V_I)$ *eine interaktive Signatur.*

(b) *Ist* Σ_I *eine interaktive Signatur und ist* $\psi_{1,\mathsf{par},z}$ *für* $\mathsf{par} \in \mathsf{Par}$ *und* $z \in \mathcal{Z}$ *gemäß (6.11) gegeben, so ist* $\psi_{2,\mathsf{par},z}$ *eindeutig gemäß (6.12) bestimmt.*

(c) *Ist* Σ_I *eine interaktive Signatur und ist* $\psi_{2,\mathsf{par},z}$ *für* $\mathsf{par} \in \mathsf{Par}$ *und* $z \in \mathcal{Z}$ *gemäß (6.12) gegeben, so ist* $\psi_{1,\mathsf{par},z}$ *eindeutig gemäß (6.11) bestimmt.*

Beweis. Zunächst halten wir fest, dass die Reihenfolge der Kommunikationsschritte für die folgenden Überlegungen keine Rolle spielt, sodass sie gleichermaßen für das Signaturprotokoll aus Abbildung 6.7 wie für das aus Abbildung 6.9 richtig sind.

Wir beginnen mit dem Beweis zu Teil (a). Seien dazu die Gleichungen (6.11) und (6.12) erfüllt. Wir zeigen, dass Bedingung (6.7) bzw. (6.9) erfüllt ist, dass also für alle $m \in \mathbb{Z}_q$ gilt:

$$\psi_{\mathsf{par},z}(\overline{r}, \overline{s}) \in S(m).$$

Sei dazu $m \in \mathbb{Z}_q$. Dann ist für alle $\overline{s} \in \mathbb{Z}_p$ die folgende Gleichungskette modulo q richtig.

$$\begin{aligned}
\psi_{2,\mathsf{par},z}(\overline{s}) &= (\rho_1(\mathsf{par}, z))^{-1}\overline{s} + \rho_3(\mathsf{par}, z) \\
&= (\rho_1(\mathsf{par}, z))^{-1}(\overline{m}x + w) + \rho_3(\mathsf{par}, z) \\
&= (\rho_1(\mathsf{par}, z))^{-1}\left((\rho_1(\mathsf{par}, z) \cdot m + \rho_2(\mathsf{par}, z)) \cdot x + w\right) + \rho_3(\mathsf{par}, z) \\
&= mx + (\rho_1(\mathsf{par}, z))^{-1}\rho_2(\mathsf{par}, z)x + (\rho_1(\mathsf{par}, z))^{-1}w + \rho_3(\mathsf{par}, z).
\end{aligned}$$

Damit gelten modulo p die Gleichungen

$$\begin{aligned} g^{\psi_{2,\mathsf{par},z}(\overline{s})} &= g^{mx+(\rho_1(\mathsf{par},z))^{-1}\rho_2(\mathsf{par},z)x+(\rho_1(\mathsf{par},z))^{-1}w+\rho_3(\mathsf{par},z)} \\ &= y^m \cdot \left(g^{\rho_2(\mathsf{par},z)x+w}\right)^{(\rho_1(\mathsf{par},z))^{-1}} g^{\rho_3(\mathsf{par},z)} \\ &= y^m \cdot \left(y^{\rho_2(\mathsf{par},z)}\overline{r}\right)^{(\rho_1(\mathsf{par},z))^{-1}} g^{\rho_3(\mathsf{par},z)} \\ &= y^m \cdot \psi_{1,\mathsf{par},z}(\overline{r}), \end{aligned}$$

und die Behauptung folgt mit Satz 6.8, falls σ_I gemäß Abbildung 6.7, und mit Satz 6.9, falls σ_I gemäß Abbildung 6.9 gegeben ist.

Wir kommen zu dem Beweis von Teil (b): Sei nun Σ_I ein interaktives Signaturschema und sei $\psi_{1,\mathsf{par},z}$ für ein $\mathsf{par} \in \mathsf{Par}$ und ein $z \in \mathcal{Z}$ gemäß (6.11) gegeben. Zunächst folgt aus der Durchführbarkeit von σ_I, dass für alle $(\overline{r}, \overline{s}) \in S\left(\varphi_{\mathsf{par},z}(m)\right)$ die Gleichung

$$g^{\psi_{2,\mathsf{par},z}(\overline{s})} = y^m \cdot \psi_{1,\mathsf{par},z}(\overline{r}) \bmod p$$

gilt. Wegen (6.11) und $y, \overline{r}, g \in \mathbb{Z}_p^*$ folgt

$$y^m = g^{\psi_{2,\mathsf{par},z}(\overline{s})} \cdot \left(y^{\rho_2(\mathsf{par},z)}\overline{r}\right)^{-(\rho_1(\mathsf{par},z))^{-1}} g^{-\rho_3(\mathsf{par},z)} \bmod p.$$

Ferner gilt $(\overline{r}, \overline{s}) \in S\left(\varphi_{\mathsf{par},z}(m)\right)$, und somit sind folgende Gleichungen modulo p erfüllt.

$$\begin{aligned} g^{\overline{s}} &= y^{\varphi_{\mathsf{par},z}(m)} \cdot \overline{r} \\ &= y^{\rho_1(\mathsf{par},z)m+\rho_2(\mathsf{par},z)} \cdot \overline{r} \\ &= (y^m)^{\rho_1(\mathsf{par},z)} \cdot y^{\rho_2(\mathsf{par},z)} \cdot \overline{r} \\ &= \left(g^{\psi_2(\mathsf{par},z,\overline{s})} \cdot \left(y^{\rho_2(\mathsf{par},z)}\overline{r}\right)^{-(\rho_1(\mathsf{par},z))^{-1}} g^{-\rho_3(\mathsf{par},z)}\right)^{\rho_1(\mathsf{par},z)} \cdot y^{\rho_2(\mathsf{par},z)} \cdot \overline{r} \\ &= g^{\psi_{2,\mathsf{par},z}(\overline{s})\cdot\rho_1(\mathsf{par},z)} \cdot g^{-\rho_3(\mathsf{par},z)\rho_1(\mathsf{par},z)}. \end{aligned}$$

Damit ergibt sich:

$$\overline{s} = \psi_{2,\mathsf{par},z}(\overline{s}) \cdot \rho_1(\mathsf{par}, z) - \rho_3(\mathsf{par}, z)\rho_1(\mathsf{par}, z) \bmod q$$

und somit durch Umstellen nach $\psi_{2,\mathsf{par},z}(\overline{s})$ die Behauptung. Dazu beachte man, dass $\rho_1(\mathsf{par}, z)$ modulo q als multiplikativ invertierbar vorausgesetzt war.

Analog zeigt man die Behauptung aus Teil (c) des Satzes: Sei dazu $\psi_2(\mathsf{par}, z)$ gemäß (6.12) gegeben und Σ_I eine interaktive Signatur. Zunächst gilt für alle $(\overline{r}, \overline{s}) \in S\left(\varphi_{\mathsf{par},z}(m)\right)$ die Gleichung

$$g^{\overline{s}} = y^{\varphi_{\mathsf{par},z}(m)} \cdot \overline{r} = y^{\rho_1(\mathsf{par},z)m+\rho_2(\mathsf{par},z)} \cdot \overline{r} \pmod p.$$

Dies impliziert zusammen mit der Durchführbarkeit von σ_I für alle $(\overline{r}, \overline{s}) \in S\left(\varphi_{\mathsf{par},z}(m)\right)$ die Richtigkeit der folgenden Gleichungskette modulo p:

$$\begin{aligned}
\psi_{1,\mathsf{par},z}(\overline{r}) &= \left(g^{\overline{s}}\right)^{(\rho_1(\mathsf{par},z))^{-1}} g^{\rho_3(\mathsf{par},z)} y^{-m} \\
&= \left(y^{\rho_1(\mathsf{par},z)m+\rho_2(\mathsf{par},z)} \cdot \overline{r}\right)^{(\rho_1(\mathsf{par},z))^{-1}} g^{\rho_3(\mathsf{par},z)} y^{-m} \\
&= \left(y^{\rho_2(\mathsf{par},z)} \cdot \overline{r}\right)^{(\rho_1(\mathsf{par},z))^{-1}} g^{\rho_3(\mathsf{par},z)}.
\end{aligned}$$

Damit ist auch (c) gezeigt. □

Damit haben wir im Falle der in Satz 6.12 vorgegebenen Blendungsfunktion $\varphi_{\mathsf{par},z}$ eine verschärfte notwendige Bedingung an die interaktive Schnorr-Signatur aus Satz 6.8 gefunden. Eine hinreichende Bedingung lässt sich jedoch nicht formulieren, ohne sich auf eine der beiden Blendungsfunktionen $\psi_{1,\mathsf{par},z}$ oder $\psi_{2,\mathsf{par},z}$ festzulegen. Anders ausgedrückt, erhält man zu einer einzigen Blendungsfunktion $\varphi_{\mathsf{par},z}$ eine Vielzahl von möglichen Blendungen der beiden Signaturparameter. Dies spiegelt sich in der Vielzahl der Varianten der Schnorr-Signatur wider, die in der Literatur bekannt sind. Dazu die folgende

Bemerkung 6.5 *Wie bei der RSA-Signatur fasst die in Satz 6.12 angegebene interaktive Signatur Schnorr-Signaturen unterschiedlicher Blindheit zusammen. Dazu gehört insbesondere die ungeblendete Schnorr-Signatur (für $\rho_1(\mathsf{par}, z) \equiv 1$ und $\rho_2(\mathsf{par}, z) \equiv 0$), aber auch die sogenannten schwach blinden und die perfekt blinden Schnorr-Signaturen ohne Hash-Funktionen in verschiedenen Varianten.*

Damit haben wir unter Verwendung der oben aufgeführten Blendungsfunktion interaktive Schnorr-Signaturen konstruiert. Wie im Fall der RSA-Signatur interessiert auch hier, welche Eigenschaften eine rechnerisch oder perfekt blinde Variante besitzen muss, sodass wir uns der Untersuchung der Blindheit zuwenden. Wir beginnen mit der perfekten Blindheit. Der folgende Satz zeigt insbesondere, dass es verschiedene Möglichkeiten gibt, die perfekte Blindheit einer interaktiven Schnorr-Signatur zu erreichen:

Satz 6.13 *Es seien Σ_I, φ und ψ wie in Satz 6.12, und es seien die Zufallsvariablen $\rho_1(\mathsf{par}, z)$, $\rho_2(\mathsf{par}, z)$ und $\rho_3(\mathsf{par}, z)$ unabhängig und gemäß einer Gleichverteilung aus $\mathbb{Z}_q$ gewählt. Dann gilt:*

(a) *Ist $\rho_2(\mathsf{par}, z) \equiv 0$ und ist die Zufallsgröße $(\rho_1(\mathsf{par}, z), \rho_3(\mathsf{par}, z))$ stochastisch unabhängig von (m, r), so ist Σ_I perfekt blind.*

(b) *Ist $\rho_1(\mathsf{par}, z) \equiv 1$ und ist die Zufallsgröße $(\rho_2(\mathsf{par}, z), \rho_3(\mathsf{par}, z))$ stochastisch unabhängig von (m, r), so ist Σ_I perfekt blind.*

Beweis. Wir zeigen zuerst Teil (a) und verwenden dazu Satz 1.1. Zunächst halten wir fest, dass m gleichverteilt in $\mathbb{Z}_q$ und r gleichverteilt auf $G :=< g >$ ist. Da g von der Ordnung q ist, besitzt G ebenfalls q Elemente. (m, r) ist demnach auf der Menge $\mathcal{T}_1 := \mathbb{Z}_q \times G$ verteilt, die q^2 Elemente besitzt. Ebenso ist $(\rho_1(\mathsf{par}, z), \rho_3(\mathsf{par}, z))$ gleichverteilt auf $\mathcal{T}_2 = \mathbb{Z}_q^2$, also ebenfalls auf einer Menge der Mächtigkeit q^2. Wir setzen $X := (m, r)$ und $Y := (\rho_1(\mathsf{par}, z), \rho_3(\mathsf{par}, z))$. Unter den Voraussetzungen von (a) gilt

$$\overline{m} = \varphi_{\mathsf{par},z}(m) = \rho_1(\mathsf{par}, z) \cdot m \pmod{q}$$

und

$$r = \psi_{1,\mathsf{par},z}(\overline{r}) = \overline{r}^{(\rho_1(\mathsf{par},z))^{-1}} g^{\rho_3(\mathsf{par},z)} \pmod{p},$$

und somit lässt sich jede Realisierung von $(\overline{m}, \overline{r})$ für alle $y \in \mathcal{T}_2$ als Funktion $f_y : \mathcal{T}_1 \times \mathcal{T}_2 \longrightarrow \mathcal{T}_3$ mit $\mathcal{T}_3 = \mathbb{Z}_q \times G$ schreiben. Für die Zufallsgröße $(\overline{m}, \overline{r})$ gilt

$$(\overline{m}, \overline{r}) = f_Y(X).$$

Da es zu jedem $y \in \mathcal{T}_2$ und jedem $x \in \mathcal{T}_1$ genau ein Paar $\overline{m}, \overline{r}$ mit $f_y(x) = (\overline{m}, \overline{r})$ gibt, ist f_y invertierbar, sodass wir Lemma 1.1 anwenden können. Damit folgt die Unabhängigkeit von $X = (m, r)$ und $(\overline{m}, \overline{r})$.

Der Beweis zu Teil (b) kann analog geführt werden. □

Wie bei der interaktiven RSA-Signatur kann man das Ergebnis von Satz 6.13 in dem Sinne interpretieren, dass für eine perfekte Blendung grundsätzlich keine Gleichverteilung der Blendungsvariablen notwendig ist, solange die Abbildungen ρ_1, ρ_2 bzw. ρ_3 geeignet gewählt sind. Ferner deutet die Aussage des Satzes bereits darauf hin, dass es verschiedene Umsetzungen einer perfekt blinden interaktiven Schnorr-Signatur geben kann. In Satz 6.13 sind nur zwei davon aufgeführt. Diese stellen die in der Literatur am häufigsten verwendeten Blendungsvarianten dar. Dementsprechend gilt die

Folgerung 6.2 *Die blinde interaktive Schnorr-Signatur ohne Hash-Funktion ist perfekt blind.*

Beweis. Die blinde interaktive Schnorr-Signatur ist ein Spezialfall der interaktiven Signatur Σ_I aus Satz 6.12: Wir wählen $\mathsf{Par} = \emptyset$ und $\mathcal{Z} = \mathbb{Z}_q \times \mathbb{Z}_q$ sowie für $(u, v) \in \mathcal{Z}$ die Abbildungen $\rho_1((u, v)) := u$, $\rho_2((u, v)) \equiv 0$ und $\rho_3((u, v)) := v$. Das Protokoll gibt vor, dass $\mathcal{R}$ die Blendungsfaktoren (u, v) gemäß einer Gleichverteilung aus $\mathbb{Z}_q \times \mathbb{Z}_q$ wählt, sodass Folgerung 6.13 die Richtigkeit der Behauptung impliziert. □

Man beachte, dass es in Satz 6.10 im Gegensatz zu der Aussage aus Satz 6.3 nicht genügt, die Unabhängigkeit von m und $\overline{m}$ zu fordern. Ist nur diese Bedingung für eine interaktive Schnorr-Signatur erfüllt, sind aber $(\overline{m}, \overline{r})$ und (m, r) nicht stochastisch unabhängig, so sprechen wir von einem **schwach blinden** Signatur-Schema. Hinter dieser Bezeichnung steckt der Gedanke, dass ein Signierer durch das Signaturprotokoll keinerlei Vorteil hat, die ungeblendete Signatur zu erraten, er erkennt sie jedoch sofort wieder, wenn er eine Protokollmitschrift des Signaturprotokolls erhält. Dies soll die im Folgenden vorgestellte interaktive Signatur $\Sigma_I = (G, \sigma_I, V_I)$ mit σ_I und V_I wie in Abbildung 6.10 bzw. 6.8 zur Schnorr-Signatur ohne Hash-Funktion illustrieren.

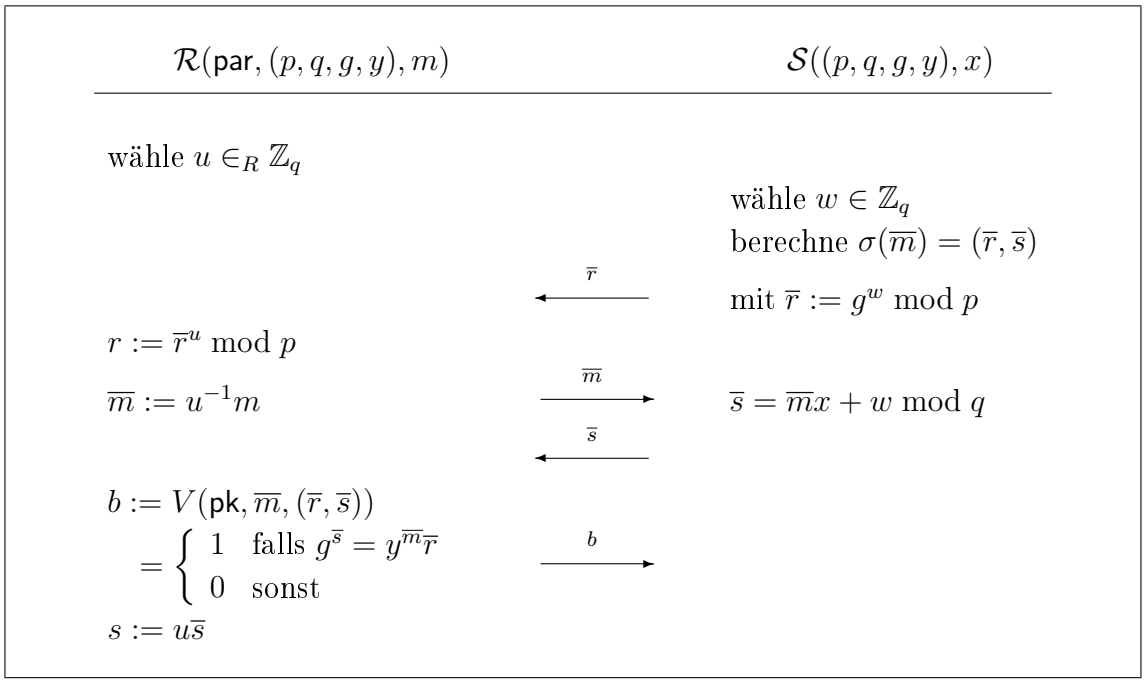


Abbildung 6.10: Signaturprotokoll der schwach blinden Schnorr-Signatur

Folgerung 6.3 *Seien σ_I und V_I wie in Abbildung 6.10 bzw. 6.8. Dann ist $\Sigma_I = (G, \sigma_I, V_I)$ eine interaktive Signatur gemäß Satz 6.9 zur Schnorr-Signatur ohne Hash-Funktion, die nicht perfekt blind ist.*

Beweis. (G, σ_I, V_I) ist eine interaktive Signatur gemäß Satz 6.9: Man wählt $\mathsf{Par} = \emptyset$ und $\mathcal{Z} = \mathbb{Z}_q$ sowie für $u \in \mathbb{Z}_q$ die Abbildungen $\varphi(\mathsf{par}, u; m) = \frac{m}{u}$ und $\psi_1(\mathsf{par}, z; \overline{r}) = \overline{r}^u$. Dann gilt:

$$r = \overline{r}^{\frac{\overline{m}}{m}},$$

sodass r sogar deterministisch von $\overline{r}$, m und $\overline{m}$ abhängt. □

Damit haben wir die verschiedenen Stufen der Blindheit durch „perfekte“ Blendung diskutiert. Im Anschluss an die Überlegungen bzgl. der Sicherheitseigenschaften wird eine Einordnung der aus der Literatur bekannten Signaturen stattfinden. Zunächst wenden wir uns jedoch der rechnerischen Blindheit zu. Hier liegt lediglich eine notwendige Bedingung für interaktive Schnorr-Signaturen vor, die wir der Vollständigkeit halber noch einmal für das betrachtete Beispiel aus Satz 6.12 angeben.

Folgerung 6.4 *Es sei $\Sigma_I = (G, \sigma_I, V_I)$ die interaktive Signatur aus Satz 6.12 mit $\mathcal{Z} = \emptyset$, $\rho_1 : \mathsf{Par} \to \mathbb{Z}_q^*$ und $\rho_2 : \mathsf{Par} \to \mathbb{Z}_q$. Ist Σ_I rechnerisch blind, so besitzt die Menge*

$$\mathcal{F} := \{\varphi_{\mathsf{par}} : \mathbb{Z}_q \to \mathbb{Z}_q \mid \mathsf{par} \in \mathsf{Par}\},$$

mit φ_{par} wie in Satz 6.12, die Eigenschaften einer gemäß Definition 1.7 gegen einen Angriff auf zwei Schlüssel rechnerisch sicheren symmetrischen Chiffre.

Beweis. Die Behauptung folgt unmittelbar aus den Sätzen 6.12 und 6.11. □

Wie bei der RSA-Signatur bereits erwähnt, ist es sinnvoll, die Abbildungen ρ_1 und ρ_2, die obige Menge definieren, als Pseudozufallszahlengeneratoren zu interpretieren. Damit liefert der obige Satz eine notwendige Bedingung für die Beschaffenheit des in der Signatur verwendeten Zufallszahlengenerators.

Somit kommen wir zu der Sicherheit der interaktiven Schnorr-Signatur mit linearer Blendung. Es wurde bereits mehrfach erwähnt, dass die Schnorr-Signatur ohne Hash-Funktion und somit auch die interaktive Schnorr-Signatur existentiell fälschbar unter einem Angriff ohne Nachrichten sind. Wie bei der RSA-Signatur lässt sich durch die Verwendung einer Hash-Funktion ein angemessenes Sicherheitsniveau erreichen. Dabei ist die Blindheit der interaktiven Signatur nicht beeinträchtigt, da man alle diesbezüglichen Argumente auf das Bild der Hash-Funktion anwenden kann. Sowohl für die Blindheit als auch für die Sicherheit der interaktiven Signatur ist jedoch wichtig, dass man eine gute Hash-Funktion verwendet, d.h. eine Hash-Funktion, die einem Zufallsorakel möglichst nahe kommt. Diese Überlegungen werden in der folgenden Bemerkung noch einmal zusammengefasst.

Bemerkung 6.6 *Wir betrachten die interaktive Schnorr-Signatur Σ_I mit linearer Blendung.*

(a) *Verwendet man in Satz 6.12 das Verifikationsprotokoll aus Abbildung 6.9, so kann man statt der Nachricht m das Bild $\mathcal{H}(r, m)$ einer Hash-Funktion*

$\mathcal{H} : \mathbb{Z}_p \times \{0,1\}^* \to \mathbb{Z}_q$ *signieren lassen. Die Verifikationsgleichung im Verifikationsprotokoll muss dementsprechend abgeändert werden: Auch hier muss statt* m *der Wert* $\mathcal{H}(m)$ *verwendet werden. Wie man leicht verifiziert, bleibt die Aussage von Satz 6.12 unter diesen Voraussetzungen erhalten.*

(b) *Die interaktive Schnorr-Signatur mit Hash-Funktion und linearer Blendung besitzt die gleichen Sicherheitseigenschaften wie die perfekt blinde Schnorr-Signatur. Obwohl für die perfekt blinde Schnorr-Signatur bisher kein Sicherheitsbeweis bekannt ist, existieren doch zahlreiche Signaturen, die auf der blinden Schnorr-Signatur basieren und beweisbar sicher im Random-Oracle-Modell sind (z.B. die Okomoto-Schnorr-Signatur, die in [PS00] ausführlich untersucht wurde, oder der Vorschlag von Abe und Okamoto in [AO00]). Da die vorgeschlagenen Blendungen, ggf. nach einer Anpassung, auch in diesen Varianten verwendet werden können, lassen sich so verschiedene Varianten von perfekt blinden Signaturen erreichen, die beweisbar sicher im Random-Oracle-Modell sind.*

(c) *Unter der Bedingung, dass die Hash-Funktion* $\mathcal{H}$ *ein Zufallsorakel darstellt, besitzt die interaktive Schnorr-Signatur mit Hash-Funktion und linearer Blendung die gleichen Blindheitseigenschaften wie die interaktive Schnorr-Signatur aus Satz 6.12.*

Zum Schluss dieses Abschitts werden wir einen Überblick über die in der Literatur bekannten Varianten von blinden Schnorr-Signaturen geben:

Wie eingangs schon erwähnt, hat die Schnorr-Signatur mit verschiedenen Blindheitsgraden viele Anwendungen gefunden. Die in Satz 6.12 vorgestellte interaktive Schnorr-Signatur stellt eine Verallgemeinerung einiger dieser Varianten dar. Um dies deutlich zu machen, sind die verschiedenen Blendungsmöglichkeiten mit den entsprechenden Verweisen auf die Literatur in der folgenden Tabelle dargestellt. Somit wird klar, dass die vorgestellte interaktive Schnorr-Signatur ein übergeordnetes Schema all dieser Vorschläge darstellt. Diese Vorschläge machen keinen Gebrauch von dem Parameterraum Par, sodass dieser in der Tabelle nicht aufgeführt ist. Ferner ist zu erwähnen, dass es einen dritten Typ von geblendeten Signaturen gibt: Bekommt der Signierer im Signaturprotokoll zwar die Nachricht, kennt jedoch die Signatur auf diese Nachricht nicht, so spricht man von **Parameter verbergenden** (Pv) Signaturen; erhält der Signierer umgekehrt zwar die geblendete Nachricht, wird aber die ungeblendete Signatur verwendet, so spricht man von **Nachrichten verbergenden** (Nv) Signaturen. Diese Unterscheidung geht wie der Begriff der schwach blinden Signaturen auf die Arbeit [HMP95] zurück. In der Tabelle sind diese Vor-

schläge erfasst, es wird aber deutlich, dass diese mindestens für die Schnorr-Signatur nicht vollständig sind.

	$\mathcal{Z}$	z	$\rho_1(z)$	$\rho_2(z)$	$\rho_3(z)$	Blindheit	verwendet in
(1)	$\mathbb{Z}_q^2$	(u,v)	1	$-v$	u	perfekt	[Oka92] [S95] [HMP94b] [AO00]
(2)	$\mathbb{Z}_q^2$	(u,v)	1	v	u	perfekt	[BD99]
(3)	$\mathbb{Z}_q^* \times \mathbb{Z}_q$	(u,v)	u^{-1}	0	v	perfekt	[HMP94b] [Br93]
(4)	$\mathbb{Z}_q^* \times \mathbb{Z}_q$	(u,v)	u^{-1}	0	$u \cdot v$	perfekt	[ChP92] [CP93]
(5)	$\mathbb{Z}_q^*$	u	u^{-1}	0	0	schwach	[HMP95] [N99]
(6)	$\mathbb{Z}_q^*$	u	u^{-1}	u^{-1}	0	schwach	[HMP95]
(7)	$\mathbb{Z}_q^*$	u	u^{-1}	1	0	schwach	[HMP95]
(8)	$\mathbb{Z}_q$	u	0	$-u$	0	Nv	[HMP95]
(9)	$\mathbb{Z}_q$	u	1	0	u	Pv	[HMP95]

Es gibt weitere Vorschläge, die Nutzen aus dem Parameterraum Par ziehen, z.B. in [BD99]: Hier enthält Par ein weiteres Schlüsselpaar zu den gleichen Parametern wie der Schlüssel des Signierers in der Schnorr-Signatur, wobei der geheime Schlüssel des Signiereres der gleiche bleibt, der öffentliche jedoch bezüglich einer Basis berechnet wird, die dem Empfänger fest zugeordnet wird. In diesem speziellen Fall werden in jedem neuen Signaturschema neue Basen erzeugt, bzgl. derer die Schnorr-Signatur verifiziert wird, sodass eine vollständige Entblendung nicht notwendig ist. Würde man eine vollständige Entblendung durchführen, so würde man eine Signatur erhalten, in der alle drei Abbildungen ρ_1, ρ_2 und ρ_3 verwendet werden.

Ferner ist zu bemerken, dass die Blendungsfunktionen aus Satz 6.12 nicht alleine auf Schnorr-Signaturen angewendet werden können. Grundsätzlich ist eine analoge Diskussion mit allen Signaturen vom El-Gamal-Typ möglich, die eine blinde Variante besitzen. Solche Signaturen wurden in [HMP94b] ausführlich untersucht und in einem Meta-Signatur-Schema zusammengefasst. Dort werden lediglich die in der obigen Tabelle angegebenen Varianten untersucht. Ein weiteres Beispiel für eine blinde Signatur vom ElGamal-Typ ist die blinde Nyberg-Rueppel-Signatur. In [CPS94]

wird im Grunde die Variante (2) verwendet, um zu einer perfekt blinden Signatur zu gelangen.

Offensichtlich liefert Satz 6.12 eine Vielzahl von perfekt oder schwach blinden Schnorr-Signaturen, die in der Literatur bisher noch nicht untersucht wurden. An dieser Stelle ist anzumerken, dass Satz 6.12 sich auch auf andere Signaturen anwenden lässt, die sich aus Identifikationsschemata konstruieren lassen, wie z.B. auf die Fiat-Shamir oder die Guillou-Quisquater-Signatur. Auch hier gibt Satz 6.12 ein übergeordnetes Signaturschema an, das ungeblendete, schwach blinde und perfekt blinde Signaturen vereint. Satz 6.13 lässt sich ebenso auf diese Signaturen übertragen, sodass auch hier Bedingungen für den Grad der Blindheit vorliegen.

6.7 Fazit

In diesem Kapitel wurde, basierend auf einem gegebenen Signaturschema Σ, ein generisches Verfahren zur Konstruktion von interaktiven Signaturschemata Σ_I angegeben. Diese lassen sich als Verallgemeinerung der Signatur Σ auffassen.

Es wurden die so entstehenden interaktiven Varianten der RSA- und Schnorr-Signaturen bezüglich ihrer Blindheitseigenschaften untersucht. Dabei konnten im Falle der interaktiven RSA-Signatur notwendige und hinreichende Bedingungen an eine perfekt oder rechnerisch blinde interaktive Signatur angegeben werden. Auf diese Weise konnte eine Verallgemeinerung der perfekt blinden RSA-Signatur von Chaum angegeben werden, die sowohl zu perfekt als auch zu rechnerisch blinden Varianten führt. Da die Blindheit von Signaturen im wesentlichen auf der Wahl von gleichverteilten Zufallszahlen basiert, die in der Praxis nicht wie in den Protokollen vorgesehen zu erreichen ist, ist die Frage nach Anforderungn an Pseudozufallszahlengeneratoren von besonderem Interesse. Hier konnte ein praxisnahes Modell, insbesondere für die blinde Chaum-Signatur, angegeben werden, dass diesem Umstand Rechnung trägt.

Ferner wurde die Schnorr-Signatur untersucht. Hier konnten notwendige und hinreichende Bedingungen für die perfekte Blindheit angegeben werden, jedoch nur eine notwendige Bedingung für die rechnerische Blindheit. Diese ermöglicht es jedoch wie bei der RSA-Signatur, Bedingungen an den verwendeten Pseudozufallszahlengenerator zu formulieren. Die Untersuchung der perfekten Blindheit resultiert in einem übergeordenten blinden interaktiven Signaturschema, das als Spezialfälle sowohl bekannte als auch etliche, bisher in der Literatur noch nicht untersuchte, perfekt blinde

Varianten der Schnorr-Signatur enthält. Ebenso ergeben sich neben den bekannten schwach blinde Schnorr-Signatur neue schwach blinde Varianten.

Damit konnte in beiden Fällen einerseits eine Verallgemeinerung und neue Varianten der bekannten blinden Signaturen angegeben werden, die den gleichen Blindheitsgrad besitzen. Andererseits konnten Bedingungen an die in der Praxis verwendeten Pseudozufallszahlengeneratoren angegeben werden. Dies ist als ein Schritt in Richtung einer angewandten Beschreibung von (perfekt) blinden interaktiven Signaturen zu verstehen.

Kapitel 7

Blindheit im Verifikationsprotokoll

Nachdem im letzten Kapitel eine generische Möglichkeit aufgezeigt wurde, das Signaturprotokoll zu blenden, wird nun die Blendung während des Verifikationsprotokolls betrachtet. Damit ist das Ziel dieses Kapitels einerseits, nachzuweisen, dass blinde interaktive Signaturen mit blindem Verifikationsprotokoll existieren, dass die in dieser Arbeit gegebene Definition von interaktiven Signaturschemata also zu einer echten Erweiterung des Begriffs der blinden Signatur führt. Andererseits sollen, wie schon im letzten Kapitel, Eigenschaften eines perfekten oder rechnerisch blinden Verifikationsprotokolls angegeben werden. Dazu betrachten wir eine spezielle Konstruktion einer interaktiven Signatur.

Es ist naheliegend, dass Eigenschaften von Zero-Knowledge-Beweisen für ein blindes Verifikationsprotokoll von Nutzen sein können: Kann der Empfänger einer Signatur in Zero-Knowledge beweisen, dass er eine Signatur auf eine Nachricht besitzt, ohne das Nachrichten-Signatur-Paar preiszugeben, so wird das Verifikationsprotokoll einen gewissen Grad an Blindheit erreichen. Hier könnte man einwenden, dass die Übermittlung der Nachricht an den Verifizierer ein Charakteristikum einer blinden Signatur ist. Betrachtet man aber blinde Signaturen als kryptographischen Baustein, so gibt es doch genug Anwendungen, die ein solches Vorgehen nicht verlangen. Dazu gehören beispielsweise elektronische Münzsysteme, in denen es nur von Wichtigkeit ist, eine Signatur auf eine beliebige Nachricht, die ggf. eine gewisse Struktur besitzen sollte, zu erhalten. Es stellt sich die Frage, welche Signaturen für ein solches Vorgehen geeignet sein könnten. Wir betrachten zunächst die Signaturen aus dem letzten Kapitel. Hier war die Verwendung einer Hash-Funktion notwendig, um ein vertretbares Sicherheitsniveau zu erhalten. Diese macht aber die Verwendung von Zero-Knowledge-Beweisen schwierig. Es ist also sinnvoll, Signaturschemata zu betrachten, die ein gewisses Sicherheitsniveau ohne die Verwendung von Hash-Funktionen erreichen, wie zum Beispiel die Signatur von Camenisch und Lysyanskaya, die in Ab-

schnitt 1.1.3.3 vorgestellt wurde. In den folgenden Abschnitten wird dieses Vorgehen genauer beleuchtet. Wir beginnen mit der Konstruktion einer interaktiven Signatur, basierend auf einem gegebenen Signaturschema und einem Zero-Knowledge-Beweis für die Kenntnis einer Signatur. In Abschnitt 7.1 wird eine Analyse der Blindheitseigenschaften durchgeführt, bevor wir in Abschnitt 7.2 die Sicherheitseigenschaften diskutieren. Zum Schluss dieses Kapitels wird in Abschnitt 7.3 die Konstruktion anhand des oben genannten Beispiels der Camenisch-Lysyanskaya-Signatur verdeutlicht.

Möchte ein Signierer die Blindheit der interaktiven Signatur angreifen, so kann er am meisten Nutzen aus einem ungeblendeten Signaturprotokoll ziehen. Die Ergebnisse aus Kapitel 4 und Kapitel 5 zeigen, dass auch in diesem Fall blinde interaktive Signaturen durch blinde Verifikationsprotokolle möglich sind. Um das oben skizzierte Vorgehen zu präzisieren, betrachten wir im Folgenden ein Signaturschema $\Sigma = (G, \sigma, V)$. Sei $(\mathsf{pk}, \mathsf{sk}) \in G(1^k)$ ein von einem Signierer $\mathcal{S}$ erzeugtes Schlüsselpaar und σ_I ein ungeblendetes, durchführbares interaktives Signaturprotokoll bezüglich σ, wie in Abbildung 7.1 beschrieben.

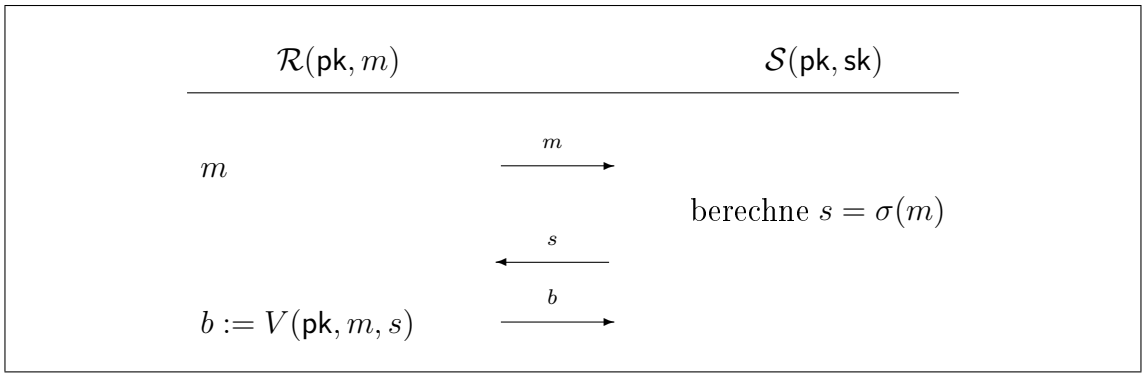


Abbildung 7.1: Signaturprotokoll σ_I

Damit erhält ein Empfänger $\mathcal{R}$ bei Eingabe von pk und m nach Durchführung von σ_I mit $\mathcal{S}$ als Ergebnis eine gültige Signatur $\sigma(m) \in S(m)$, wenn sich $\mathcal{R}$ und $\mathcal{S}$ an die Vorgaben von σ_I halten. Es sei weiterhin $\mathcal{C}$ ein Commitmentschema und $\mathsf{com}(m)$ ein damit generiertes Commitment auf die Nachricht m sowie Π ein interaktiver Beweis für die Kenntnis eines Nachrichten-Signatur-Paares, also dafür, dass der Empfänger $\mathcal{R}$ ein Paar

$$(m, \sigma(m)) \in \{\ (m, s)\ | V(\mathsf{pk}, m, s) = 1, c = \mathsf{com}(m)\}$$

kennt. Wir betrachten das in Abbildung 7.2 beschriebene Verifikationsprotokoll.

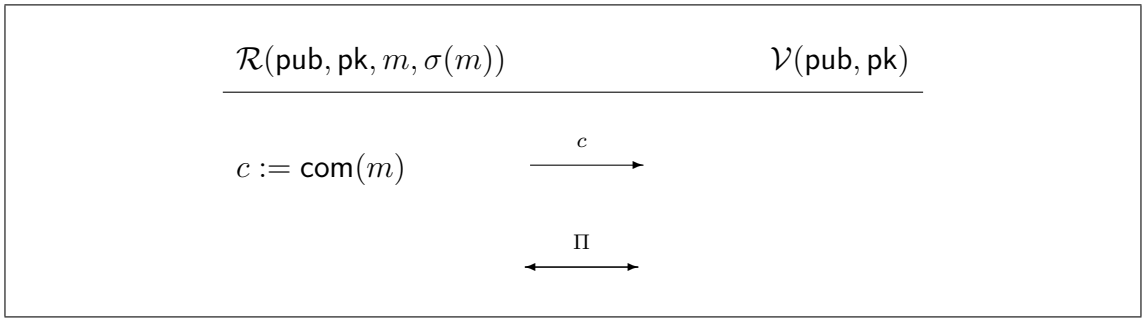


Abbildung 7.2: Verifikationsprotokoll V_I

Man beachte, dass das Commitment $\mathsf{com}(m)$ zur Festlegung der Nachricht zu Anfang des Protokolls dient. Damit ist klar, dass sich alle folgenden Schritte auf die Nachricht m beziehen.

Unter der Voraussetzung, dass ein interaktiver Beweis für die Kenntnis von $\sigma(m)$ existiert, erwartet man intuitiv, dass (G, σ_I, V_I) eine interaktive Signatur ist.

Satz 7.1 *Sei $\Sigma = (G, \sigma, V)$ ein Signaturschema zum Nachrichtenraum $\mathcal{M}$ und $\mathcal{C}$ ein Commitmentschema. Weiterhin sei σ_I ein Zwei-Parteien-Protokoll gemäß Abbildung 7.1 sowie V_I das in Abbildung 7.2 beschriebene Protokoll. Dann gilt:*
Ist Π ein interaktiver Beweis, so ist $\Sigma_I = (G, \sigma_I, V_I)$ ein interaktives Signaturschema.

Beweis. Die Durchführbarkeit von σ_I folgt aus der Durchführbarkeit des Signaturschemas Σ. Wir betrachten das Verifikationsprotokoll. Hier folgt die Durchführbarkeit aus der Durchführbarkeit von $\mathcal{C}$ und des interaktiven Beweises. □

Somit haben wir unter Verwendung einer Signatur Σ, eines Commitmentschemas $\mathcal{C}$ und eines interaktiven Beweises Π eine interaktive Signatur Σ_I konstruiert. In den nächsten Abschnitten werden wir zunächst untersuchen, unter welchen Bedingungen an die Bausteine sich die Blindheit von Σ_I umsetzen lässt. Weiterhin werden wir die Sicherheitsanforderungen diskutieren.

7.1 Analyse der Blindheit

Das Ziel dieses Abschnitts ist, Bedingungen an das Commitmentschema $\mathcal{C}$ und den interaktiven Beweis Π anzugeben, unter denen rechnerische oder perfekte Blindheit erreicht wird. Diese Aussagen dienen insbesondere zur Untersuchung der im letzten Abschnitt dieses Kapitels vorgestellen interaktiven Signatur.

7.1.1 Perfekte Blindheit

Wir untersuchen zunächst die perfekte Blindheit. Wie oben bereits erwähnt, ist es plausibel, dass man aus Zero-Knowledge-Eigenschaften von Π Nutzen für die Blindheit ziehen kann. Allerdings genügt diese Forderung nicht: Auch wenn durch Π keinerlei Information über (m, s) übermittelt werden, ist der Verifizierer doch im Besitz von $c = \mathsf{com}(m)$. Verrät der Wert c etwas über die Nachricht m, so wird man kein perfekt blindes Verifikationsprotokoll erhalten. Demzufolge gilt

Satz 7.2 *Sei $\Sigma_I = (G, \sigma_I, V_I)$ ein interaktives Signaturschema gemäß Satz 7.1. Dann gilt:*
Ist $\mathcal{C}$ ein perfekt verbergendes Commitmentschema und besitzt Π die perfekte Zero-Knowledge-Eigenschaft, so ist Σ_I perfekt blind.

Beweis. Sei Π ein perfekter Zero-Knowledge-Beweis und $\mathcal{C}$ ein perfekt verbergendes Commitmentschema. Es seien x_1 und x_2 die Inhalte der Zufallsbänder von $\mathcal{R}$ bzw. $\mathcal{V}$. Wir betrachten die Zufallsvariablen $view_{\mathcal{V}}(V_I)$, c, $s := (m, \sigma(m))$ und $\mathsf{key} := (\mathsf{pk}, \mathsf{sk})$.

Zunächst stellen wir fest, dass $view_{\mathcal{V}}(V_I)$ eine deterministische Funktion in den beschriebenen Zufallsvariablen ist: $view_{\mathcal{V}}(V_I) = f(x_1, x_2, s, \mathsf{key})$.

Weiterhin sind $(m, \sigma(m))$ und c stochastisch unabhängig, da $\mathcal{C}$ ein perfekt verbergendes Commitmentschema ist.

Zuletzt folgt aus der perfekten Zero-Knowledge-Eigenschaft, dass es eine probabilistische polynomielle Turing-Maschine M gibt, sodass die Zufallsvariable $M(c)$, welche die Ausgabe von M bei Eingabe von c beschreibt, und $view_{\mathcal{V}}(V_I)$ die gleiche Verteilung besitzen. Damit gilt:

$$\begin{aligned} & I[(m, \sigma(m)), view_{\mathcal{V}}(V_I)] \\ &= H((m, \sigma(m))) + H(view_{\mathcal{V}}(V_I)) - H((m, \sigma(m)), view_{\mathcal{V}}(V_I)) \\ &= H((m, \sigma(m))) + H(M(c)) - H((m, \sigma(m)), M(c)) \\ &= H((m, \sigma(m))) + H(M(c)) - H((m, \sigma(m))) - H(M(c)) = 0, \end{aligned}$$

und die Behauptung folgt mit Satz 4.3 , da Voraussetzung (b) aus Satz 4.1 für die betrachtete Konstruktion erfüllt ist. □

Zum Schluss unserer Überlegungen bzgl. der perfekten Blindheit beleuchten wir die Frage nach der Existenz von Protokollen V_I gemäß Abbildung 7.2. Da es verschiedene Möglichkeiten gibt, perfekt verbergende Commitmentschemata zu konstruieren,

können wir uns bei dieser Frage auf die nach der Existenz von interaktiven Beweisen für die Kenntnis eines geeigneten Nachrichten-Signatur-Paares beschränken. Dieser Punkt wird in der folgenden Bemerkung noch einmal näher beleuchtet.

Bemerkung 7.1 *Setzt man den Empfänger als polynomiell beschränkten Angreifer voraus (wie wir es in Abschnitt 3.2 getan haben), so gibt es (unter der Voraussetzung, dass nicht-uniforme Einweg-Permutationen existieren) für jedes Problem der Komplexitätsklasse* **NP** *ein sogenanntes perfektes Zero-Knowledge-Argument, also einen Beweis, der die perfekte Zero-Knowledge-Eigenschaft besitzt, für den jedoch die Korrektheit nur bzgl. effizienter Algorithmen gegeben ist. Damit erhält man auf diese Weise eine ganze Klasse von neuen blinden interaktiven Signaturen.*

7.1.2 Rechnerische Blindheit

Wir betrachten die rechnerische Blindheit. Hier kommt man im Prinzip zu dem gleichen Ergebnis wie im Falle der perfekten Blindheit. Da die parallele Durchführung von Zero-Knowledge-Protokollen nicht zwingend zu Protokollen mit Zero-Knowledge-Eigenschaft führt (vgl. [Go03]), betrachten wir zunächst sequentielle Angriffe auf die Blindheit.

Satz 7.3 *Sei Σ_I ein interaktives Signaturschema gemäß Satz 7.1. Dann gilt: Ist $\mathcal{C}$ ein rechnerisch verbergendes Commitmentschema und Π ein interaktiver Beweis, der die rechnerische Zero-Knowledge-Eigenschaft besitzt, so ist Σ_I rechnerisch blind unter einem sequentiellen Angriff.*

Bevor wir diesen Satz beweisen, zeigen wir folgendes

Lemma 7.1 *Es sei ν eine vernachlässigbare Funktion in k und f eine nicht vernachlässigbare Funktion in k. Dann ist auch $f + \nu$ nicht vernachlässigbar.*

Beweis. Wir nehmen an, dass $f+\nu$ vernachlässigbar ist und zeigen, dass f in diesem Falle ebenfalls vernachlässigbar sein muss. Sei dazu $c \in \mathbb{N}$. Dann gibt es für $c+1$ ein $k_1 \in \mathbb{N}$, sodass für alle $k \geq k_1$ gilt:

$$|f(k) + \nu(k)| \leq k^{-c-1}.$$

Ferner gibt es ein k_2, sodass für alle $k \geq k_2$ gilt

$$|\nu(k)| \leq k^{-c-1}.$$

Sei $k_c := \max\{k_1, k_2\} + 2$. Dann gilt für alle $k \geq k_c$:

$$|f(k) + \nu(k)| + |\nu(k)| \leq k^{-c-1} + k^{-c-1} = 2k^{-c-1} \leq k \cdot k^{-c-1} = k^{-c}.$$

Da gilt

$$|f(k)| - |\nu(k)| \leq |f(k) + \nu(k)|,$$

also

$$|f(k)| \leq |f(k) + \nu(k)| + |\nu(k)|,$$

folgt die Vernachlässigbarkeit von f und damit die Behauptung. □

Mithilfe des Lemmas führen wir den Beweis zu Satz 7.3:

Beweis. Sei $\mathcal{C}$ ein rechnerisch verbergendes Commitmentschema und Π ein rechnerischer Zero-Knowledge-Beweis. Wir nehmen an, dass V_I nicht rechnerisch blind unter einem sequentiellen Angriff ist, und konstruieren aus einem effizienten Angreifer $\mathcal{A}$, der die rechnerische Blindheit mit Erfolgswahrscheinlichkeit ε angreift, einen effizienten Angreifer $\mathcal{A}^*$ auf $\mathcal{C}$. Dabei ist $1/2 - \varepsilon(k)$ eine nicht vernachlässigbare Funktion in k. Wir zeigen, dass $\mathcal{A}^*$ folgendes Spiel gegen den Empfänger $\mathcal{R}$ mit einer nicht vernachlässigbar von $1/2$ abweichenden Erfolgswahrscheinlichkeit $\varepsilon^*(k)$ gewinnt. Sei dazu v der öffentliche Parameter des Commitmentschemas $\mathcal{C}$. Man beachte, dass alle Empfänger für ihre Commitments den gleichen öffentlichen Parameter verwenden

1. $\mathcal{R}$ wählt ein Bit $b \in_R \{0, 1\}$.
2. Das Orakel $\mathcal{O}$ erhält v und b als Eingabe.
3. $\mathcal{A}^*$ startet $\mathcal{A}$, der ein Schlüsselpaar $(\mathsf{pk}, \mathsf{sk})$ sowie zwei Nachrichten-Signatur-Paare $s_0 := (m_0, \sigma(m_0))$ und $s_1 := (m_1, \sigma(m_1))$ erzeugt. $\mathcal{A}$ gibt s_0, s_1 und pk an $\mathcal{A}^*$ weiter.
4. $\mathcal{A}^*$ gibt m_0, m_1 an das Orakel $\mathcal{O}$, das $c := \mathsf{com}(m_b)$ berechnet und dieses an $\mathcal{A}^*$ zurückgibt.
5. $\mathcal{A}^*$ verwendet nun den Simulator M für das Zero-Knowledge-Protokoll, um $view_{\mathcal{V}}(V_I)$ zu simulieren, und führt auf diese Weise V_I (aus Sicht von $\mathcal{A}$ in der Rolle des Orakels) aus.
6. $\mathcal{A}^*$ gibt die Ausgabe von $\mathcal{A}$, das Bit b' aus.

Verlangt $\mathcal{A}$ während des Spiels Zugriff auf sein Orakel $\mathcal{O}_b$, so kann $\mathcal{A}^*$ seine eigenes Orakel $\mathcal{O}_b^*$, das nach Definition 1.13 gegeben ist, verwenden, um korrekte Anworten auf die Anfragen gemäß dem oben beschriebenen Vorgehen zu erzeugen.

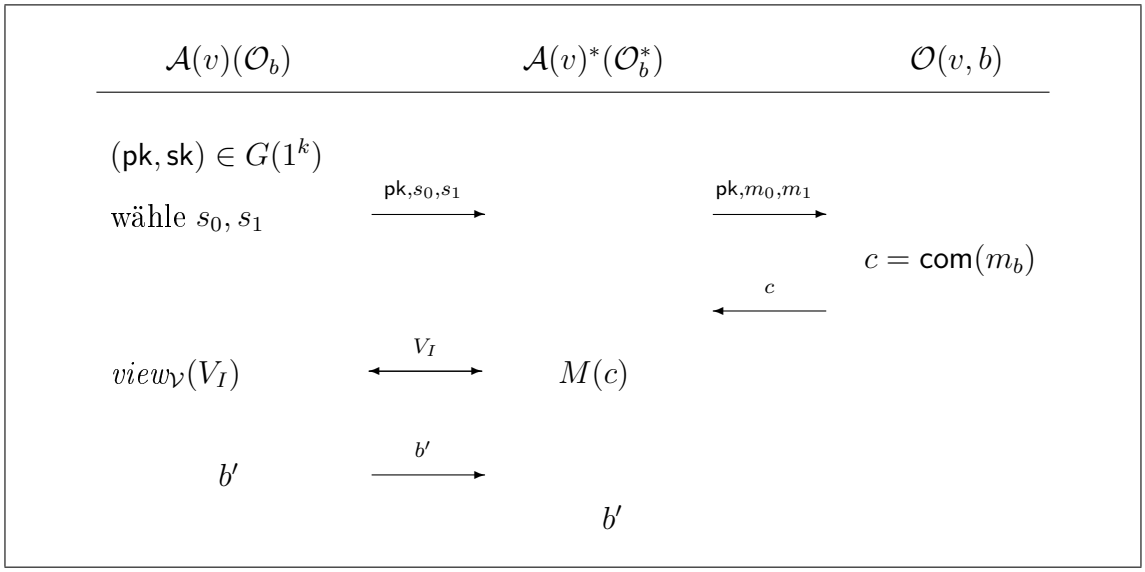


Abbildung 7.3: Beweis zu Satz 7.3

$\mathcal{A}^*$ ist effizient, da einerseits M ein effizienter Algorithmus und andererseits $\mathcal{A}$ eine effiziente interaktive Turing-Maschine ist. Wir zeigen, dass für die Erfolgswahrscheinlichkeit $\varepsilon^*(k)$ von $\mathcal{A}^*$ gilt: $1/2 - \varepsilon^*(k)$ ist nicht vernachlässigbar. Offensichtlich gewinnt $\mathcal{A}^*$ das Spiel genau dann, wenn $\mathcal{A}$ das richtige Bit ausgibt, d.h. es gilt

$$\varepsilon^*(k) = \mathcal{P}(\mathcal{A}^*(\mathsf{pk}, m_0, m_1, c) = b) = \mathcal{P}(\mathcal{A}((\mathsf{pk},\mathsf{sk}), s_0, s_1, c, M(c)) = b).$$

Man beachte, dass

$$\varepsilon(k) = \mathcal{P}(\mathcal{A}((\mathsf{pk},\mathsf{sk}), s_0, s_1, c, view_{\mathcal{V}}(V_I)) = b)$$

ist. Ferner besitzt Π die rechnerische Zero-Knowledge-Eigenschaft, d.h.

$$\mathcal{P}(\mathcal{A}((\mathsf{pk},\mathsf{sk}), s_0, s_1, c, M(c)) = b) - \mathcal{P}(\mathcal{A}((\mathsf{pk},\mathsf{sk}), s_0, s_1, c, view_{\mathcal{V}}(V_I)) = b)$$

ist vernachlässigbar in k. Damit ist $\varepsilon(k) - \varepsilon^*(k)$ vernachlässigbar in k. Nach Lemma 7.1 ist somit

$$1/2 - \varepsilon^*(k) = (1/2 - \varepsilon(k)) + (\varepsilon(k) - \varepsilon^*(k))$$

nicht vernachlässigbar in k. Somit weicht die Erfolgswahrscheinlichkeit von $\mathcal{A}^*$ nicht vernachlässigbar von $1/2$ ab. □

Man beachte, dass die rechnerische Zero-Knowledge-Eigenschaft die schwächste Forderung für Zero-Knowledge-Protokolle ist, sodass Satz 7.3 auch gilt, wenn man von Π fordert, dass es die statistische oder perfekte Zero-Knowledge-Eigenschaft besitzt.

Nachdem wir die rechnerische Blindheit unter sequentiellen Angriffen gezeigt haben, stellt sich die Frage, ob sich ein vergleichbares Ergebnis unter parallelen Angriffen erzielen lässt. Diese wird in der nächsten Bemerkung beleuchtet.

Bemerkung 7.2 *Man beachte, dass die oben beschriebene interaktive Signatur zunächst nicht unbedingt rechnerisch blind unter einem parallelen Angriff ist, da die parallele Durchführung von Zero-Knowlege-Protokollen nicht mehr unbedingt die Zero-Knowledge-Eigenschaft besitzt. Dieses Problem lässt sich jedoch beheben, indem man zu nicht-interaktiven Zero-Knowledge-Beweisen übergeht: Hier besteht die Verifikation nur aus einem einzigen Kommunikationsschritt des Empfängers, der sowohl das Commitment als auch den nicht-interaktiven Beweis an den Verifizierer weitergibt. Auf diese Weise kann der Verifizierer aus einer parallelen Durchführung keinen Nutzen ziehen, sodass hier die rechnerische Blindheit erreicht ist.*

Zum Schluss dieses Abschnitts betrachten wir die Frage nach der Existenz von rechnerisch blinden Verifikationsprotokollen gemäß Abbildung 7.2:

Bemerkung 7.3 *Unter der Bedingung, dass Einweg-Funktionen existieren, gibt es zu jeder Sprache aus NP einen Beweis, der die rechnerische Zero-Knowledge-Eigenschaft besitzt. Im Gegensatz zu den perfekt blinden Signaturen muss man hier keine Einschränkungen an den Empfänger machen.*

7.2 Analyse der Sicherheit

Zum Schluss unserer Überlegungen sollen die Sicherheitseigenschaften der vorgeschlagenen Signatur untersucht werden.

Zunächst stellen wir fest, dass das Protokoll σ_I nur aus den Kommunikationsschritten besteht, die eine Erzeugung einer regulären Signatur fordern. Gehen wir also von einem unter einem Angriff mit gewählten Nachrichten sicheren Signaturschema (G, σ, V) aus, so wird kein effizienter Angreifer in der Lage sein, eine Signatur auf eine Nachricht vorzuweisen, die der Signierer nie zu Gesicht bekommen hat. Damit bleibt nur ein Betrug während des Verifikationsprotokolls. Hier sorgt aber die Korrektheit des interaktiven Beweises sowie die Sicherheit des Commitmentschemas für die Sicherheit der Konstruktion. Dieses Ergebnis werden wir in der folgenden Bemerkung etwas präzisieren:

Bemerkung 7.4 *Es sei die interaktive Signatur aus Satz 7.1 gegeben. Ist* $\Sigma = (G, \sigma, V)$ *sicher gegen einen adaptiven Angriff mit gewählten Nachrichten, ist das Commitmentschema* $\mathcal{C}$ *rechnerisch bindend und ist der interaktive Beweis* Π *korrekt,*

so ist auch Σ_I sicher gegen einen parallelen Angriff mit gewählten Nachrichten.

Um dies zu sehen, nehmen wir an, dass es einen betrügerischen effizienten Empfänger $\mathcal{R}^*$ gibt, dem es in einem adaptiven, parallelen Angriff mit ℓ Signaturen gelingt, den Verifizierer in $\ell + 1$ Protokolldurchführungen von V_I davon zu überzeugen, dass er eine Signatur besitzt. Da der interaktive Beweis korrekt ist, sind es auch parallele Durchführungen des interaktiven Beweises, und so muss $\mathcal{R}^*$ tatsächlich im Besitz von $\ell + 1$ Nachrichten-Signatur-Paaren sein, wobei das während der Verifikation übermittelte Commitment jeweils zu der Nachricht passt. Da das Commitmentschema rechnerisch (oder sogar perfekt) bindend ist, muss $\mathcal{R}^*$ sogar im Besitz von $\ell + 1$ verschiedenen Nachrichten sein, selbst wenn er zweimal das gleiche Commitment versendet. Damit muss der Empfänger $\mathcal{R}^*$ tatsächlich $\ell + 1$ verschiedene Nachrichten-Signatur-Paare kennen. Da die Signatur σ sicher gegen einen adaptiven Angriff mit gewählten Nachrichten ist, kann er diese aber nur aus $\ell+1$ Durchführungen des Signaturprotokolls erhalten haben. Damit haben wir einen Widerspruch zur Annahme erreicht.

Somit ist die Sicherheit unserer Konstruktion gewährleistet. Damit beschließen wir die Untersuchung der allgemeinen Konstruktion und kommen zu einem Beispiel für ein interaktives Signaturschema nach diesem Vorgehen.

7.3 Die interaktive Camenisch-Lysyanskaya-Signatur

In [CL02] wurde ein Signaturschema vorgestellt, das sich für eine Konstruktion nach Satz 7.1 eignet (vgl. Abschnitt 1.1.3.3). In dem selben Papier wurde eine Möglichkeit aufgezeigt, nachzuweisen, dass man eine Signatur auf einen festgelegten Wert besitzt. Daraus ergibt sich die Möglichkeit, die oben beschriebene Konstruktion umzusetzen. Diese werden wir im Folgenden beschreiben.

Es sei das Signaturschema $\Sigma = (G, \sigma, V)$ wie in Abschnitt 1.1.3.3 beschrieben. Nach der Ausführung des Schlüsselerzeugungsalgorithmus' besitzt der Signierer $\mathcal{S}$ das Schlüsselpaar

$$(\mathsf{pk}, \mathsf{sk}) = ((n, a, b, c), p)$$

sowie die Parameter $L := (l_m, l_e, l_s)$, wobei alle Parameter die dort angegebenen Eigenschaften haben. Das Signaturprotokoll sei das in Abbildung 7.4 dargestellte. Nach Durchführung des Signaturprotokolls besitzt $\mathcal{R}$ eine Signatur (e, s, v) zur Nachricht m. Im Verifikationsprotokoll wird das in Abschnitt 1.1.4 vorgestellte Commitmentschema $\mathcal{C}$ verwendet.

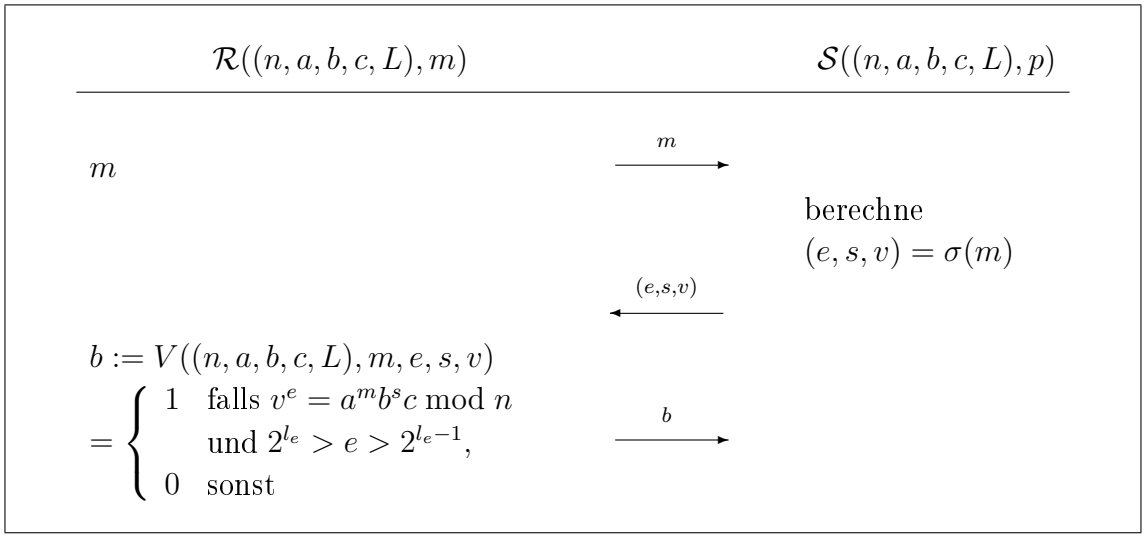


Abbildung 7.4: Signaturprotokoll der interaktiven CL-Signatur

Seien dazu (n, g, h) die öffentlichen Parameter, die bei Eingabe eines Sicherheitsparameters k von dem Schlüsselerzeugungsalgorithmus von $\mathcal{C}$ generiert wurden, d.h. h ist ein quadratischer Rest modulo n und g ein Element der von h erzeugten Gruppe.

Der interaktive Beweis Π sei der in [CL02] angegebene Zero-Knowledge-Beweis für die Kenntnis eines Tupels (x, r, s, e, v), sodass $c = g^x h^r \bmod n$ und $V(m, s, e, v) = 1$ gilt. Mit diesen Bausteinen sei V_I wie in Abbildung 7.5 angegeben.

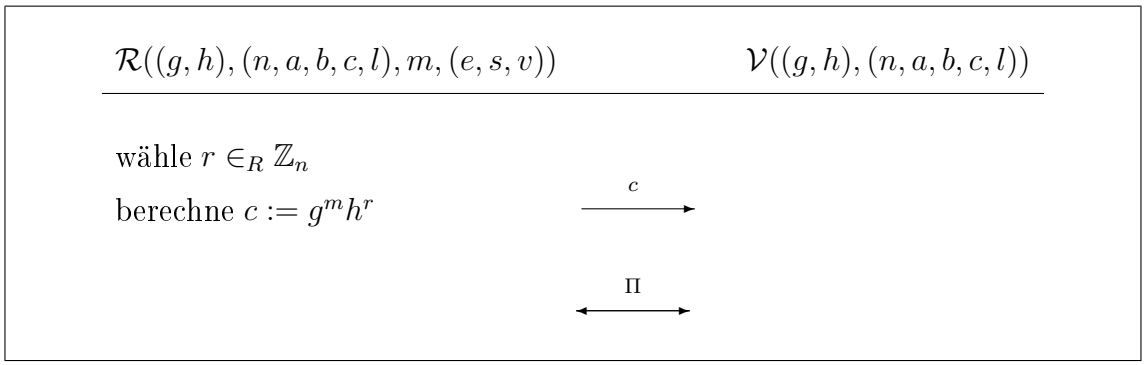


Abbildung 7.5: Verifikationsprotokoll der interaktiven CL-Signatur

Bevor wir die in den vorangegangenen Abschnitten gewonnen Ergebnisse auf diese Konstruktion anwenden, werden wir zunächst die benötigten Eigenschaften der einzelnen Bausteine zusammenstellen:

Die Camenisch-Lysyanskaya-Signatur Σ ist unter der starken RSA-Annahme rechnerisch sicher gegen existentielle Fälschbarkeit unter einem adaptiven Angriff mit gewählten Nachrichten (vgl. Abschnitt 1.1.3.3). Das Commitmentschema $\mathcal{C}$ ist statistisch verbergend und rechnerisch bindend. (vgl. [CL02], Lemma 6). Ferner ist der interaktive Beweis Π ein statistischer Zero-Knowledge-Beweis für die Kenntnis eines Tupels (x, r, s, e, v), sodass $c = g^x h^r \bmod n$ und $V(m, s, e, v) = 1$ gilt (s. [CL02], Lemma 8).

Damit können wir unsere Ergebnisse aus den Abschnitten 7.1 und 7.2 auf die vorgestellte Konstruktion anwenden:

Satz 7.4 *Sei $\Sigma = (G, \sigma, V)$ die CL-Signatur, σ_I wie in Abbildung 7.4 und V_I wie in Abbildung 7.5 gegeben. Dann gilt unter der starken RSA-Annahme:*

(a) *$\Sigma_I = (G, \sigma_I, V_I)$ ist eine sichere interaktive Signatur.*

(b) *Σ_I ist rechnerisch blind.*

Beweis. Die Aussage folgt direkt aus den oben angegebenen Eigenschaften der einzelnen Bausteine und den Sätzen 7.1 und 7.3 sowie Bemerkung 7.4. □

An dieser Stelle sei erwähnt, dass in [CL02] keine derartige Zusammenstellung vorgeschlagen wurde. Dort wurde die Konstruktion mit dem Ziel erarbeitet, ein Signaturschema zu erhalten, mit dem man Werte signieren kann, die in einem Commitment verborgen sind. Auch das Protokoll Π wurde zu diesem Zweck angegeben. Da wir diesen Schritt für unsere Zwecke nicht benötigen, stellen wir hier eine neue Variante vor.

Ferner ist noch zu erwähnen, dass die Bausteine der angegebenen Konstruktion stärkere Eigenschaften haben, als für die rechnerische Blindheit notwendig sind:

Bemerkung 7.5 *Das interaktive Protokoll Π ist ein statistischer Zero-Knowledge-Beweis und das Commitmentschema $\mathcal{C}$ ist statistisch verbergend. Die statistische Zero-Knowledge-Eigenschaft bedeutet anschaulich, dass die Verteilung der Zufallsvariable, die die Ausgabe des Simulators M beschreibt, nur vernachlässigbar von der Verteilung der Zufallsvariable $view_{\mathcal{V}}(V_I)$ abweicht. Für das Commitmentschema bedeutet das, dass sich die Verteilungen von $\mathsf{com}(m_0)$ und $\mathsf{com}(m_1)$ zu zwei verschiedenen Nachrichten m_0 und m_1 nur vernachlässigbar unterscheiden. Man beachte, dass diese Eigenschaften unabhängig von den Kapazitäten des Angreifers sind. Bezogen auf die Blindheit des Protokolls V_I bedeutet das, dass auch ein unbeschränkter Angreifer Spiel 5.3 aus der Definition der rechnerischen Blindheit nur*

mit vernachlässigbarer Wahrscheinlichkeit gewinnen kann. In diesem Sinne ist die Blindheit von Σ_I stärker als die rechnerische Blindheit.

Zum Schluss dieses Abschnitts sei erwähnt, dass sich die in [STY00] vorgestellten **blind auditable membership proofs** zu einer analogen Konstruktion eigenen, die im wesentlichen die gleichen Sicherheits- und Blindheitseigenschaften besitzt.

7.4 Fazit

In diesem Kapitel haben wir die Konstruktion einer interaktiven Signatur angegeben, die unter gewissen Bedingungen ein blindes Verifikationsprotokoll ermöglicht. Dazu haben wir ein ungeblendetes Signaturprotokoll betrachtet und das Verifikationsprotokoll als interaktiven Beweis für die Kenntnis einer interaktiven Signatur vorgeschlagen.

Es wurde eine Analyse dieser Konstruktion durchgeführt, die hinreichende Bedingungen für die perfekte bzw. rechnerische Blindheit sowie für die Sicherheit der vorgeschlagenen interaktiven Signatur ergab. Da die Blindheitseigenschaften auf der perfekten bzw. rechnerischen Zero-Knowledge-Eigenschaft des interaktiven Protokolls beruhen, wurde die Frage nach der Existenz solcher Signaturen zunächst allgemein beantwortet: Unter kryptographischen Standardannahmen, wie der Existenz von Einweg-Funktionen bzw. von (nicht-uniformen) Einwegpermutationen, gibt es für jedes Signaturschema und jedes Commitmentschema ein interaktives Protokoll mit der rechnerischen bzw. perfekten Zero-Knowledge-Eigenschaft, das sich zur Umsetzung der Konstruktion eignet. Allerdings sind die Konstruktionen interaktiver Zero-Knowledge-Protokolle, die zu solchen Existenzaussagen führen, im Allgemeinen nicht effizient genug, um Protokolle anzugeben, die praktisch von Nutzen sind.

Um zu zeigen, dass es dennoch Beispiele für praktisch umsetzbare interaktive Signaturen mit blindem Verifikationsprotokoll gibt, wurde basierend auf der Camenisch-Lysyanskaya-Signatur ein Beispiel für eine rechnerisch blinde interaktive Signatur angegeben. Bei der Untersuchung der Eigenschaften konnte insbesondere auf die vorher erzielten Ergebnisse zurückgegriffen werden.

Damit haben wir in diesem Kapitel gezeigt, dass der Begriff der blinden interaktiven Signatur eine echte Erweiterung des Begriffs der blinden Signaturen darstellt, d.h. es ist nicht nur aus allgemeinen Überlegungen heraus sinnvoll, die Interaktion von Empfänger und Verifizierer einer Signatur während der Verifikationsphase mit in das Konzept der blinden Signatur aufzunehmen, sondern man erhält so eine neue Klasse von blinden (interaktiven) Signaturen.

Zusammenfassung und Ausblick

Zusammenfassung

Die vorliegende Arbeit stellt einen neuen kryptographischen Baustein vor: Die interaktiven Signaturen. Diese modellieren im Gegensatz zu den klassischen Signaturen zusätzlich zu den Algorithmen die vorhandenen Kommunikationsschritte zwischen den an einer Signatur beteiligten Parteien Sender, Empfänger und Verifizierer.

Im ersten Teil der Arbeit wurde ein allgemeines Sicherheitsmodell für interaktive Signaturen angegeben. Dabei lag der Schwerpunkt auf der Untersuchung der Blindheitseigenschaften interaktiver Signaturen. Hier unterscheidet man zwischen dem stärksten Blindheitsbegriff, der perfekten Blindheit, und einem abgeschwächten, für die Praxis jedoch ausreichenden Blindheitsbegriff, der rechnerischen Blindheit. In beiden Fällen konnten Bedingungen sowohl an das Signatur- als auch an das Verifikationsprotokoll formuliert werden, die zu blinden Signaturen führen: Es wurde festgestellt, dass man die Blindheit in einer interaktiven Signatur durch das Verbergen der Nachricht während des Signaturprozesses oder durch das Verbergen des Nachrichten-Signaturpaares während des Verifikationsprozesses erreichen kann.

Im zweiten Teil der Arbeit wurden die Ergebnisse des ersten Teils zur Konstruktion von perfekt bzw. rechnerisch blinden interaktiven Signaturen eingesetzt. Dabei wurden zwei Wege beschritten:
Einerseits wurde ein generisches Konstruktionsprinzip für interaktive Signaturen mit Blendungsfunktionen angegeben. In diesen interaktiven Signaturen wird die Blendung während der Signatur erreicht. Es wurde im Falle der RSA- und der Schnorr-Signatur gezeigt, dass dieses Prinzip eine Verallgemeinerung der bekannten blinden RSA-Signatur von Chaum und verschiedener Varianten der (schwach) blinden Schnorr-Signatur darstellt. Weiterhin konnten verschiedene Varianten der Schnorr-Signatur in einem übergeordneten interaktiven Signaturschema zusammengefasst werden. Mit den im ersten Teil der Arbeit entwickelten Methoden konnte eine genaue Untersuchung der Blindheitseigenschaften der verallgemeinerten interaktiven

Signaturen durchgeführt werden, die zum einen eine praxisnahe Beschreibung des Blindheitsgrades in den Fällen der oben genannten Signaturen erlaubt. Zum anderen konnten im Falle der Schnorr-Signatur verschiedene Blendungsvarianten identifiziert werden. Durch die Verallgemeinerung der Chaum- und Schnorr-Signatur konnten in beiden Fällen neue Varianten für interaktive blinde Signaturen angegeben werden.

Ferner wurde eine Konstruktion für blinde interaktive Signaturen angegeben, in denen eine Blendung während der Verifikation realisiert wird. Auch hier wurden die Blindheitseigenschaften mit Hilfe der entwickelten Methoden mit dem Ergebnis analysiert, dass eine blinde interaktive Signatur angegeben werden konnte, die auf dem Prinzip der Blendung im Verifikationsprotokoll beruht. Als Basis dieser interaktiven Signatur wurde das von Camenisch und Lysyanskaya vorgestellte digitale Signaturschema verwendet. Die im ersten Teil der Arbeit gefundenen Bedingungen ermöglichen hier den Nachweis, dass der Vorschlag eine blinde interaktive Signatur darstellt. Damit konnte gezeigt werden, dass blinde interaktive Signaturen mindestens eine neue Klasse von blinden Signaturen beinhalten, nämlich diejenigen interaktiven Signaturen, deren Blindheit im Verifikationsprotokoll erreicht wird.

Ausblick

Wie in der Einleitung beschrieben wurde, sind die klassischen blinden Signaturen grundlegende Bausteine für viele elektronische Münzsysteme (z.B. [Ch82],[Br93], [Fe93]). Ein elektronisches Münzsystem stellt, vereinfacht gesagt, Methoden zur Verfügung, die es einem Kunden ermöglichen, eine digitale Münze von einer Bank abzuheben und diese bei einem Händler auszugeben, der die Münze schließlich wieder bei der Bank einlöst. Dabei kann man sich eine digitale Münze als Datensatz vorstellen, der eine bestimmte Struktur besitzt, sodass der Händler die Echtheit der Münze überprüfen kann. Im Allgemeinen geht man davon aus, dass der Händler keine online Verbindung zu der Bank besitzt, sodass die Sicherheit des Systems vollständig von den verwendeten kryptographischen Methoden abhängig ist. Durch die Verwendung von klassischen blinden Signaturen werden Eigenschaften wie die Echtheit der Münze und die Anonymität des Kunden gegenüber der Bank sichergestellt. Dabei wird während der Abhebung ein Datensatz signiert, der bei der Bezahlung von dem Händler verifiziert wird. Das bedeutet, dass die Anonymität des Kunden bereits während der Abhebung umgesetzt wird. Grundsätzlich bieten sich digitale Signaturen zur Sicherstellung der Echtheit einer Münze an. Die Anonymität kann jedoch auch während der Verifikation hergestellt werden, wie die Vorschläge in [STY00] und [CHL05] zeigen.

Für elektronischen Münzsysteme, die auf blinden Signaturen beruhen, wurde in [N99] ein generisches Konstruktionsprinzip angegeben, sodass Sicherheitseigenschaften der verwendeten Bausteine abstrakt beschrieben werden konnten. Insbesondere konnte basierend auf der angegebenen Konstruktion eine Sicherheitsanalyse von elektronischen Münzsystemen durchgeführt werden, wobei die Kommunikationsumgebung berücksichtigt wurde. Die zuletzt genannten Vorschläge für Münzsysteme konstituieren eine andere Vorgehensweise, sodass diese durch die oben genannte Konstruktion nicht mehr beschrieben werden. Eine vergleichbare Untersuchung erforderte die Formulierung eines zweiten Konstruktionsprinzips für Münzsysteme oder einen Austausch der verwendeten Bausteine der Konstruktion in [N99]. In Anbetracht der Struktur blinder interaktiver Signaturen bietet sich diese als Ersatz für die verwendeten blinden Signaturen an. Die ausführliche Diskussion dieser Frage würde den Rahmen der vorliegenden Arbeit sprengen, verspricht jedoch, ein interessanter Forschungsgegenstand der Zukunft zu sein.

Bezeichnungen

In dieser Arbeit werden, teils ohne weiteren Kommentar, folgende Bezeichnungen verwendet:

$\mathbb{N}$	Menge der natürliche Zahlen $\{0, 1, 2, \ldots\}$
$\mathbb{Z}$	Menge der ganzen Zahlen $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$
$\mathbb{Z}_n$	Restklassenring modulo n
$\mathbb{Z}_n^*$	Gruppe der primen Restklassen modulo n
$\mathcal{QR}_n$	Menge der quadratischen Reste modulo n
$\phi(n)$	Eulersche Phi-Funktion
$\{0,1\}^n$	die Menge aller Bitstrings der Länge n
$\{0,1\}^*$	die Menge aller Bitstrings endlicher Länge
$x \in_R X$	Wahl eines Elementes x gemäß einer Gleichverteilung aus der Menge X
$view_{\mathcal{A}}(\Pi)$	Protokollansicht eines Algorithmus' $\mathcal{A}$ im Protokoll Π
$\mathcal{H}(m)$	Bild einer Einweg-Hash-Funktion
$H(X)$	Entropie der Zufallsvariablen X
$\mathcal{I}(X, Y)$	Information der Zufallsvariablen X und Y

Literaturverzeichnis

[A96] M. Abe, E. Fujisaki: „How to date blind signatures“ *Advances in Cryptology - Asiacrypt 1996*, Lecture Notes in Computer Science, Vol. 1163, Springer-Verlag, Berlin, 1996, S. 244-251

[AO00] M. Abe, T. Okamoto: „Provably Secure Partially Blind Signatures“, *Advances in Cryptology - Crypto 2000*, Lecture Notes in Computer Science, Vol. 1880, Springer-Verlag, Berlin, 2000, S. 271-286

[BNPS03] M. Bellare, C. Namprempre, D. Pointcheval, M. Semanko: „The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature“, *Journal of Cryptology*, Vol. 16(3), Springer-Verlag, New York, 2003, S. 185-215

[BP02] M. Bellare, A. Palacio: „GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks“, *Advances in Cryptology - Crypto 2002*, Lecture Notes in Computer Science, Vol. 2442, Springer-Verlag, Berlin, 2002, S. 162-177

[BNS05] A. Beutelspacher, H. Neumann, T. Schwarzpaul: *Kryptografie in Theorie und Praxis*, Vieweg-Verlag, Wiesbaden, 2005

[BD99] F. Bao, R. H. Deng: „A New Type of "Magic Ink" Signatures - Towards Transcript-Irrelevant Anonymity Revocation“, *Public Key Cryptography: Second International Workshop on Practice and Theory in Public Key Cryptography - PKC 1999*, Lecture Notes in Computer Science, Vol. 1560, Springer-Verlag, Berlin, 1999, S. 1-11

[Br93] S. Brands: „Untracebale Off-line Cash in Wallet with Observers“, *Advances in Cryptology - Crypto 1993*, Lecture Notes in Computer Science, Vol. 773, Springer-Verlag, Berlin, 1994, S. 302-318

[CHL05] J. Camenisch, S. Hohenberger, A. Lysyanskaya: „Compact E-Cash“, Eurocrypt 2005, preprint erhältlich unter
`http://eprint.iacr.org/2005/060` (Stand 09.05.2005)

[CKW04] J. Camenisch, M. Koprowski, B. Warinschi: „Efficient Blind Signatures without Random Oracles“, *Security in Communication Networks: 4th International Conference - SCN 2004*, Lecture Notes in Computer Science, Vol. 3352, S. 134-148. Springer-Verlag, Berlin, 2005

[CL02] J. Camenisch, A. Lysyanskaya: „A Signature with Efficient Protocols“ *Security in Communication Networks: Third International Conference - SCN 2002*, Lecture Notes in Computer Science, Vol. 2576, Springer-Verlag, Berlin, 2003, S. 268-289

[CPS94] J. Camenisch, J. M. Piveteau, M. Stadler: „Blind Signatures Based on the Discrete Logarithm Problem“, *Advances in Cryptology - Eurocrypt 1994*, Lecture Notes in Computer Science, Vol. 950, Springer-Verlag, Berlin, 1995, S.428-432

[CPS95] J. Camenisch, J. M. Piveteau, M. Stadler: „Fair Blind Signatures“, *Advances in Cryptology - Eurocrypt 1995*, Lecture Notes in Computer Science, Vol. 921, Springer-Verlag, Berlin, 1995, S. 209-219

[Ch82] D. Chaum: „Blind signatures for untraceable payments“, *Advances in Cryptology: Proc. of Crypto 1982*, Plenum Press, New York, 1983, S. 199-203

[ChP92] D. Chaum, T. Pedersen: „Wallet Databases with Observers“, *Advances in Cryptology - Crypto 1992*, Lecture Notes in Computer Science , Vol. 740, Springer-Verlag, Berlin, 1993

[CP93] R. Cramer, T. Pedersen: „Improved Privacy in Wallets with Observers“, *Advances in Cryptology - Eurocrypt 1993*, Lecture Notes in Computer Science, Vol. 765, Springer-Verlag, Berlin, 1994, S. 329-343

[DF01] I. Damgård, E. Fujisaki: „An integer commitment scheme based on groups with hidden order“, erhältlich unter
`http://eprint.iacr.org/2001/064` (Stand 09.05.2005)

[Fe93] N. Ferguson: „Single-Term Off-Line Coins“, *Advances in Cryptology - Crypto 1993*, Lecture Notes in Computer Science, Vol. 773, Springer-Verlag, Berlin, 1994, S. 292-301

[FO98] E. Fujisaki, T. Okamoto: „A Practical and Provably Secure Scheme for Publicly Verifiable Secret Sharing and Its Applications “, *Advances in Cryptology - Eurocrypt 1998*, Lecture Notes in Computer Science, Vol. 1403, Springer-Verlag, Berlin, 1998, S. 32-46

[Go03] Oded Goldreich: *Foundations of Cryptography I, Basic Tools*, 2. Auflage, Cambridge University Press, Cambridge, 2003

[Go04] Oded Goldreich: *Foundations of Cryptography II, Basic Applications*, Cambridge University Press, Cambridge, 2004

[GB01] S. Goldwasser, M. Bellare: „Lecture Notes on Cryptography ", Skript, erhältlich unter
`http://www.cs.ucsd.edu/users/mihir/papers/gb.html` (Stand 09.05.2005)

[HMP94a] P. Horster, M. Michels, H. Petersen: „Meta-ElGamal signature schemes", *Proc. of the 2nd ACM Conference on Computer and Communications Security*, ACM Press, New-York, 1994, S. 86-107

[HMP94b] P. Horster, M. Michels, H. Petersen: „Meta-Message Recovery and Meta-Blind Signature Schemes Based on the Discrete Logarithm Problem and Their Applications", *Advances in Cryptology - Asiacrypt 1994*, Lecture Notes in Computer Science, Vol. 917, Springer-Verlag, Berlin, 1994, S. 224-237

[HMP95] P. Horster, M. Michels, H. Petersen: „Hidden signature schemes based on the discrete logarithm problem and related concepts" , Technical Report TR-94-10 , University of Technology Chemnitz-Zwickau, Chemnitz, 1995

[HP94] P. Horster, H. Petersen: „Classification of blind signature schemes and examples of hidden and weak blind signatures" , Technical Report TR-94-1-E, University of Technology Chemnitz-Zwickau, Chemnitz, 1994

[K04] M. Krüger: „Internet-Zahlungssysteme aus der Sicht der Verbraucher. Ergebnisse der Online-Umfrage IZV7", 2004, erhältlich unter
`http://www.iww.uni-karlsruhe.de/izv/pdf/izv7_auswertung.pdf` (Stand 09.05.2005)

[JLO97] A. Juels, M. Luby, R. Ostrovsky: „Security of blind digital signatures", *Advances in Cryptology - Crypto 1997*, Lecture Notes in Computer Science, Vol. 1294, Springer-Verlag, Berlin, 1997, S. 150-164

[Ly02] A. Lysyanskaya: „Signature Schemes and Applications to Cryptographic Protocol Design", Dissertation, Massachusetts Institute of Technology, 2002, erhältlich unter
`http://theory.lcs.mit.edu/ cis/cis-theses.html` (Stand 09.05.2005)

[ME81] N. F. G. Martin, J. W. England: *Mathematical Theory of Entropy*, Encyclopedia of Mathematics and its Applications, Addison-Wesley, London, 1981

[M77] R. J. McEliece: *The Theory of Information and Coding*, Encyclopedia of Mathematics and its Applications, Addison-Wesley, London, 1977

[MvOV97] A. Menezes, P. van Oorschot, S. Vanstone: *Applied Cryptography*, CRC Press, 1997

[N99] H. Neumann: „Sicherheit in Bezahlsystemen auf der Basis von digitalen Münzen“ , Dissertation, Justus-Liebig-Universität Gießen, erschienen in den *Mitteilungen aus dem Mathematischen Seminar Gießen*, Gießen, 2000

[Oka92] T. Okamoto: „Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes“, *Advances in Cryptology - Crypto 1992*, Lecture Notes in Computer Science, Vol. 740, Springer-Verlag, Berlin, 1993, S. 31-53

[PS00] D. Pointcheval, J. Stern: „Security Arguments for Digital Signatures and Blind Signatures“, *Journal of Cryptology*, Vol. 13(3), Springer-Verlag, New York, 2000, S. 361-396

[R92] S. Roman: *Coding and Information Theory*, Springer-Verlag, New York, 1992

[ST99] T. Sander, A. Ta-Shma: „Auditable, Anonymous Electronic Cash“, *Advances in Cryptology - Crypto 1999*, Lecture Notes in Computer Science, Vol. 1666, Springer-Verlag, Berlin, 1999, S. 555-572

[STY00] T. Sander, A. Ta-Shma, M. Yung: „Blind, Auditable Membership Proofs“, *Financial Cryptography: 4th International Conference - FC 2000*, Lecture Notes of Computer Science 1962, Springer-Verlag, Berlin, 2001, S. 555-572

[S95] L.A.M. Schoenmakers: „An efficient electronic payment system withstanding parallel attacks“, CWI Technical Report CS-R9522, Amsterdam, 1995

[Sch96] B. Schneier: *Applied Cryptography* , 2. Edition, John Wiley & Sons, New York, 1996

[Schn89] C. P. Schnorr: „Efficient Identification and Signatures for Smart Cards“, *Advances in Cryptology - Crypto 1989*, Lecture Notes in Computer Science, Vol. 435, Springer-Verlag, Berlin, 1990, S. 239-252

[Schn91] C. P. Schnorr: „Efficient signature generation by smart cards“, *Journal of Cryptology*, Vol. 4(4), Springer-Verlag, New York, 1991, S. 161-174
Eine frühere Version dieser Arbeit ist [Schn89].

[Wei99] S. Weicker: „Beweisbare Sicherheit digitaler Signaturverfahren im Random Oracle-Modell“, Diplomarbeit, Justus-Liebig-Universität Gießen, 1999