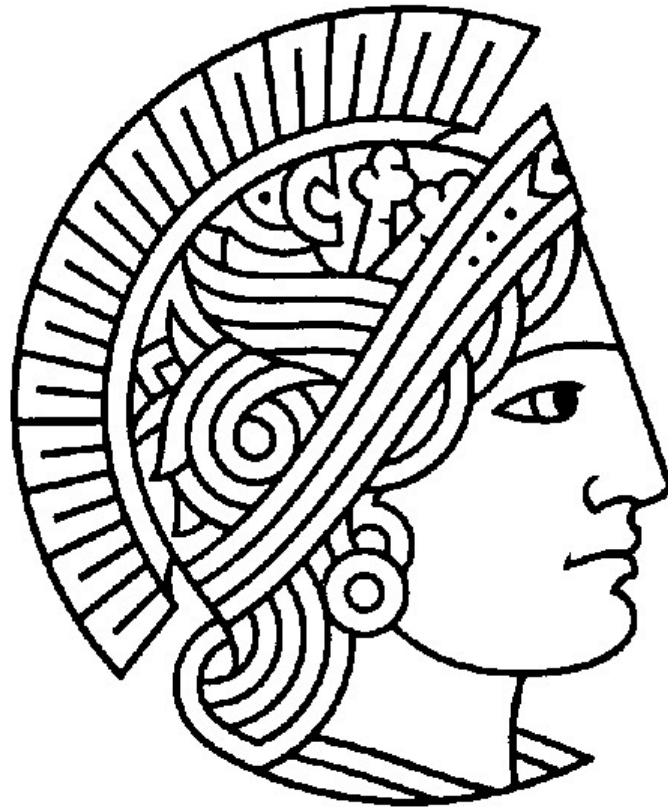


Blinde Signaturen und Post-Quantum-Kryptographie

Bachelorarbeit

zur Erlangung des Grades Bachelor of Science (B.Sc.)
am Fachbereich Informatik der
Technischen Universität Darmstadt



Prüfer	Prof. Dr. J. Buchmann
Betreut von	Lucie Langer
Eingereicht von	Benjamin Kahl

EIDESSTATTLICHE ERKLÄRUNG

Hiermit versichere ich, die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Inhaltsverzeichnis

1	Einführung	4
1.1	Blinde Signaturen	4
1.2	Formulierung der Problemstellung	4
1.3	Recherche	4
1.4	Aufbau der Arbeit und Ergebnisse	5
2	Grundlagen	6
2.1	Sicherheitsziele und Klassifizierung von Fälschungen	6
2.2	Modifiziertes Check Vectors Verfahren	7
2.3	Zero-Knowledge-Beweise	8
2.4	Kryptographische Hashfunktionen	9
2.5	Interaktive Signaturen	10
2.6	Blinde interaktive Signaturen	11
3	Untersuchung der Protokolle	13
3.1	Ein weak blindes Signaturverfahren	13
3.1.1	Weak Signaturverfahren	13
3.1.2	Check Vectors ergeben ein weak Signaturverfahren	13
3.1.3	Blendung des interaktiven Verifikationsprotokolls	14
3.1.4	Blendung des interaktiven Signaturprotokolls	14
3.1.5	Fazit	15
3.2	Verhältnis zwischen digitalen und blinden interaktiven Signaturen	15
3.2.1	Sind alle blinden Signaturverfahren geblendete digitale Signaturverfahren?	16
3.2.2	Lässt sich jedes digitale Signaturverfahren blenden?	17
3.3	Blinde Signaturen und Kryptosysteme	18
3.3.1	Blinde Signaturen nach D. Chaum	18
3.3.2	Blinde (interaktive) Signaturverfahren aus Kryptosystemen	19
3.3.3	Blindes interaktives Signaturverfahren aus einem Kryptosystem (I)	19
3.3.4	Blindes interaktives Signaturverfahren aus zwei Kryptosystemen (II)	21
3.3.5	Blindes interaktives Signaturverfahren: Spezialfall aus II (III)	23
4	Schluss	25
4.1	Ergebnisse	25
4.2	Ausblick	25

1 Einführung

1.1 Blinde Signaturen

Blinde Signaturen, wie sie von D. Chaum in [4] eingeführt wurden, sind eine Erweiterung digitaler Signaturen, bei der die zu signierende Nachricht vor dem Signierer versteckt wird. Da der Signierer die Nachricht, die er signiert, nicht sehen kann, bezeichnet man ihn als blind. Zur Veranschaulichung schlug D. Chaum folgendes Szenario vor: Der Empfänger deckt das zu unterschreibende Blatt mit Kohlepapier ab, steckt dieses in einem Briefumschlag und schickt diesen zum Signierer. Der Signierer unterschreibt den Briefumschlag und schickt ihn zurück an den Empfänger. Der Empfänger öffnet den Briefumschlag und entnimmt sein unterschriebenes Blatt. Erhält der Signierer das Blatt zu einem späteren Zeitpunkt, so kann er keine Aussage darüber treffen, wem er das Blatt unterschrieben hat.

Da blinde Signaturen keine Rückschlüsse auf den Empfänger zulassen, spielen sie vor allem in den Bereichen eine große Rolle, in denen Privatheit¹ und Anonymität im Vordergrund stehen. Die bekanntesten Einsatzgebiete sind elektronisches Geld und elektronische Wahlen. Bei elektronischem Geld möchte der Kunde seiner Bank vorenthalten, wo, bzw. was er gekauft hat. Bei elektronischen Wahlen muss geheim bleiben, wer wie gewählt hat.

1.2 Formulierung der Problemstellung

(Blinde) Signaturverfahren basieren häufig auf zahlentheoretischen Problemen, wie z.B. dem Faktorisierungs²- oder dem diskreten Logarithmusproblem³. Man vermutet, daß diese Probleme von herkömmlichen Rechenmaschinen nicht effizient gelöst werden können. Durch das Aufkommen von Quantencomputern⁴ werden diese Probleme jedoch in gewissem Sinne leicht, d.h. die (blinden) Signaturverfahren weitgehend unbrauchbar. Es stellt sich die Frage, ob es (blinde) Signaturverfahren gibt, die von Quantencomputern nicht gebrochen werden können. Falls dem so sei, stellt sich ferner die Frage, worauf die Sicherheit dieser Verfahren basiert und weiterhin, ob sie auch Angriffen folgender Computergenerationen standhalten. Für digitale Signaturverfahren wurden die Fragen teilweise beantwortet [12]. Im Rahmen dieser Arbeit wird nach Antworten für blinde Signaturverfahren gesucht.

1.3 Recherche

Recherchiert wurde vor allem online. Es wurden herangezogen:

- Veröffentlichungen unter <http://eprint.iacr.org/>,

¹Siehe auch [13]

²<http://de.wikipedia.org/wiki/Faktorisierungsverfahren>

³http://de.wikipedia.org/wiki/Diskreter_Logarithmus

⁴<http://de.wikipedia.org/wiki/Quantencomputer>

- Veröffentlichungen unter <http://www.schneier.com/biblio/>,
- Fach- und Seminararbeiten,
- Kongreßberichte,
- Papers,
- Grundlagenliteratur in der dnb.

1.4 Aufbau der Arbeit und Ergebnisse

Die Arbeit besteht im Wesentlichen aus drei Teilen. Im ersten Teil werden einige Konzepte erläutert, auf die im Rest der Arbeit zurückgegriffen wird. Die Auflistung ist nicht vollständig. Der Autor empfiehlt die Lektüre von [3], [2]. Im zweiten Teil werden Antworten zu den aufgeworfenen Fragen (s.o.) gesucht. Dazu wurden Verfahren gesucht oder entwickelt. Im dritten Teil werden die Ergebnisse zusammengefasst und ein Ausblick gegeben.

Die Frage, ob es künftig quantencomputersichere blinde Signaturverfahren geben wird, konnte mit einem schwachen Ja beantwortet werden; die Beantwortung der anderen Fragen warf weitere Fragen auf, auf die im Verlauf der Arbeit tiefer eingegangen wurde. Eine detaillierte Zusammenfassung der Ergebnisse findet sich im Kapitel 4.

2 Grundlagen

In diesem Kapitel werden, vorbereitend für Kapitel 3, einige kryptographische Primitive und Konzepte eingeführt. Für manche Konzepte und besonders für die in dieser Arbeit betrachteten Protokolle sind drei Teilnehmer erforderlich: Ein Empfänger \mathcal{R} , der Daten von mindestens einem weiteren Teilnehmer anfordert, ein Signierer \mathcal{S} , der Signaturen $\sigma(m)$ zu gegebenen Nachrichten m erstellen kann und ein Verifizierer \mathcal{V} , der Signaturen auf ihre Korrektheit prüft.

2.1 Sicherheitsziele und Klassifizierung von Fälschungen

Durch digitale Signaturen kann eine Reihe von Sicherheitszielen abgedeckt werden. Dazu zählen Authentizität (z.B. können sich Kunden gegenüber einem Dienstleister legitimieren), Integrität (wird ein Dokument nach Signierung verändert, ist die Signatur mit hoher Wahrscheinlichkeit nicht mehr gültig) und Verbindlichkeit (z.B. kann ein Dienstleister nachweisen, daß er einen Auftrag von einem Kunden erhalten hat)⁵.

Durch blinde Signaturen wird ein neues Sicherheitsziel abgedeckt, die Anonymität. Anonymität ist der Wunsch des Empfängers mit einer abgeschlossenen Aktion nicht mehr in Verbindung gebracht werden zu können.

Von jedem Signaturverfahren versucht man zu bestimmen, ob und wie es die Sicherheitsziele erfüllt. Eine Möglichkeit dies herauszufinden, ist das Verfahren unterschiedlich starken Angriffen auszusetzen und die jeweiligen Erfolgsaussichten zu bestimmen. Man unterscheidet i.A. zwischen vier Angriffstypen und vier Erfolgsstufen.

Die Angriffstypen sind⁶:

- **Angriff ohne bekannte Signaturen:** Dem Angreifer sind keine Nachricht-Signatur-Paare bekannt. Er hat nur Zugriff auf das Verifikationsverfahren.
- **Angriff mit bekannten Signaturen:** Dem Angreifer werden zufällig ausgewählte Nachricht-Signatur-Paare zu Verfügung gestellt. Er hat weiterhin Zugriff auf das Verifikationsverfahren.
- **Angriff mit gewählten Nachrichten:** Der Angreifer darf zufällige Nachricht-Signatur-Paare erzeugen lassen. Er hat weiterhin Zugriff auf das Verifikationsverfahren.
- **Adaptiver Angriff mit gewählten Nachrichten:** Der Angreifer darf mehrere Nachricht-Signatur-Paare erzeugen lassen. Die Paare dürfen in beliebiger Weise voneinander abhängen. Er hat weiterhin Zugriff auf das Verifikationsverfahren.

Die Erfolgsstufen sind:

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

⁵Eine vollständige Auflistung findet sich in [13].

⁶Siehe auch [10], [9] und [6]. Die Darstellung wurde an [6] angelehnt.

- **Selektive Fälschbarkeit:** Der Angreifer kann Signaturen zu einigen von ihm gewählten Nachrichten fälschen.
- **Universelle Fälschbarkeit:** Der Angreifer kann die Signatur einer beliebigen Nachricht fälschen, kennt aber den geheimen Schlüssel nicht.
- **Kompromittierung des Schlüssels:** Der Angreifer ist in der Lage den geheimen Schlüssel des Signierers zu berechnen.

2.2 Modifiziertes Check Vectors Verfahren

Das Check Vectors Verfahren wurde von T. Rabin in [11] eingeführt. Es handelt sich um ein Protokoll mit drei Teilnehmern: Einem Erzeuger \mathcal{D} , einem Zwischengeschalteten \mathcal{INT} und einem Empfänger \mathcal{R} . \mathcal{D} kennt einen Wert s , den er \mathcal{R} via \mathcal{INT} übermitteln möchte. \mathcal{R} akzeptiert \hat{s} , also den von \mathcal{INT} erhaltenen Wert, wenn er überzeugt ist, daß $s = \hat{s}$. Das Protokoll wurde so konstruiert, daß es beweisbar folgende drei Eigenschaften hat:

- \mathcal{R} verwirft jeden Wert \hat{s} , der nicht von \mathcal{D} stammt, sofern \mathcal{D} ehrlich ist,
- \mathcal{R} hat keine Information über s , bis er s von \mathcal{INT} erhalten hat, sofern \mathcal{D} und \mathcal{INT} ehrlich sind und
- \mathcal{INT} weiß von einem Wert s ob \mathcal{R} ihn akzeptieren wird, unabhängig davon, ob \mathcal{D} ehrlich ist oder nicht.

Das Protokoll verläuft wie folgt:

1. Sei $k, p \in \mathbb{N}$. $\mathcal{D}(s)$ wählt zufällig $b_1, y_1, \dots, b_{2k}, y_{2k} \in \mathbb{Z}_p$,
2. \mathcal{D} berechnet $c_i = s * b_i + y_i$ f.a. $1 \leq i \leq 2k$,
3. \mathcal{D} sendet an \mathcal{INT} : s, y_1, \dots, y_{2k} ,
4. \mathcal{D} sendet an \mathcal{R} : $b_1, c_1, \dots, b_{2k}, c_{2k}$ (modifizierter Check Vector),
5. \mathcal{INT} wählt zufällig k Indizes d_1, \dots, d_k mit $1 \leq d_i \leq 2k$ aus und bittet \mathcal{R} ihm die Werte $b_{d_1}, c_{d_1}, \dots, b_{d_k}, c_{d_k}$ zu offenbaren,
6. \mathcal{R} veröffentlicht die Werte und \mathcal{INT} prüft, ob für alle d_i die Gleichung $s * b_{d_i} + y_{d_i} = c_{d_i}$ erfüllt ist. Ist dies der Fall, kann \mathcal{INT} davon ausgehen, daß \mathcal{R} seinen Wert s akzeptieren wird und übermittelt s, y_1, \dots, y_{2k}

2.3 Zero-Knowledge-Beweise

Ein Zero-Knowledge-Beweis ist ein Protokoll, bei dem ein Beweiser \mathcal{P} einem Verifizierer \mathcal{V} die Kenntnis eines Geheimnisses beweist, wobei nach [15] fünf Eigenschaften erfüllt sein müssen:

- Es soll *kein Wissenstransfer* stattfinden, d.h. \mathcal{V} darf während des Beweises keine Informationen über das Geheimnis ableiten dürfen,
- die Behauptung soll *korrekt* sein, d.h. \mathcal{P} darf \mathcal{V} nur überzeugen können, wenn er im Besitz des Geheimnisses ist,
- das Protokoll soll *robust* sein, d.h. \mathcal{V} darf auch durch vom Protokoll abweichendes Verhalten keine Informationen über das Geheimnis ableiten dürfen,
- das Protokoll muss mindestens eine *nicht statische* Komponente haben, damit der Beweisversuch von \mathcal{V} gegenüber einer dritten Partei scheitert (Man-in-the-middle-Angriffe sind davon ausgenommen),
- das Protokoll muss *vollständig* sein, d.h. \mathcal{V} muss von jeder wahren Aussage überzeugt werden können.

Beispiel 2.1 (Tartaglias Zero-Knowledge-Beweis⁷). *1535 kannte Niccolò Tartaglia, (1499-1557) ein Verfahren zur Lösung kubischer Gleichungen. Dies wollte er Antonio Maria Fiore gegenüber beweisen, ohne das Verfahren zu veröffentlichen. Fiore konstruierte daraufhin 30 kubische Gleichungen, die er Tartaglia zur Lösung vorlegte. Tartaglia löste die Gleichungen und bewies damit die Kenntnis eines Verfahrens. Dieses Beweisverfahren erfüllt die Eigenschaften eines Zero-Knowledge-Beweises. Fiore konnte nicht auf das Lösungsverfahren schließen, Tartaglia musste ein Verfahren kennen, da er die Gleichungen sonst nicht hätte lösen können, Fiore konnte das Verfahren nicht ohne Einverständnis von Tartaglia in Erfahrung bringen, Fiore konnte den Beweis nicht gegenüber einer dritten Partei führen, da er damit rechnen musste, andere Gleichungen vorgelegt zu bekommen, die er nicht lösen kann und Tartaglia konnte zu einer beliebigen Gleichung der Form $ax^3 + bx^2 + cx + d = 0$ eine Lösung angeben.*

Um Zero-Knowledge-Beweise formal definieren zu können, benötigt man den Begriff des interaktiven Beweissystems.

Definition 2.3.1 (Interaktive Beweissysteme und die Klasse IP^8). *Ein interaktives Beweissystem für eine Menge M ist ein Spiel zwischen zwei Beteiligten, einem Verifizierer \mathcal{V} , der eine probabilistische polynomiell-zeitbeschränkte Strategie und einem Beweisführer \mathcal{P} , der eine Strategie ohne Berechnungseinschränkungen verfolgt, das zwei Bedingungen erfüllt:*

1. *Vollständigkeit: Für jedes $x \in M$ akzeptiert \mathcal{V} nach dem Dialog mit \mathcal{P} bei gemeinsamer Eingabe x .*

⁷Siehe Wikipedia, [1]

⁸Siehe auch [1].

2. *Zuverlässigkeit*: Es existieren Polynome p , so daß für jedes $x \notin M$ und jede polynomiell-zeitbeschränkte Strategie P^* , \mathcal{V} mit einer Wahrscheinlichkeit von mindestens $\frac{1}{p(|x|)}$ nach der Interaktion mit P^* bei gemeinsamer Eingabe x die Behauptung $x \in M$ ablehnt.

Die Klasse von Problemen mit interaktiven Beweissystemen heißt *IP*.

Durch $O(p(|x|)^2)$ -maliges sequentielles Wiederholen eines Beweises kann man die Wahrscheinlichkeit, daß \mathcal{V} eine falsche Behauptung akzeptiert, von $1 - \frac{1}{p(|x|)}$ auf $2^{-p(|x|)}$ reduzieren.

Definition 2.3.2 (Zero-Knowledge-Beweise⁹). Sei $(\mathcal{P}, \mathcal{V})$ ein interaktives Beweissystem für eine Sprache \mathcal{L} . $(\mathcal{P}, \mathcal{V})$ hat die Zero-Knowledge-Eigenschaft, falls für jede probabilistische polynomiell-zeitbeschränkte interaktive Maschine \mathcal{V}^* ein probabilistischer polynomiell-zeitbeschränkter Algorithmus \mathcal{M} existiert, so daß für jedes $x \in \mathcal{L}$ die beiden folgenden Zufallsvariablen gleich verteilt sind:

- $\langle \mathcal{P}, \mathcal{V}^* \rangle(x)$, also die Ausgabe der interaktiven Maschine \mathcal{V}^* nach der Interaktion mit der interaktiven Maschine \mathcal{P} bei gemeinsamer Eingabe x ,
- $\mathcal{M}(x)$, also die Ausgabe des Algorithmus \mathcal{M} bei Eingabe x .

Der Algorithmus \mathcal{M} wird Simulator für die Interaktion von \mathcal{V}^* mit \mathcal{P} genannt.

2.4 Kryptographische Hashfunktionen

Eine Hashfunktion ist eine Funktion, die jedem Element eines potentiell unendlichen Definitionsbereiches ein Element eines endlichen Wertebereiches zuordnet.

Definition 2.4.1 (Hashfunktion). Sei $n \in \mathbb{N}$. Eine Abbildung $\mathcal{H} : \{0, 1\}^* \rightarrow \{0, 1\}^n$ heißt Hashfunktion, 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 von einer Hashfunktion meist zusätzlich die Kollisionsresistenz und die Einwegigkeit.

Definition 2.4.2 (Kollisionsresistente Hashfunktion). Unter einer kollisionsresistenten Hashfunktion versteht man eine Hashfunktion \mathcal{H} für die gilt:

Es gibt keinen effizienten Algorithmus \mathcal{A} , der zwei Werte $m \neq \hat{m}$ mit $\mathcal{H}(m) = \mathcal{H}(\hat{m})$ findet.

Definition 2.4.3 (Einweg-Hashfunktion). Unter einer Einweg-Hashfunktion versteht man eine Hashfunktion \mathcal{H} für die gilt:

Es gibt keinen effizienten Algorithmus, der bei Eingabe eines zufällig gewählten $y \in \{0, 1\}^n$ ein m mit $y = \mathcal{H}(m)$ ausgibt.

⁹Siehe auch [1], [6]

2.5 Interaktive Signaturen

In [6] wurden interaktive Signaturen eingeführt. Interaktive Signaturen erweitern digitale Signaturen um ein Signaturprotokoll σ_I und ein Verifikationsprotokoll V_I .

In σ_I wird festgelegt, wie die Signatur zu erstellen ist. D.h. für jeden Teilnehmer wird festgelegt:

- wann er welchen Wert erhält,
- welche Operationen er auf diesem Wert auszuführen hat und
- an wen er welche ggf. berechneten Werte zu senden hat.

In V_I wird festgelegt, wie eine gegebene Signatur zu verifizieren ist. Auch hier werden den einzelnen Teilnehmern Operationen und Kommunikationsschritte, sowie deren Zeitpunkte zugeordnet.

Beispiel 2.2 (Die interaktive RSA-Signatur). *Die interaktive RSA-Signatur ist eine Erweiterung der RSA-Signatur. Die RSA-Signatur funktioniert wie folgt:*

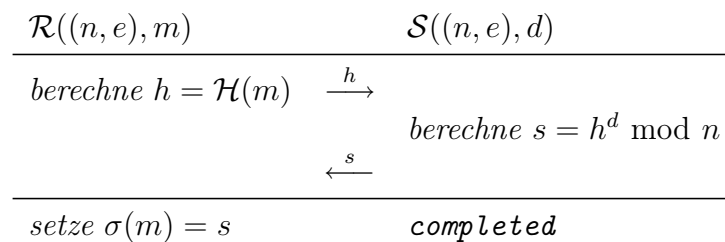
Anfangs wird ein öffentlicher ($p_k = (n, e)$) und ein privater ($s_k = ((n, e), d)$) Schlüssel generiert. Dazu werden Primzahlen p und q für $n = p \cdot q$, sowie ein e mit $\gcd(e, \varphi(n)) = 1$ gewählt. Ferner wird ein $d \equiv e^{-1} \pmod{\varphi(n)}$ berechnet.

Die Signatur $\sigma(m)$ der Nachricht m ist das kleinste, nichtnegative s , das die Kongruenz $s \equiv m^d \pmod{n}$ erfüllt.

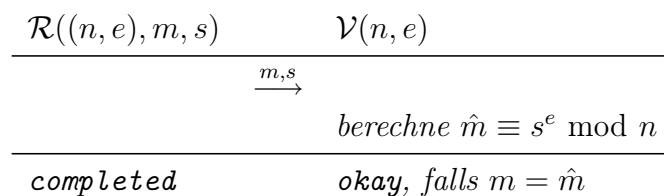
Ein Verifizierer akzeptiert s als Signatur von m , wenn die Kongruenz $m \equiv s^e \pmod{n}$ erfüllt ist.

Um daraus eine interaktive RSA-Signatur zu konstruieren, muss die RSA-Signatur um ein Signaturprotokoll σ_I und ein Verifikationsprotokoll V_I erweitert werden.

Erweiterung um σ_I :



Erweiterung um V_I :



Definition 2.5.1 (Interaktive Signatureschemata¹⁰). *Es sei $k \in \mathbb{N}$ ein Sicherheitsparameter, 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 folgt beschrieben.*

- *G sei ein probabilistischer, polynomieller Algorithmus, der bei Eingabe von 1^k ein Paar (p_k, s_k) ausgibt. Wir schreiben $(p_k, s_k) \in G(1^k)$ und nennen p_k öffentlichen Schlüssel sowie s_k geheimen Schlüssel.*
- *σ_I sei ein Protokoll mit polynomieller Rundenzahl zwischen \mathcal{S} mit Eingabe (p_k, s_k) und \mathcal{R} mit Eingaben p_k 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} *completed*, sonst *not completed*.*
- *V_I sei ein Protokoll mit polynomieller Rundenzahl zwischen \mathcal{R} mit Eingabe $(p_k, m, \sigma_I(m))$ und \mathcal{V} mit Eingabe (p_k) . Nach der Durchführung von V_I sei die Ausgabe von \mathcal{V} entweder *ok* 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 Signatureschema oder interaktive Signatur, falls die Protokolle σ_I und V_I für alle $(p_k, s_k) \in G(1^k)$ 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 $(p_k, s_k) \in G(1^k)$ ist im Protokoll σ_I die Ausgabe von \mathcal{R} mit Eingabe (p_k, 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*.

2.6 Blinde interaktive Signaturen

Man unterscheidet zwischen zwei Arten von blinden interaktiven Signaturen:

1. Interaktive Signaturen mit geblendetem Signaturprotokoll und
2. interaktive Signaturen mit geblendetem Verifikationsprotokoll.

Bei interaktiven Signaturverfahren mit geblendetem Signaturprotokoll lässt sich Empfänger \mathcal{R} von Signierer \mathcal{S} zu einer Nachricht m eine Signatur $\sigma(m)$ generieren, ohne daß \mathcal{S} auf m oder $\sigma(m)$ schließen kann.

¹⁰Entnommen aus [6].

Bei interaktiven Signaturen mit geblendetem Verifikationsprotokoll ist Empfänger \mathcal{R} im Besitz eines Nachricht-Signatur-Paares $(m, \sigma(m))$ und möchte Verifizierer \mathcal{V} von dessen Gültigkeit überzeugen, ohne Informationen über $(m, \sigma(m))$ preiszugeben. Hier bietet sich die Verwendung eines Zero-Knowledge-Beweises an.

In [6] wurde ein generisches blindes Signaturverfahren abgeleitet.

Gegeben sei ein interaktives Signaturschema $\Sigma_I = (G, \sigma_I, V_I)$ mit Nachrichtenraum M . Zu jedem $m \in M$ bezeichne $S(m)$ die Menge aller gültigen Signaturen. Ferner bezeichne \mathcal{Z} die Menge der Blendfaktoren und Par eine Menge von Parametern. Um Σ_I blenden zu können wird eine Blendungsfunktion

$$\varphi_{par,z} : \mathcal{M} \rightarrow \mathcal{M}$$

und eine Funktion zum extrahieren der Signaturen

$$\psi_{par,z} : S(\varphi_{par,z}(m)) \rightarrow S(m)$$

benötigt.

Generisches blindes interaktives Signaturprotokoll:

$\mathcal{R}(n, e, m, s)$	$\mathcal{V}(n, e)$
wähle $z \in \mathcal{Z}$	
und berechne $\hat{m} = \varphi_{par,z}(m)$	
	$\xrightarrow{\hat{m}}$
	berechne $\hat{s} = \sigma(\hat{m})$
	$\xleftarrow{\hat{s}}$
berechne $s = \psi_{par,z}(\hat{s})$	

Die Verifikation geschieht über ein Commitment auf die Nachricht m und einem interaktiven Beweis Π .

Generisches blindes interaktives Verifikationsprotokoll:

$\mathcal{R}(m, \sigma(m), p_k, par)$	$\mathcal{V}(p_k, par)$
$c = \text{commitment}(m)$	
	\xrightarrow{c}
	$\xleftrightarrow{\Pi}$

3 Untersuchung der Protokolle

3.1 Ein weak blindes Signaturverfahren

Vielen herkömmlichen blinden Signaturverfahren liegt ein, wie auch immer geartetes, allgemeines Problem zugrunde, das sich in konkreter Ausprägung durch die Kenntnis eines Geheimnisses lösen lässt. Beispiele solcher Probleme sind das Faktorisierungsproblem, das diskrete Logarithmusproblem (DLP), das Diffie-Hellman-Problem (DHP) in den vielfältigsten Variationen (Gap-DHP, GDHP mit bilinearen Abbildungen¹¹) und derer mehr.

Da in dieser Arbeit nach Bausteinen für blinde (interaktive) Signaturverfahren gesucht wird, stellt sich als Erstes die Frage, ob ein solches Problem für die Konstruktion eines Verfahrens notwendig ist.

In [5] leiteten M. Franklin und M. Yung ein Verfahren zur Erstellung von weak blinden Signaturverfahren ab, das auf dem Check Vectors Verfahren basiert. Sie erhoben den Anspruch *ohne* kryptographische Primitive wie z.B. Hashfunktionen auszukommen. Insbesondere basiert das Verfahren *nicht* auf einem zahlentheoretischen o.ä. gearteten Problem. Dazu wurde ein sogenanntes nicht-kryptographisches Szenario entworfen. In diesem Szenario wurden Unlösbarkeitsannahmen, wie z.B. die RSA-Annahme, durch Annahmen an das darunterliegende Kommunikationsmodell ersetzt.

Im konkreten Fall wurden sichere Kommunikationskanäle vorausgesetzt. Ein sicherer Kommunikationskanal wird hier als Ende-zu-Ende-Verbindung zwischen zwei Teilnehmern \mathcal{A} und \mathcal{B} verstanden, die von einem Angreifer \mathcal{C} nicht kompromittiert werden kann.

3.1.1 Weak Signaturverfahren

Bei einem weak Signaturverfahren wird ein *vertrauenswürdiger, immer verfügbarer*, vom Signierer *separierter* Teilnehmer (Checking Center) \mathcal{C} und sichere Kommunikationskanäle zwischen allen Kommunikationspartnern vorausgesetzt. \mathcal{C} nimmt sowohl am Signier-, als auch am Verifikationsvorgang teil, beschränkt sich dabei jedoch darauf, Werte von der Signierstelle zu empfangen, zu speichern und ggf. Linearkombinationen daraus zu berechnen. Die sicheren Kommunikationskanäle stellen sicher, daß sich ein Angreifer nicht als \mathcal{C} oder \mathcal{S} ausgeben kann.

3.1.2 Check Vectors ergeben ein weak Signaturverfahren

Das Check Vectors Verfahren wurde in Abschnitt 2.2 beschrieben. Im Signaturverfahren sei \mathcal{INT} nun \mathcal{R} , also derjenige Teilnehmer, der sich eine Nachricht m signieren lassen möchte. Ferner übernimmt \mathcal{S} die Rolle des Senders \mathcal{D} und \mathcal{C} die Rolle des Empfängers \mathcal{R} . Für das Signaturverfahren wird $k = 1$ gewählt.

Das interaktive Signaturprotokoll σ_I verläuft ähnlich den Schritten (1)-(4) im Check Vectors Verfahren:

¹¹Eine gute Darstellung der Probleme und ein Beispiel findet sich unter [8]

1. \mathcal{R} sendet an \mathcal{S} : m
2. Sei $p \in \mathbb{N}$. \mathcal{S} wählt zufällig $b_1, y_1, b_2, y_2 \in \mathbb{Z}_p$,
3. \mathcal{S} berechnet $c_i = s * b_i + y_i$ f.a. $1 \leq i \leq 2$,
4. \mathcal{S} sendet an \mathcal{INT} : s, c_1, c_2 ,
5. \mathcal{S} sendet an \mathcal{C} : b_1, y_1, b_2, y_2 .

Nach Ablauf des Protokolls liegt folgende Konfiguration vor:

- \mathcal{S} hat drei Vektoren $\vec{c}, \vec{b}, \vec{y} \in \mathbb{Z}_p^2$ derart konstruiert, daß die Gleichung $\vec{c} = m * \vec{b} + \vec{y}$ erfüllt ist,
- \mathcal{R} hat von \mathcal{S} den Vektor \vec{y} empfangen und hält mit $(m, \sigma(m)) = (m, \vec{c})$ das Nachricht-Signatur-Paar,
- \mathcal{C} hat von \mathcal{S} die Vektoren \vec{b}, \vec{y} empfangen und hält damit die Verifikationsinformation für das Nachricht-Signatur-Paar.

Zum Verifikationsprotokoll V_I : Möchte \mathcal{V} die Signatur prüfen, erfragt er von \mathcal{C} die Verifikationsinformation \vec{b}, \vec{y} und prüft die Gleichheit $\vec{c} = m * \vec{b} + \vec{y}$.

3.1.3 Blendung des interaktiven Verifikationsprotokolls

Die Blendung von V_I wird dadurch erreicht, daß die Verifikationsinformation, die \mathcal{C} von \mathcal{S} erhält, modifiziert wird. \mathcal{R} generiert zwei Blendvektoren $\Delta b, \Delta y \in \mathbb{Z}_p^2$ und sendet diese an \mathcal{C} . \mathcal{C} berechnet $\hat{b} = b + \Delta b \bmod p$, $\hat{y} = y + \Delta y \bmod p$. Die neue Verifikationsinformation ist (\hat{b}, \hat{y}) . \mathcal{R} berechnet $\hat{c} = \vec{c} + s * \Delta b + \Delta y$. Das neue Nachricht-Signatur-Paar ist (m, \hat{c}) . \mathcal{S} kann, sollte er später als Verifizierer \mathcal{V} auftreten, nur schwer auf das ursprüngliche \vec{c} schließen.

3.1.4 Blendung des interaktiven Signaturprotokolls

Die Blendung von σ_I wird dadurch erreicht, daß die Nachricht, die \mathcal{R} an \mathcal{S} sendet, geblendet wird. \mathcal{R} wählt zufällig $r \in \mathbb{Z}_p^*$, berechnet $\hat{m} = r * m \bmod p$ und läßt \hat{m} von \mathcal{S} signieren.

Nach dem Protokoll ist folgender Zustand erreicht:

- \mathcal{S} hat drei Vektoren $\vec{b}, \vec{y}, \vec{c} \in \mathbb{Z}_p^2$ generiert, so daß die Gleichung $\vec{c} = \hat{m} * \vec{b} + \vec{y} = r * m * \vec{b} + \vec{y}$ erfüllt ist,
- \mathcal{R} kennt die Nachricht m , die geblendete Nachricht \hat{m} und eine dazugehörige Signatur $\sigma(\hat{m}) = \vec{c}$,
- \mathcal{C} kennt die Verifikationsvektoren \vec{b}, \vec{y} zu der Nachricht \hat{m} .

\mathcal{R} und \mathcal{C} konstruieren aus der Signatur für \hat{m} eine Signatur für m . \mathcal{R} sendet den Blendfaktor r an \mathcal{C} . \mathcal{C} berechnet $\hat{b} = r * \vec{b} \bmod p$ und speichert \hat{b}, \vec{y} als Verifikationsinformation. Bei der Verifikation wird geprüft, ob die Gleichung $\vec{c} = m * \hat{b} + \vec{y}$ erfüllt ist. \mathcal{S} kann, sollte er später als Verifizierer \mathcal{V} auftreten, nur schwer auf das ursprüngliche m schließen.

3.1.5 Fazit

Jeder Angreifer kann selbst Signaturen erstellen, da dazu nur ein überbestimmtes Gleichungssystem um die Nachricht konstruiert werden muss. Beim Verifikationsvorgang hat ein Angreifer jedoch keine Möglichkeit zu betrügen, da dem System ein vertrauenswürdiger Teilnehmer \mathcal{C} und sichere Kommunikationskanäle zugrunde liegen.

M. Franklin und M. Yung haben ein Signaturverfahren konstruiert, das zwar *ohne kryptographische Annahmen* und *ohne zahlentheoretische o.ä. Probleme* auskommt, dafür andere Probleme (Wie lässt sich ein sicherer Kommunikationskanal ohne kryptographische Methoden realisieren?) aufwirft. Weiterhin wurde viel Freiraum für Implementationsdetails gelassen. Beispielsweise stellt sich bei mehreren Signaturen die Frage, wie einem Paar $(m, \sigma(m))$ die korrespondierende Verifikationsinformation \vec{b}, \vec{y} zugeordnet werden kann.

Das Ergebnis wird nun in einem Satz festgehalten:

Satz 3.1.1. *Für ein weak blindes Signaturverfahren sind hinreichend:*

1. *Ein vertrauenswürdiger (on-line) Server (Checking Center) und*
2. *sichere Kommunikationskanäle.*

Bei der Konstruktion des Verfahrens fällt auf, daß erst ein weak Signaturverfahren konstruiert wurde, das anschließend im Verifikations-, als auch im Signaturprotokoll geblendet wurde. Ein ähnliches Vorgehen liegt auch der blinden RSA-, bzw. der blinden Schnorr-Signatur zugrunde. Diese Beobachtung motiviert die Fragen:

Ausblick

Ist jedes blinde Signaturverfahren ein geblendetes digitales Signaturverfahren?

Lässt sich aus jedem digitalen Signaturverfahren ein blindes Signaturverfahren erstellen?

Falls nein, welche Eigenschaften müssen $\varphi_{par,z}(m)$ und $\psi_{par,z}(\sigma(m))$ erfüllen, damit σ_I und/oder V_I geblendet werden kann?

3.2 Verhältnis zwischen digitalen und blinden interaktiven Signaturen

Dieser Abschnitt versucht die im letzten Abschnitt aufgeworfene Frage der Blendbarkeit digitaler Signaturverfahren zu beantworten. Um Signier- und Verifikationsvorgang

getrennt behandeln zu können, werden in diesem Abschnitt hauptsächlich interaktive Signaturverfahren untersucht. Dies stellt keine Einschränkung dar, da nach [6] jedes (blinde) Signaturverfahren in ein (blindes) interaktives Signaturverfahren überführt werden kann.

3.2.1 Sind alle blinden Signaturverfahren geblendete digitale Signaturverfahren?

Mit dem folgenden Satz wird die erste Frage beantwortet:

Satz 3.2.1. *Jedes blinde interaktive Signaturverfahren ist ein geblendetes digitales interaktives Signaturverfahren.*

Proof. Es reicht zu zeigen, daß es zu jedem blinden interaktiven Signaturverfahren $B = (G_B, \sigma_{I_B}, V_{I_B})$ ein digitales interaktives Signaturverfahren $S = (G_S, \sigma_{I_S}, V_{I_S})$ gibt, aus dem B hätte hervorgehen können.

$\sigma_{I_B} \Rightarrow \sigma_{I_S}$:

Fall 1: σ_{I_B} ist blind, d.h. \mathcal{S} erhält keine Informationen über m oder $\sigma(m)$. Ein gültiges σ_{I_S} könnte so aussehen, daß \mathcal{R} die Nachricht m an \mathcal{S} sendet, welcher σ_{I_B} mit sich selbst ausführt und das resultierende $\sigma(m)$ an \mathcal{R} sendet.

Fall 2: σ_{I_B} ist nicht blind, d.h. $\sigma_{I_S} = \sigma_{I_B}$.

$V_{I_B} \Rightarrow V_{I_S}$:

Fall 1: V_{I_B} ist blind, d.h. \mathcal{V} erhält keine Informationen über m oder $\sigma(m)$. Ein gültiges V_{I_S} könnte so aussehen, daß \mathcal{R} die Nachricht m und die Signatur $\sigma(m)$ an \mathcal{V} sendet, welcher V_{I_B} mit sich selbst ausführt und die Signatur akzeptiert, bzw. verwirft.

Fall 2: V_{I_B} ist nicht blind, d.h. $V_{I_S} = V_{I_B}$. □

Korollar 3.2.1. *Die Existenz eines digitalen interaktiven Signaturverfahrens ist Voraussetzung für die Existenz eines blinden interaktiven Signaturverfahrens.*

Die Entwicklung eines blinden interaktiven Signaturverfahrens kann demnach in zwei Schritten erfolgen:

1. Entwickeln eines digitalen interaktiven Signaturverfahrens $S = (G, \sigma_I, V_I)$ und anschließend
2. prüfen, ob sich S durch Blendung in ein digitales interaktives Signaturverfahren überführen lässt.

Aus Satz 3.2.1 geht hervor, daß jedes blinde interaktive Signaturverfahren mit dem o.g. Verfahren gefunden werden kann. Ließe sich die Frage, ob jedes digitale interaktive Signaturverfahren geblindet werden kann ebenfalls mit Ja beantworten, wäre gezeigt, daß blinde interaktive Signaturverfahren *nicht* komplexer sind als digitale interaktive Signaturverfahren. Ließe sich weiterhin ein generisches Verfahren zur Blendung digitaler interaktiver Signaturverfahren angeben, wäre eine Eigenschaft gefunden, die blinde interaktive Signaturverfahren charakterisiert.

3.2.2 Lässt sich jedes digitale Signaturverfahren blenden?

Im Folgenden wird überlegt, ob es ein generisches Verfahren zur Blendung geben kann.

Wie in Kapitel 2 beschrieben, erweitern blinde interaktive Signaturen die Schutzziele digitaler interaktiver Signaturen um Anonymität. Bisher wurde Anonymität dadurch gewährleistet, daß die Nachricht m für \mathcal{S} unkenntlich gemacht wurde. Diese konnte dadurch keinem konkreten Signiervorgang mehr zugeordnet werden. Eine andere Möglichkeit Anonymität zu gewährleisten besteht darin \mathcal{R} vor \mathcal{S} zu verstecken, bzw. \mathcal{R} nicht direkt mit \mathcal{S} kommunizieren zu lassen. Ausgehend von dieser Idee lässt sich ein weak Schema konstruieren, in dem \mathcal{R} eine Signatur von \mathcal{S} erhält, ohne daß \mathcal{S} bei der Verifikation überprüfen kann, wer die Signatur hat erstellen lassen.

Für das weak Schema werden folgende vier Teilnehmer

- ein Empfänger \mathcal{R} , der eine Nachricht m signieren lassen möchte,
- ein Signierer \mathcal{S} , der Signaturen $s(m)$ zu beliebigen Nachrichten erstellen kann,
- ein Verifizierer \mathcal{V} , der für jedes Paar $(m, s(m))$ entscheiden kann, ob es sich um ein gültiges Nachricht-Signatur-Paar handelt,
- einen *vertrauenswürdigen* Proxy¹² \mathcal{P}_S zu \mathcal{S} ,

zwei digitale Signaturverfahren und ein asym. Verschlüsselungsverfahren mit Verschlüsselungsfunktion E_k und Entschlüsselungsfunktion D_k benötigt. Für die Signaturverfahren gilt: Die Signaturen werden generiert, indem s_1, s_2 auf die Nachricht m angewandt wird.

Das weak blinde interaktive Signaturprotokoll σ_B :

1. \mathcal{R} wählt einen Schlüssel (pk, sk) und sendet (m, pk) an \mathcal{P}_S ,
2. \mathcal{P}_S sendet (m, pk) an \mathcal{S} ,
3. \mathcal{S} generiert $\sigma(m) = s_1(m)$ und verschlüsselt diese mit pk , d.h. berechnet $s_1^{pk} = E_{pk}(s_1(m))$. s_1^{pk} wird anschließend mit s_2 signiert, d.h. es wird $s_1^{pk}_2 = s_2(s_1^{pk})$ generiert,
4. \mathcal{S} sendet $(s_1^{pk}, s_1^{pk}_2)$ an \mathcal{P}_S ,
5. \mathcal{P}_S leitet $(s_1^{pk}, s_1^{pk}_2)$ an \mathcal{R} weiter und
6. \mathcal{R} prüft, ob $s_1^{pk}_2$ eine gültige Signatur von s_1^{pk} bzgl. S_2 ist. Falls ja kann $\sigma(m) = D_{sk}(s_1^{pk}_2)$ extrahiert werden. Das gültige Nachricht-Signatur-Paar ist $(m, \sigma(m))$.

¹²Die Idee einen Proxy zu verwenden wurde digitalen Signaturverfahren entlehnt. Ein Beispiel findet sich unter [7]. Grundlage dafür bilden die von Shamir in [14] eingeführten ID-basierten Signaturverfahren.

Durch die Verschlüsselung wird sichergestellt, daß \mathcal{P}_S die Signatur nicht lesen kann. Der verschlüsselte Wert wird signiert, damit \mathcal{R} sicher sein kann, daß $\sigma(m)$ wirklich von \mathcal{S} verschlüsselt wurde und es sind zwei Signaturverfahren notwendig, da \mathcal{R} sonst zwei Signaturen pro Signiervorgang erhalten würde.

Bekommt \mathcal{S} das Nachricht-Signatur-Paar zu sehen, weiß er, daß m für \mathcal{P}_S signiert wurde. Er weiß allerdings nicht, an wen m weitergeleitet wurde.

Das interaktive Verifikationsprotokoll V_I :

1. \mathcal{R} sendet $(m, \sigma(m))$ an \mathcal{V} ,
2. \mathcal{V} prüft die Gültigkeit von $\sigma(m)$.

Hier wird von einem ehrlichen Teilnehmer \mathcal{V} ausgegangen. Sollte dies nicht gegeben sein, könnte \mathcal{P}_S auch im Verifikationsprotokoll eingesetzt werden. Das Protokoll verliefte ähnlich dem Signaturprotokoll. \mathcal{P}_S fungiert als Bote, bzw. Treuhänder.

Das Ergebnis wird nun in einem Satz festgehalten:

Satz 3.2.2. *Aus jedem digitalen interaktiven Signaturverfahren S lässt sich ein weak blindes interaktives Signaturverfahren erstellen, wenn*

- *es eine weitere, von S verschiedene Signaturmethode und*
- *ein asymmetrisches Verschlüsselungsverfahren gibt.*

Bedauerlicherweise wurde selbst nach intensiver Recherche nicht genügend Material gefunden, um ein Verfahren konstruieren zu können, mit dem sich digitale interaktive Signaturverfahren in „starke“ blinde interaktive Signaturenverfahren überführen lassen.

3.3 Blinde Signaturen und Kryptosysteme

3.3.1 Blinde Signaturen nach D. Chaum

Im weiteren Verlauf der Arbeit spielen blinde Signaturen nach D. Chaum [4] eine zentrale Rolle. Er forderte

- eine Signaturfunktion s' , die nur \mathcal{S} bekannt ist und die korrespondierende, öffentlich bekannte Inverse s derart, daß $s(s'(x)) = x$, wobei s keine Information über s' preisgibt,
- zwei nur \mathcal{R} bekannte Funktionen c und ihre Inverse c' , so daß $c'(s'(c(x))) = s'(x)$, wobei $c(x)$ und $s'(x)$ keine Informationen über x preisgeben und
- ein redundanzüberprüfendes Prädikat r , das die Suche nach gültigen Signaturen möglich macht.

Weiterhin erstellte er ein Protokoll, das den Ablauf des blinden Signaturverfahrens wiedergibt. Es besteht aus vier Schritten:

1. \mathcal{R} wählt ein zufälliges x , so daß $r(x)$. Er berechnet $c(x)$ und sendet $c(x)$ an \mathcal{S} .

2. \mathcal{S} signiert $c(x)$ durch Anwendung von s' und sendet den signierten Wert $s'(c(x))$ zurück an \mathcal{R} .
3. \mathcal{R} extrahiert die Signatur durch Anwendung von c' , bzw. er berechnet $c'(s'(c(x))) = s'(x)$.
4. Jeder \mathcal{V} kann sich von der Gültigkeit der Signatur überzeugen, indem er prüft, ob $r(s(s'(x)))$.

3.3.2 Blinde (interaktive) Signaturverfahren aus Kryptosystemen

Da sich die Frage, ob sich jedes digitale interaktive Signaturverfahren blenden lässt (vorerst?) nicht mit Ja beantworten lässt, wird in den folgenden Abschnitten untersucht inwiefern die Funktionen c (bzw. $\varphi_{z,par}$) und c' (bzw. $\psi_{z,par}$) vom Signaturverfahren abhängen dürfen, bzw. welchen Einschränkungen ein Signaturverfahren unterliegt, wenn c und c' nicht frei wählbar sind, sondern implizit aus dem Signaturverfahren hervorgehen müssen. Da die Anwendung von c im weitesten Sinne eine Verschlüsselung ist, macht es Sinn, sich vorerst auf Signaturverfahren zu beschränken, denen ein Verschlüsselungsverfahren, bzw. ein entsprechendes Kryptosystem zugrunde liegt. Der Freiheitsgrad von c wird im Verlauf des Kapitels sukzessive erweitert und die Ergebnisse der Erweiterung jeweils in einem Satz festgehalten.

Während der folgenden Betrachtungen spielt die Vertauschbarkeitseigenschaft von Funktionen eine wichtige Rolle. Das folgende Lemma wird in fast jedem Abschnitt verwendet.

Lemma 3.3.1 (Im Text mit † referenziert). *Gegeben seien Funktionen f, g mit $\mathcal{D}(f) = \mathcal{D}(h) = \mathcal{P}(f) = \mathcal{P}(h) = M$, M beliebig. Dann gilt:*

$$h(f(x)) = f(h(x)) \Leftrightarrow h^{-1}(f^{-1}(x)) = f^{-1}(g^{-1}(x)), x \in M$$

Proof. \Rightarrow : Es gilt $x = f^{-1}fx = f^{-1}h^{-1}hfx \stackrel{\text{nach Annahme}}{=} f^{-1}h^{-1}fhx$. Es gilt auch: $x = h^{-1}hx = h^{-1}f^{-1}fhx$. Zusammenfassend gilt demnach: $f^{-1}h^{-1}fhx = x = h^{-1}f^{-1}fhx$. Durch Substitution von fhx durch y erhält man:

$$f^{-1}h^{-1}y = h^{-1}f^{-1}y, y \in M$$

\Leftarrow : Da $f = (f^{-1})^{-1}$ und $g = (g^{-1})^{-1}$ kann analog zu \Rightarrow argumentiert werden. \square

3.3.3 Blindes interaktives Signaturverfahren aus einem Kryptosystem (I)

In diesem Abschnitt wird aus einem einzelnen asym. Kryptosystem $\mathcal{C} = (M, M, K, E_k, D_k)$ ein blindes Signaturverfahren konstruiert. Die herausgearbeiteten Anforderungen werden am Ende in einem Satz festgehalten.

\mathcal{R} möchte sich eine Nachricht m von \mathcal{S} blind signieren lassen. Um m blenden zu können, muss \mathcal{R} zunächst einen Schlüssel $(sk_r, pk_r) = k_r \in K$ wählen und hat danach die Möglichkeit $\hat{m} = E_{pk_r}(m)$ oder $\hat{m} = D_{sk_r}(m)$ zu berechnen.

Fall $\hat{m} = E_{pk_r}(m)$: \mathcal{R} sendet \hat{m} an \mathcal{S} ; dieser signiert mit der Operation $\hat{s} = D_{sk_s}(\hat{m})$ und sendet \hat{s} zurück an \mathcal{R} . \mathcal{R} extrahiert nun die digitale Signatur $\sigma(m) = D_{sk_r}(\hat{s})$.

Fall $\hat{m} = D_{sk_r}(m)$: Der einzige Unterschied zum vorangegangenen Fall besteht in der Extraktion der Signatur. Diese wird nun über $\sigma(m) = E_{pk_r}(\hat{s})$ berechnet.

Um ein Nachricht-Signatur-Paar von einem Verifizierer \mathcal{V} prüfen lassen zu können, muß \mathcal{R} zunächst m und $\sigma(m)$ an \mathcal{V} senden. Dieser berechnet $\hat{m} = E_{pk_s}(\sigma(m))$ und akzeptiert, falls $m = \hat{m}$.

Damit das blinde Signaturverfahren funktioniert, muß \mathcal{C} einige Eigenschaften erfüllen. Die Verifikationsgleichung der digitalen Signatur ist $m \stackrel{?}{=} E_{pk_s}(D_{sk_s}(m))$. Die Funktionen E_{pk} und D_{sk} müssen demnach vertauschen, bzw. es muss gelten:

$$D_{sk}(E_{pk}(x)) = E_{pk}(D_{sk}(x)) = x, \quad \text{f.a. } x \in M, (pk, sk) \in K$$

Die Verifikationsgleichung des blinden Signaturverfahrens ist $m \stackrel{?}{=} E_{pk_s}(D_{sk_r}(D_{sk_s}(E_{pk_r}(m))))$, falls $\hat{m} = E_{pk_r}(m)$ und $m \stackrel{?}{=} E_{pk_s}(E_{pk_r}(D_{sk_s}(D_{sk_r}(m))))$, falls $\hat{m} = D_{sk_r}(m)$. Wie man leicht sieht, ergeben beide Gleichungen genau dann **true**, wenn zusätzlich $D_{sk_r}(D_{sk_s}(x)) = D_{sk_s}(D_{sk_r}(x))$, bzw. $E_{pk_r}(E_{pk_s}(x)) = E_{pk_s}(E_{pk_r}(x))$ erfüllt ist. Diese Aussagen sind nach † äquivalent. Für das blinde Signaturverfahren muss \mathcal{C} die Eigenschaft

$$E_{pk_1}(E_{pk_2}(x)) = E_{pk_2}(E_{pk_1}(x)), \quad \text{f.a. } x \in M, (pk_1, sk_1), (pk_2, sk_2) \in K$$

haben.

Die folgenden Abbildungen geben das Verfahren schematisch wieder:

(I) Signaturprotokoll σ_I:	
$\mathcal{R}(m, pk_s)$ <hr/> wählt $(pk_r, sk_r) \in K$ berechnet $\hat{m} = E_{pk_r}(m)$	$\mathcal{S}(sk_s)$ <hr/> <div style="text-align: center;"> $\xrightarrow{\hat{m}}$ berechnet $\hat{s} = D_{sk_s}(\hat{m})$ $\xleftarrow{\hat{s}}$ </div> <hr/> completed
<hr/> berechnet $\sigma(m) = D_{sk_r}(\hat{s})$ <hr/> completed	<hr/> completed
(I) Verifikationsprotokoll V_I:	
$\mathcal{R}(m, \sigma(m))$ <hr/> <div style="text-align: center;"> $\xrightarrow{m, \sigma(m)}$ </div> <hr/> completed	$\mathcal{V}(pk_s)$ <hr/> berechnet $\hat{m} = E_{pk_s}(\sigma(m))$ $b := (m \stackrel{?}{=} \hat{m})$ <hr/> okay, falls $b = 1$

Unglücklicherweise ist das Signaturverfahren in dieser Form existentiell fälschbar unter einem Angriff mit bekannten Signaturen. Angenommen einem Angreifer steht ein

Nachricht-Signatur-Paar (m, s) zu Verfügung, so kann er ein beliebiges $(pk_a, sk_a) \in K$ wählen und das Paar $(D_{sk_a}(m), D_{sk_a}(s))$ berechnen. Da die Entschlüsselungsfunktionen nach Annahme vertauschen, gilt

$$(D_{sk}(m), D_{sk}(s)) = (D_{sk}(m), D_{sk}(D_{sk_s}(m))) \quad (1)$$

$$= (D_{sk}(m), D_{sk_s}(D_{sk}(m))) \quad (2)$$

$$= (m', D_{sk_s}(m')) \quad (3)$$

$$= (m', s') \quad (4)$$

Dieses Problem kann jedoch durch Hinzunahme einer kryptographischen Hashfunktion h gelöst werden. \mathcal{R} lässt von \mathcal{S} nicht mehr die Nachricht m , sondern deren Hash $h(m)$ signieren. \mathcal{V} verifiziert, indem er die Gleichheit $h(m) = E_{pk_s}(\sigma(m))$ prüft. Ein Angreifer kann jetzt Signaturen zu zufälligen $h(m)$ fälschen, wegen der Einwegeigenschaft von h aber nicht auf das zugehörige m schließen. Das blinde Signaturverfahren ist jetzt sicher unter einem adaptiven Angriff mit gewählten Nachrichten.

Die Ergebnisse werden nun in einem Satz festgehalten.

Satz 3.3.1. *Aus einem Kryptosystem lässt sich ein blindes Signaturverfahren konstruieren, wenn folgende Bedingungen erfüllt sind:*

- $D_{sk}(E_{pk}(x)) = E_{pk}(D_{sk}(x)) = x$, f.a. $x \in M, (pk, sk) \in K$
- $E_{pk_1}(E_{pk_2}(x)) = E_{pk_2}(E_{pk_1}(x))$, f.a. $x \in M, (pk_1, sk_1), (pk_2, sk_2) \in K$
- *Es existiert eine kryptographische Hashfunktion h*

3.3.4 Blindes interaktives Signaturverfahren aus zwei Kryptosystemen (II)

In Abschnitt 3.3.3 wurde ein blindes Signaturverfahren aus *einem* Kryptosystem konstruiert. Um \mathcal{R} einen größeren Spielraum bei der Blendung von Nachrichten zu geben, wird ein zweites Kryptosystem dazugenommen.

Benötigt werden zwei Kryptosysteme:

- $\mathcal{C}_1 = (M, M, K^{\mathcal{C}_1}, E_k^{\mathcal{C}_1}, D_k^{\mathcal{C}_1})$, asymmetrisch, mit dem das Signaturverfahren modelliert wird,
- $\mathcal{C}_2 = (M, M, K^{\mathcal{C}_2}, E_k^{\mathcal{C}_2}, D_k^{\mathcal{C}_2})$, (a)symmetrisch, das zur Blendung der Nachrichten verwendet wird.

Das Verfahren funktioniert ähnlich dem aus Abschnitt 3.3.3. \mathcal{R} wählt ein $k \in K^{\mathcal{C}_2}$ und blendet die Nachricht m . Er erhält $\hat{m} = E_k^{\mathcal{C}_2}(m)$. Für den Fall $\hat{m} = D_k^{\mathcal{C}_2}(m)$ reicht es am Ende des Abschnitts auf die Verifikationsgleichung einzugehen. \mathcal{R} sendet \hat{m} an \mathcal{S} , welcher \hat{m} mittels $\hat{s} = D_{sk_s}^{\mathcal{C}_1}(m)$ signiert und \hat{s} an \mathcal{R} sendet. \mathcal{R} extrahiert die Signatur, indem er seine Blendung entfernt: $\sigma(m) = D_k^{\mathcal{C}_2}(\hat{s})$.

Am Verifikationsprotokoll zwischen \mathcal{R} und \mathcal{V} ändert sich nichts. \mathcal{R} sendet $m, \sigma(m)$ an \mathcal{V} , welcher $\hat{m} = E_{pk_s}^{\mathcal{C}_1}(\sigma(m))$ berechnet und prüft, ob $m = \hat{m}$ erfüllt ist.

Die Verifikationsgleichung der digitalen Signatur ist $m \stackrel{?}{=} E_{pk_s}^{C_1}(D_{sk_s}^{C_1}(m))$. Die Gleichung ist nur erfüllt, wenn $E_k^{C_1}$ und $D_k^{C_1}$ f.a. $k \in K^{C_1}$ vertauschen, d.h. wenn

$$E_{pk}^{C_1}(D_{sk}^{C_1}(x)) = D_{sk}^{C_1}(E_{pk}^{C_1}(x)) \quad \text{f.a. } x \in M, (pk, sk) \in K^{C_1}$$

gilt.

Die Verifikationsgleichung der zugehörigen blinden Signatur ist $m \stackrel{?}{=} E_{pk_s}^{C_1}(D_k^{C_2}(D_{sk_s}^{C_1}(E_k^{C_2}(m))))$, falls $\hat{m} = E_{pk_s}^{C_1}$, bzw. $m \stackrel{?}{=} E_{pk_s}^{C_1}(E_k^{C_2}(D_{sk_s}^{C_1}(D_k^{C_2}(m))))$, falls $\hat{m} = D_{sk_s}^{C_1}$.

Man sieht leicht, daß beide Gleichungen nur erfüllt sind, wenn die Funktionen $E_{pk_s}^{C_1}$ und $E_k^{C_2}$, bzw. $D_{sk_s}^{C_1}$ und $D_k^{C_2}$ vertauschen. Die Aussage ist nach \dagger äquivalent. Es reicht demnach die Annahme

$$E_{pk}^{C_1}(E_k^{C_2}(x)) = E_k^{C_2}(E_{pk}^{C_1}(x)) \quad \text{f.a. } x \in M, (pk, sk) \in K^{C_1}, k \in K^{C_2}$$

zu treffen.

Die folgenden Abbildungen geben das Verfahren schematisch wieder:

(II) Signaturprotokoll σ_I:	
$\mathcal{R}(m, pk_s)$	$\mathcal{S}(sk_s)$
<hr/> wählt $k_r \in K^{C_2}$ berechnet $\hat{m} = E_{k_r}^{C_2}(m)$	
$\xrightarrow{\hat{m}}$	berechnet $\hat{s} = D_{sk_s}^{C_1}(m)$
<hr/> $\xleftarrow{\hat{s}}$	
berechnet $\sigma(m) = D_{k_r}^{C_2}(\hat{s})$	completed
(II) Verifikationsprotokoll V_I:	
$\mathcal{R}(m, \sigma(m))$	$\mathcal{V}(pk_s)$
<hr/> $\xrightarrow{m, \sigma(m)}$	
berechnet $\hat{m} = E_{pk_s}^{C_1}(\sigma(m))$	
$b := (m \stackrel{?}{=} \hat{m})$	
<hr/>	
completed	okay, falls $b = 1$

Wie in Verfahren (I) kann ein Angreifer \mathcal{A} auch hier Signaturen existentiell fälschen. Zu einem gegebenen Paar (m, s) wählt \mathcal{A} ein $k \in K^{C_2}$ und berechnet $(D_k^{C_2}(m), D_k^{C_2}(s))$. Da $D_{k'}^{C_1}$ und $D_k^{C_2}$ vertauschen, hat \mathcal{A} eine gültiges Nachricht-Signatur-Paar generiert.

$$(D_k^{C_2}(m), D_k^{C_2}(s)) = (D_k^{C_2}(m), D_k^{C_2}(D_{sk_s}^{C_1}(m))) \quad (5)$$

$$= (D_k^{C_2}(m), D_{sk_s}^{C_1}(D_k^{C_2}(m))) \quad (6)$$

$$= (m', D_{sk_s}^{C_1}(m')) \quad (7)$$

$$= (m', s') \quad (8)$$

Das Problem lässt sich auch hier durch den Einsatz einer Hashfunktion lösen. Die Ergebnisse werden nun in einem Satz festgehalten.

Satz 3.3.2. *Aus zwei Kryptosystemen, einem asymmetrischen und einem beliebigen, lässt sich ein blindes Signaturverfahren konstruieren, wenn folgende Eigenschaften erfüllt sind:*

- $E_{pk}^{C_1}(D_{sk}^{C_1}(x)) = D_{sk}^{C_1}(E_{pk}^{C_1}(x))$ f.a. $x \in M, (pk, sk) \in K^{C_1}$
- $E_{pk}^{C_1}(E_k^{C_2}(x)) = E_k^{C_2}(E_{pk}^{C_1}(x))$ f.a. $x \in M, (pk, sk) \in K^{C_1}, k \in K^{C_2}$
- *Es existiert eine kryptographische Hashfunktion h*

3.3.5 Blindes interaktives Signaturverfahren: Spezialfall aus II (III)

Im vorangegangenen Abschnitt wurde für C_2 ein beliebiger Schlüsselraum K^{C_2} zugelassen. Betrachtet man den Spezialfall, daß $K^{C_2} = M$, können die Funktionen E_2^C und D_2^C als innere Verknüpfungen auf M betrachtet werden. Die Anforderung

$$E_{pk}^{C_1}(E_k^{C_2}(x)) = E_k^{C_2}(E_{pk}^{C_1}(x)) \quad \text{f.a. } x \in M, (pk, sk) \in K^{C_1}, k \in K^{C_2}$$

lässt sich dadurch etwas abschwächen.

Sei $\mathcal{C} = (M, M, K, E_k, D_k)$ ein asymmetrisches Kryptosystem. Ferner seien f und f' zwei innere Verknüpfungen auf M , wobei für f' gilt: $f'(f(x_1, x_2), x_1) = x_2$. Intuitiv ist $f(x_1, x_2)$ eine Verschlüsselung von x_2 mit dem Schlüssel x_1 und f' die Entschlüsselung. Um Verwechslungen zu vermeiden, wird die Verschlüsselung mit f im Folgenden mit E_{pk_s} bezeichnet, die Verschlüsselung mit E_{pk_s} mit verschlüsseln bezeichnet.

Das Protokoll verläuft folgendermaßen: \mathcal{R} wählt einen Schlüssel b aus M , im Folgenden Blendfaktor genannt und verschlüsselt diesen mit dem öffentlichen Schlüssel von \mathcal{S} . Er erhält $\hat{b} = E_{pk_s}(b)$. Die Nachricht m wird mit \hat{b} geblendet, d.h. es wird ein $\hat{m} = f(\hat{b}, m)$ berechnet, das zum Signierer \mathcal{S} gesendet wird. \mathcal{S} signiert \hat{m} , d.h. berechnet ein $\hat{s} = D_{sk_s}(\hat{m})$ und sendet die Signatur zurück an \mathcal{R} . \mathcal{R} extrahiert das gesuchte $\sigma(m) = f'(\hat{s}, b)$.

Die Verifikation verläuft wie gehabt: \mathcal{R} sendet $m, \sigma(m)$ an \mathcal{V} , welcher $\hat{m} = E_{pk_s}(\sigma(m))$ berechnet und prüft, ob $m = \hat{m}$ erfüllt ist.

Die Verifikationsgleichung der digitalen Signatur ist $m \stackrel{?}{=} E_{pk_s}(D_{sk_s}(m))$. Die Gleichung ist nur erfüllt, wenn E_k und D_k f.a. $k \in K$ vertauschen, bzw. wenn

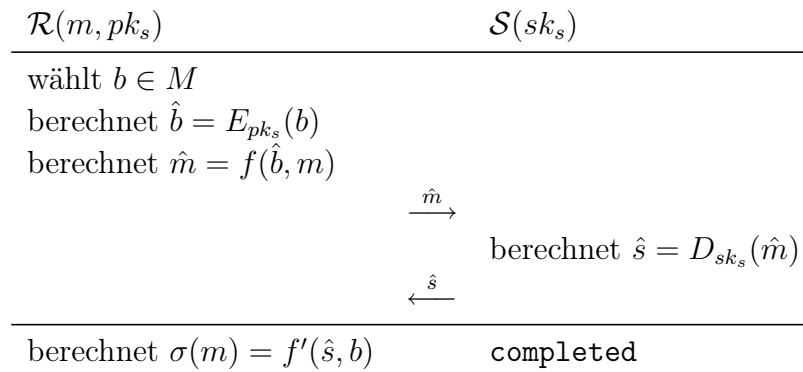
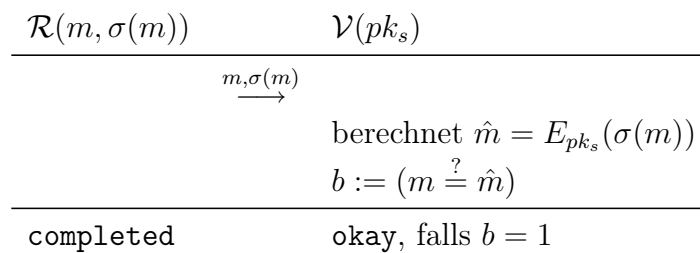
$$E_{pk}(D_{sk}(x)) = D_{sk}(E_{pk}(x)) \quad \text{f.a. } x \in M, (pk, sk) \in K$$

gilt.

Die Verifikationsgleichung der blinden Signatur ist $m \stackrel{?}{=} E_{pk_s}(f'(D_{pk_s}(f(E_{pk_s}(b), m)), b))$. Die Gleichung ist nur erfüllt, wenn für D_{k_s} und f gilt:

$$D_k(f(x_1, x_2)) = f(D_k(x_1), D_k(x_2)) \quad \text{f.a. } x_1, x_2 \in M, k \in K$$

Die folgenden Abbildungen geben das Verfahren schematisch wieder:

(III) Signaturprotokoll σ_I :**(III) Verifikationsprotokoll V_I :**

Das Schema bietet einem Angreifer keine Möglichkeit Signaturen zu fälschen. Die Sicherheit des Verfahrens hängt im konkreten Fall allein von der Güte der Funktionen E_{pk} und D_{sk} ab.

Die Vorteile dieses Schemas liegen darin, daß keine Hashfunktion mehr benötigt wird und mit dem Schlüssel b gerechnet werden kann.

Die Ergebnisse werden nun in einem Satz festgehalten.

Satz 3.3.3. *Aus einem asym. Kryptosystem $\mathcal{C} = (M, M, K, E_k, D_k)$ und zwei inneren Verknüpfungen f, f' auf M lässt sich ein blindes Signaturverfahren konstruieren, wenn \mathcal{C} , f und f' folgende Eigenschaften haben:*

- $E_{pk}(D_{sk}(x)) = D_{sk}(E_{pk}(x))$ f.a. $x \in M, (pk, sk) \in K$
- $D_k(f(x_1, x_2)) = f(D_k(x_1), D_k(x_2))$ f.a. $x_1, x_2 \in M, k \in K$
- $f'(f(x_1, x_2), x_1) = x_2$ f.a. $x_1, x_2 \in M$

Beispiel 3.1 (Das blinde RSA-Signaturverfahren). *Eine bekannte Ausprägung dieses Spezialfalles ist die blinde RSA-Signatur. \mathcal{C} ist hierbei ein RSA-Kryptosystem, f die Multiplikation, f' ein Algorithmus, in dem erst die multiplikative Inverse von b ermittelt und anschließend mit b^{-1} multipliziert wird.*

4 Schluss

4.1 Ergebnisse

Ziel der Arbeit war es ursprünglich den größten gemeinsamen Teiler aller blinden Signaturverfahren zu finden. Während der Recherche wurden einige Verfahren begutachtet. Es stellte sich sehr bald heraus, daß es einen ggT aufgrund der grundsätzlich verschiedenen Ansätze nicht geben kann. Beispielhaft wurde dazu ein weak blindes Signaturverfahren angeführt, dem ein gänzlich anderer Ansatz als z.B. dem blinden RSA-Signaturverfahren zugrunde liegt.

Eine genauere Betrachtung des weak Signaturverfahrens ergab, daß das Verfahren resistent gegenüber Angriffen durch Quantencomputer ist. Der Nachteil dabei ist, das Vertrauen gegenüber einem Teilnehmer gefordert wird. Unterliegen alle Teilnehmer einer Autorität, so wird das Verfahren für diese transparent.

Während der Recherche fiel weiterhin auf, daß die bisherigen blinden Signaturverfahren geblendete digitale Signaturverfahren sind. Dies warf zwei Fragen auf:

1. Ist jedes blinde Signaturverfahren ein geblendetes digitales Signaturverfahren?
2. Lässt sich jedes digitale Signaturverfahren blenden?

Frage 1 lässt sich abschließend mit Ja beantworten, d.h. jedes blinde Signaturverfahren lässt sich in zwei Komponenten spalten. Eine dient der Signatur, eine der Blendung. Eine Konsequenz ist, daß ein digitales Signaturverfahren für die Konstruktion eines blinden Signaturverfahrens notwendig ist.

Frage 2 ließ sich nicht abschließend beantworten. Es konnte jedoch ein Schema entwickelt werden, mit dem aus digitalen interaktiven Signaturverfahren weak blinde interaktive Signaturverfahren konstruieren lassen. Im weiteren Verlauf der Arbeit wurden blinde Signaturschemata entworfen, die auf dem Ansatz D. Chaums basieren. Die Ergebnisse dieser Entwürfe sind:

1. *Ein* asymmetrisches Verschlüsselungsverfahren kann *allein* ausreichend sein ein blindes Signaturverfahren zu konstruieren und
2. eine innere Verknüpfung auf dem Nachrichtenraum muß zwei Bedingungen erfüllen, damit sie als Blendfunktion verwendet werden kann.

4.2 Ausblick

Von gehobenem Interesse könnte die endgültige Beantwortung der Frage nach der Blendbarkeit digitaler Signaturverfahren sein. Ließe sich die Frage mit Ja beantworten, wäre, ausgehend von den Ergebnissen von [12] sichergestellt, daß es auch im Post-Quantum-Zeitalter blinde Signaturverfahren gibt. Ließe sich die Frage allerdings mit Nein beantworten, so könnten, ausgehend von den Ergebnissen aus Abschnitt 3.2.2, weak Signaturverfahren für die Forschung interessant werden. Mögliche Fragestellungen wären:

Lässt sich das benötigte Vertrauen quantifizieren, und falls ja, wieviel Vertrauen wird benötigt? Vertrauen welcher Art wird benötigt? Kann man erzwingen, daß ein Teilnehmer dem ihm gegenüber erbrachten Vertrauen gerecht wird?

Literatur

- [1] Michael Arnold. theorie.informatik.uni-ulm.de/lehre/ss5/krypto-seminar/zero-knowledge.pdf.
- [2] Albrecht Beutelspacher, Jörg Schwenk, and Klaus-Dieter Wolfenstetter. *Moderne Verfahren der Kryptographie*. Vieweg Verlagsgesellschaft, 2006.
- [3] Johannes Buchmann. *Einführung in die Kryptographie*. Springer, Berlin, 2003.
- [4] David Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203, 1982.
- [5] Matthew Franklin and Moti Yung. The blinding of weak signatures (extended abstract).
- [6] Christine Fremdt. *Analyse und Konstruktion blinder interaktiver Signaturen*. PhD thesis, Justus-Liebling-Universität Gießen, 2005.
- [7] Chunxiang Gu and Yuefei Zhu. An efficient id-based proxy signature scheme from pairings. Technical report, Network Engineering Department Information Engineering University Zhengzhou, 2006.
- [8] Jung Zhong Dake He. A new type of group blind signature scheme based on bilinear pairings. Technical report, School of Information Science and Technology, Southwest Jiaotong University, 2006.
- [9] Jonathan Katz and Moti Yung. Complete characterization of security notions for probabilistic private-key encryption. In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 245–254, New York, NY, USA, 2000. ACM.
- [10] Jonathan Katz and Moti Yung. Unforgeable encryption and chosen ciphertext secure modes of operation. In *FSE '00: Proceedings of the 7th International Workshop on Fast Software Encryption*, pages 284–299, London, UK, 2001. Springer-Verlag.
- [11] Tal Rabin. Robust sharing of secrets when the dealer is honest or cheating. *J. ACM*, 41(6):1089–1109, 1994.
- [12] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC '90: Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 387–394, New York, NY, USA, 1990. ACM.
- [13] Schutzziele. <http://www.cryptoshop.com/de/knowledgebase/securitytargets/index.php>.
- [14] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc.
- [15] Heiko Stamer. <http://www.gaos.org/stamer/znp-folien.pdf>.