Bachelorarbeit im Fachgebiet Theoretische Informatik

Fachgebiet Theoretische Informatik - Kryptographie und Computeralgebra

Fachbereich Informatik

TU Darmstadt

Minimale Voraussetzungen für blinde Signaturen

Martin Bergner

Betreuer: Prof. Dr. Johannes Buchmann

Verantwortliche Mitarbeiter: Lucie Langer
Axel Schmidt

April 2008

Ehrenwörtliche 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.

Darmstadt, den 15. April 2008

Inhaltsverzeichnis

1 Einführung

1.1 Einleitung

Im Zeitalter der Quantencomputer ist es für die Kryptographie bedeutsam effiziente blinde Signaturen zu entwickeln. Blinde Signaturen sind eine Erweiterung einer digitalen Signatur, bei denen es nicht möglich sein soll, das signierte Dokument nach oder während der Signatur dem Benutzer zuzuordnen, der es sich hat signieren lassen. Da es zur Zeit keine effizienten blinden Signaturen gibt, die nicht auf der Sicherheit zahlentheoretischer Probleme basieren und somit auch nicht im Zeitalter der Quantencomputer sicher sind, ist die Suche nach solch sicheren blinden Signaturen von Interesse. Viele der bestehenden Signaturen basieren etwa auf der Sicherheit des diskreten Logarithmusproblems, auf dem Faktorisierungsproblem oder auf dem RSA-Problem. Die Frage ist, unter welchen Voraussetzungen es auch andere effiziente blinde Signaturen geben kann, oder wie diese auszusehen haben.

1.2 Problembeschreibung

Ziel dieser Arbeit ist es minimale Anforderungen blinder Signaturen zu bestimmen. Da blinde Signaturen in der Quantenkryptographie noch nicht effizient realisierbar sind, könnte man aus den Ergebnissen eventuell eine blinde Signatur erzeugen, die auch in der Quantenkryptographie sicher ist. Die Frage ist daher, ob es sinnvolle Mindestanforderungen gibt, die eine Existenz von blinden Signaturen in der Quantenkryptographie sicherstellt. Dabei werden sowohl bestehende Signaturen, als auch allgemeine Verfahren zur Konstruktion blinder Signaturen untersucht.

1.3 Aufbau und Ergebnisse

Da es sich für die Betrachtung der Mindestanforderungen anbietet, zuerst bestehende blinde Signaturen zu untersuchen, werden am Anfang dieser Arbeit Grundlagen erklärt, die für ein Verständnis aller betrachteten Verfahren notwendig sind. Danach wird untersucht, welche Arten von Mindestanforderungen einige klassische blinde Signaturverfahren besitzen. Dabei werden zwei Aspekte betrachtet: Die Anforderungen des gesamten Schemas und die Anforderungen des blinden Signaturschemas zusätzlich zum digitalen Signaturschema, auf dem es basiert.
Im nächsten Abschnitt wird ein Resultat vorgestellt, das zeigt, wie man aus einem beliebigen digitalen Signaturschema ein blindes Signaturschema erzeugen kann. Zuletzt wird untersucht, ob man Voraussetzungen für ein blindes Signaturschema formulieren kann, die nur wenige kryptographische Primitive benötigen.
Dabei war es nicht möglich existenzielle Voraussetzungen an ein blindes Signaturschema zu erarbeiten, die über die Ergebnisse aus [16] hinausgehen,

und so werden nur Resultate vorgestellt, die eine blinde Signatur sicherstellen. Abschließend wird herausgearbeitet, inwieweit sich die vorgestellten Verfahren in der Quantenkryptographie verwenden lassen. Eine detaillierte Zusammenfassung aller Ergebnisse findet sich in Kapitel 6.

2 Grundlagen

2.1 Geschichte und Anwendungsbereiche blinder Signaturen

Die erste blinde Signatur wurde von David Chaum [8] veröffentlicht. Er entwickelte sie vor allem als Möglichkeit, elektronisches Geld zu realisieren. Dabei sollte eine vom Kunden gewählte Zufallszahl mit Hilfe einer blinden Signatur von der Bank in eine elektronische Münze umgewandelt werden. Die Händler, die dieses elektronische Geld erhielten, mussten die Korrektheit der Signatur überprüfen. Eine Fälschung war dann ausgeschlossen, wenn nur die Bank die korrekte Signatur erzeugen konnte. Auf der Seite der Bank wurde beim Signieren der Zufallszahl das Geld vom Konto des Kunden abgehoben. Es war danach nicht mehr möglich, die ausgegebene Münze der Person, die sie ursprünglich erhalten hat, zuzuordnen.
Es gibt jedoch auch noch andere Anwendungen blinder Signaturen, wie zum Beispiel elektronische Wahlen. Hier wird der ausgefüllte Wahlzettel blind signiert und es ist so auch im Nachhinein nicht mehr möglich, einen Wahlzettel dem jeweiligen Wähler zuzuordnen.
Neben den normalen blinden Signaturen gibt es auch noch verschiedene Varianten. So wurden zum Beispiel partielle blinde Signaturen [1] entworfen, bei denen ein Teil der signierten Nachricht von beiden Parteien jederzeit eingesehen werden kann. Außerdem gibt es faire blinde Signaturen [24], bei denen es im Nachhinein mit der Hilfe einer vertrauenswürdigen Instanz möglich ist, eine Signatur mit einer Interaktion zu verbinden, um beispielsweise den Benutzer herauszufinden.

2.2 Begriffe

Im folgenden Abschnitt werden einige Begriffe erläutert, die benötigt werden, bestimmte blinde Signaturen zu verstehen.

Das **Random Oracle Modell**[1] ist ein Modell, das dazu dient, die Sicherheit kryptographischer Verfahren zu untersuchen. In diesem Modell wird eine kryptographische Hashfunktion als ein zufälliges Orakel betrachtet. Das heißt, dass dieses zu jeder Eingabe eine völlig zufällige, gleichmäßig aus dem Wertebereich gewählte Ausgabe ausgibt. Dabei erzeugt aber dieselbe Eingabe immer wieder die gleiche Ausgabe. Man verwendet dieses Modell zum Beispiel, wenn sehr hohe Anforderungen an die Zufälligkeit der Hashfunktion gestellt werden müssen. In diesem Modell konnte die Sicherheit vieler blinder Signaturen bewiesen werden. Ein Beweis im Random Oracle Modell reicht aber nicht aus, um vollständig sicher sein zu können, dass das Verfahren im konkretem Fall mit einer bestimmten Hashfunktion anstelle des Orakels auch wirklich sicher ist. Es wurde in diesem Zusammenhang

[1] siehe z.B. http://www.wikipedia.org/wiki/Random_oracle_model

bewiesen, dass es kryptographische Probleme gibt, die im Random Oracle Modell sicher sind, aber für jede Realisierung im Standardmodell unsicher werden [7]. Daher ist ein Beweis im Random Oracle Modell zwar besser als gar kein Beweis, obwohl damit die Sicherheit des Problems in der Realität noch nicht vollständig geklärt ist.

Das **Common Reference String Modell**[2] ist ebenfalls ein Modell zur Analyse von kryptographischen Protokollen und Verfahren, in dem alle beteiligten Parteien einen gemeinsamen String erhalten. In diesem Modell kann zum Beispiel ein nicht interaktiver Zero Knowledge Beweis durchgeführt werden. Ein Problem kann darin bestehen, den String sicher an alle beteiligten Parteien zu verteilen. Das Modell findet auch in der Untersuchung von universell einsetzbaren und verknüpfbaren Protokollen, den so genannten *universally composable protocols* (siehe unten) Verwendung. In vielen Fällen garantiert es auch bei parallelen Angriffen Sicherheit.

Das **Standardmodell** ist ein Modell, in dem keine zusätzlichen Voraussetzungen an die Parteien oder die Umgebung gestellt werden, um die Sicherheit kryptographischer Verfahren zu betrachten. Damit spiegelt dieses Modell die tatsächliche Umgebung in der kryptographische Verfahren eingesetzt werden am besten wider. Es ist in jedem Fall erstrebenswert, die Sicherheit eines Verfahrens im Standardmodell zu beweisen.

Die sogenannten **universally composable protocols**[3] sind Protokolle, die in jeder beliebigen Umgebung und jeder beliebigen Kombination mit anderen Protokollen sicher sind. So kann zum Beispiel ein universell einsetzbares Protokoll in einem größeren Verfahren verwendet werden, wodurch die Sicherheit nicht gefährdet ist. Andere Verfahren könnten in dieser Kombination trotz der Sicherheit der einzelnen Komponenten unsicher werden. Bei universell kombinierbaren Protokollen kann dies nicht passieren. Daher sind diese Protokolle gut dazu geeignet, komplexere Kryptosysteme aufzubauen.

Ein **Commitment Schema** ist ein Verfahren, in dem ein Teilnehmer sich auf eine Zahl oder auch auf irgendeine andere Information festlegt. Dabei wird weder die Information selbst verraten, noch kann sie nachträglich geändert werden. Der Teilnehmer hat aber die Möglichkeit, die Information bei Bedarf zu veröffentlichen. Dies kann zum Beispiel dann sinnvoll sein, wenn die Informationen einem anderen Teilnehmer einen Vorteil verschaffen. Mit einem Commitment ist es also möglich, sich auf etwas festzulegen, ohne es später ändern zu können und ohne dass andere Parteien zur Zeit des Commitments die festgelegten Informationen einsehen können.

[2] Siehe z.B. [6]
[3] Siehe z.B. [5]

Eine bekannte Veranschaulichung [4] für den Nutzen eines Commitment Schemas ist der Münzwurf über ein Telefon. Dabei möchten Alice und Bob eine Münze werfen. Bob sagt Alice dazu, ob er Kopf oder Zahl wählt. Alice wirft dann die Münze und sagt Bob das Ergebnis. Offensichtlich muss Bob darauf vertrauen, dass Alice das richtige Ergebnis zurück gibt. Mit einem Commitment wäre es Bob nun möglich, sich auf eine Wahl festzulegen, ohne dass Alice diese kennt. Sie wäre somit nicht mehr in der Lage, das Ergebnis des Münzwurfs zu ihren Gunsten zu verändern. Auch ist es Bob nicht möglich, seine Wahl nachträglich wieder zu ändern.

Eine **Einweg-Trapdoorfunktion** oder auch **Einweg-Trapdoorpermutation**[5] ist eine mathematische Funktion. Sie besitzt die Eigenschaft, dass sie nur mit Hilfe einer zusätzlichen Information effizient zu invertieren ist. Ohne diese Information sollte es nicht möglich sein, zu einem gegebenen Funktionswert ein Urbild zu finden. Bei einer Einweg-Trapdoorpermutation sind dabei der Funktions- und der Wertebereich gleich. Es ist nicht bekannt, ob es echte Einweg-Trapdoorfunktionen gibt. Es gibt nur gute Kandidaten, die tatsächlich schwer zu invertieren sind. Einweg-Trapdoorfunktion, die nicht auf zahlentheoretischen Problemen beruhen sind vor allem in der Quantenkryptographie gefragt, da die bestehenden Kandidaten, die beispielsweise auf dem diskreten Logarithmusproblem oder RSA beruhen, dort unsicher werden.

Ein **Zero Knowledge-Beweis**[6] ist ein Protokoll, mit dem eine Partei eine andere davon überzeugen kann, dass sie ein Geheimnis kennt, ohne dass sie dieses verrät. Dabei ist wichtig, dass das Protokoll nur dann akzeptiert wird, wenn die andere Partei von der Korrektheit überzeugt wird, und dass es nur dann, wenn auch mit kleiner Fehlerwarscheinlichkeit, abgelehnt wird, wenn der Beweisende das Geheimnis nicht kennt. Des Weiteren sollten während des Protokolls keine Informationen über das Geheimnis veröffentlicht werden. Formal müssen die folgenden drei Punkte erfüllt werden:

1. **Vollständigkeit** Wenn die Aussage korrekt ist, wird ein ehrlicher Benutzer das Protokoll eines ehrlichen Beweisführers akzeptieren.

2. **Zuverlässigkeit** Wenn die Aussage falsch ist, wird ein betrügender Beweisführer nur mit kleiner Wahrscheinlichkeit die Wahrheit beweisen können. Im Weiteren wird diese Wahrscheinlichkeit als soundness-Wahrscheinlichkeit bezeichnet.

3. **Zero-Knowledge** Wenn die Aussage korrekt ist, werden während des Protokolls keine Informationen außer diesem Beweis veröffentlicht.

[4] Siehe z.B. http://www.wikipedia.org/wiki/commitment_scheme
[5] Im Ursprung aus [9], aber auch in http://www.wikipedia.org/wiki/Trapdoor_function
[6] Siehe z.B. http://www.wikipedia.org/wiki/Zero-knowledge_proof

Da die Sicherheitsbegriffe auf Simulationen basieren, kann man für den Fall, dass eine Simulation fast immer wie ein korrekter Beweis aussieht, eine knowledge-Wahrscheinlichkeit definieren. Diese gibt an, mit welcher Wahrscheinlichkeit eine Simulation von einer korrekten Interaktion unterschieden werden kann[7].
Zero-Knowledge-Beweise sind beispielsweise dazu geeignet, sicherzustellen, dass alle Parteien ein Protokoll korrekt befolgen und niemand zu betrügen versucht.

2.3 Sicherheit blinder Signaturen

Damit eine blinde Signatur sicher ist, gibt es zunächst zwei offensichtliche Bedingungen. Es sollte dem Signierer nicht möglich sein, schon während dem Signaturprozess die zu signierende Nachricht zu erfahren. Außerdem sollte er bei einer Veröffentlichung der Nachricht mit der Signatur nicht in der Lage sein, die konkrete Interaktion zu bestimmen, bei der er diese Nachricht signiert hat. Zwar hat er die Nachricht signiert, jedoch ist ihm das zum Zeitpunkt der Signatur nicht bekannt.
Daneben gibt es noch verschiedene Arten von Attacken. Bei der **sequenziellen Attacke** darf der Angreifer lediglich eine Signatur nach der anderen anfragen und bei der **gleichzeitigen Attacke** darf der Angreifer auch eine neue Interaktion beginnen, bevor vorherige Interaktionen abgeschlossen sind. Zusätzlich kann noch zwischen gleichzeitigen und **parallelen Attacken** unterschieden werden. Während ein Angreifer bei parallelen Attacken streng genommen alle Interaktionen zu einem bestimmten Zeitpunkt beginnen muss, kann er bei gleichzeitigen Attacken auch neue Interaktionen starten, nachdem schon vorher gestartete Interaktionen abgelaufen sind.
Des Weiteren kann man Angriffe auch nach den vorhandenen Angriffspunkten kategorisieren. Hat der Angreifer etwa die Möglichkeit, die Nachrichten, die er sich signieren lässt, während der Attacke anzupassen, so nennt man dies einen **adaptive-chosen-message Angriff**. Dabei kann er insbesondere die Nachricht auch nach abgeschlossenen Interaktionen ändern, um so die neu gewonnenen Informationen einzusetzen.
Da sich lange Zeit niemand mit der Sicherheit blinder Signaturen befasste, betrachtete man alleine die Tatsache, dass der Algorithmus nicht gebrochen wurde, als Indiz für dessen Sicherheit. Heutzutage ist man jedoch bestrebt, die Sicherheit kryptographischer Protokolle zu beweisen. Pointcheval und Stern gaben dazu eine formale Definition für die Sicherheit blinder Signaturen an. In ihrem Artikel [21] nennen sie die folgenden Sicherheitsklassen:

1. $(l, l+1)$**-Fälschung** Ein Angreifer kann aus l Interaktionen mit dem Signierer $l+1$ gültige blinde Signaturen erzeugen.

[7] Nähere Informationen zu Zero-Knowledge Beweisen und deren Aufbau finden sich ausführlich in [12]

2. **„One-More"-Fälschung** Für eine ganze Zahl l (polynomiell im Sicherheitsparameter k) kann ein Angreifer nach höchstens l Interaktionen $l+1$ gültige Signaturen erzeugen.

3. **Starke „One-More"-Fälschung** Ein Angreifer kann eine $(l, l+1)$-Fälschung nach l polylogarithmisch beschränkten Interaktionen erzeugen, das heißt, dass für eine Konstante α und den Sicherheitsparameter k gelten muss: $l \leq (\log k)^{\alpha}$.

Diese Sicherheitsklassen gelten offensichtlich insbesondere für E-Cash Systeme, wo der Angreifer aus l Münzen nicht noch weitere erzeugen kann. Die Blindheit der Signatur ist an dieser Stelle wichtig, sollte aber gegebenenfalls wieder aufgehoben werden können, zum Beispiel wenn ein Angreifer eine Straftat begangen hat. Hier können faire blinde Signaturen zur Anwendung kommen. Dagegen ist bei elektronischen Wahlen die Blindheit der Signatur wichtig, um das Wahlgeheimnis zu gewährleisten. In jüngster Zeit wurden diese Sicherheitsbegriffe jedoch noch erweitert und verstärkt [13], um auch mit den aktuellen Beweismethoden sinnvolle Sicherheitsbegriffe zur Verfügung zu haben.

3 Anforderungen bestehender Signaturen

Um die Minimalanforderungen blinder Signaturen besser charakterisieren zu können, werden in diesem Abschnitt einige bestehende Signaturen betrachtet. Dabei wird insbesondere ein Augenmerk auf die Voraussetzungen des gesamten Schemas und auf die Anforderungen des blinden Signaturschemas gelegt, die zu denen des zugrunde liegenden digitalen Signaturschema hinzukommen. Da in dieser Arbeit nicht alle existierenden blinden Signaturschemata vorgestellt werden können, wird eine Auswahl der ersten veröffentlichten Schemata behandelt. Es existieren aber noch andere Schemata, und so wurde erst kürzlich ein neues Schema entwickelt [14], welches auf elliptischen Kurven basiert.

3.1 RSA-basierte Signaturen

Im folgenden Unterabschnitt wird die klassische Chaum Signatur vorgestellt, die erste veröffentlichte digitale Signatur überhaupt. Sie sollte ursprünglich in einer E-Cash Umgebung eingesetzt werden und ist in ihrer Eleganz nicht zu übertreffen.

3.1.1 Die Chaum Signatur

Die erste von Chaum vorgestellte Signatur [8] basierte auf der RSA-Signatur und funktioniert wie folgt:
Es gelte $n = pq$ Produkt von großen Primzahlen, $h(\cdot)$ eine geeignete Einwegfunktion wie z. B. eine Hashfunktion, sowie $de = 1 \mod \varphi(n)$.

1. Der Benutzer wählt x und r zufällig und schickt $B = r^e h(x) \mod n$ an den Signierer. Selbstverständlich muss r invertierbar sein.
2. Der Signierer, gibt $r \cdot h(x)^d \mod n$ zurück.
3. Der Benutzer berechnet $C = h(x)^d \mod n$ aus B.
4. Die Signatur ist nun das Paar $(x, h(x)^d \mod n)$.

Da der Signierer hier nur einen einzigen Schritt machen muss, ist diese Signatur genau so sicher gegen parallele wie gegen sequenzielle Attacken. Die Signatur selbst galt lange Zeit als relativ sicher, obwohl bis 2001 kein formaler Beweis ihrer Sicherheit geführt wurde. Erst Bellare et al. [2] haben mit einigen neuen Annahmen die Sicherheit der Chaum Signatur im Random Oracle Modell für den Fall nachgewiesen, dass der Exponent e prim ist. Dazu formulierten sie neue Annahmen, die mit der Sicherheit von RSA zusammenhängen, die so genannten ,,one-more-RSA-inversion" Probleme. Dabei bekommt der Angreifer Zugang zu einem Orakel, das zu $y \in \mathbb{Z}_N^\star$ den Punkt $x = \mathrm{RSA}_{N,e}^{-1}(y) = y^d$ liefert, ohne d direkt zurückzugeben. Dazu hat

der Angreifer im einen Fall für die Umkehrung von n Punkte nur $n-1$ Orakelaufrufe zur Verfügung (Known-Target-Inversion Problem). Im zweiten Fall kann er aus n Punkten $m+1$ auswählen und muss diese mit m Aufrufen des Orakels invertieren (Chosen-Target-Inversion Problem). Diese Problemstellungen sind sinnvoll, da der Angreifer bei der blinden Signatur eigentlich ein RSA-Inversionsorakel besitzt, nämlich den Signierer. Dieser wird zu jeder Nachricht $M = x^e$ die RSA-Umkehrung $M^d = x$ berechnen. Da der Signierer keine Kontrolle über diese Nachricht hat, muss er sich darauf verlassen, dass der Benutzer dies nicht ausnutzt, um beliebige Punkte x^d damit zu invertieren.
Mit der Annahme, dass die neu formulierten Probleme schwer zu lösen sind, konnte schließlich die Sicherheit der Chaum-Signaturen im Random Oracle Modell bewiesen werden.

Anforderungen

Die Anforderungen dieser blinden Signatur können kurz erklärt werden. Das zugrunde liegende RSA-Problem erfordert einen Restklassenring modulo einer zusammengesetzten Zahl. Des Weiteren wird eine Hashfunktion benötigt. Betrachtet man die Anforderungen der blinden Signatur zusätzlich zur RSA-Signatur, so wird schnell deutlich, dass diese nicht ansteigen. Der Restklassenring wird bei der normalen Signatur genauso benötigt wie die Hashfunktion. Offensichtlich hat die blinde Signatur keine höheren Anforderungen als das zugrunde liegende Signaturverfahren.

3.2 Signaturen basierend auf der Sicherheit des diskreten Logarithmus

Im folgenden Abschnitt werden einige blinde Signaturen vorgestellt, die auf der Sicherheit des diskreten Logarithmus aufgebaut sind. Dabei ist für die kryptographische Sicherheit eine Voraussetzung nötig, die mit der Schwierigkeit einen diskreten Logarithmus zu berechnen vergleichbar ist. Dennoch sind die Signaturschemata interessant, da sie recht einfach gehalten sind.

3.2.1 Das Schnorr Signaturschema

Die blinde Schnorr Signatur [19] setzt Folgendes voraus: Es sind p und q zwei große Primzahlen mit $q|(p-1)$ sowie $g \in \mathbb{Z}_p^\star$ mit $\mathrm{ord}(g) = q$. Zusätzlich benötigt man einen privaten Schlüssel $x \in \mathbb{Z}_q^\star$ und einen öffentlichen Schlüssel $y = g^{-x} \mod p$. Weiter sei h eine Hashfunktion.

1. Der Signierer wählt $k \in \mathbb{Z}_q^\star$ und schickt $r = g^k \mod p$ an den Benutzer.

2. Der Benutzer wählt nun $\alpha, \beta \in \mathbb{Z}_q$, berechnet $r' = rg^{-\alpha}y^{-\beta} \mod p$, $e' = h(m, r') \mod q$ und schickt $e = e' + \beta \mod q$ an den Signierer.

3. Dieser sendet s mit $g^s y^e = r \mod p$ zurück.

4. Der Benutzer berechnet nun $s' = s - \alpha \mod q$. Die gültige Signatur ist nun das Paar (e', s').

Um diese Signatur zu verifizieren, überprüft man

$$e' = h(m, g^{s'} y^{e'} \mod p).$$

Anforderungen

Betrachten wir auch hier die Anforderungen des Verfahrens. Es ist dann sicher, wenn das diskrete Logarithmus Problem in einer q-elementigen Untergruppe der Einheitengruppe $\mathbb{Z}_p^\star$ schwer zu lösen ist. Die Anforderungen der blinden Signatur sind auch hier die gleichen, wie die der digitalen Signatur, da man den Blendungsschritt 2 ohne weiter Voraussetzungen durchführen kann.

3.2.2 Das Okamoto-Schnorr Signaturschema

Es gelten die folgenden Vorbedingungen: Es seien p und q zwei Primzahlen mit $p|(q-1)$. Weiter sind g und h Elemente von $\mathbb{Z}_p^\star$ mit Ordnung q. Der geheime Schlüssel des Signierers ist $(r, s) \in (\mathbb{Z}_q)^2$, der öffentliche Schlüssel ist $y = g^{-r} h^{-s} \mod p$.

1. Der Signierer wählt $t, u \in \mathbb{Z}_q$, berechnet $a = g^t h^u \mod p$ und schickt a an den Benutzer.

2. Der Benutzer wählt $\beta, \gamma, \delta \in \mathbb{Z}_q$, berechnet $\alpha = a g^\beta h^\gamma y^\delta \mod p$ sowie $\epsilon = h(m, \alpha)$ und schickt $e = \epsilon - \delta$ an den Signierer.

3. Dieser berechnet $R = t + er \mod q$ und $S = u + es \mod q$ und schickt das Paar (R,S) an den Benutzer. Dieses Paar erfüllt die Gleichung $a = g^R h^S y^e \mod p$.

4. Der Benutzer berechnet die Signatur (ρ, σ), mit $\rho = R + \beta \mod q$ und $\sigma = S + \gamma \mod q$.

Es gilt dann: $\alpha = g^\rho h^\sigma y^\epsilon \mod p$.

Eine ,,one-more"-Fälschung des Okamoto-Schnorr Signaturschemas ist auch unter einem parallelen Angriff äquivalent zu dem diskreten Logarithmus Problem in einer Untergruppe [20].

Anforderungen

Die Anforderungen dieses Signaturschemas sind die gleichen wie die des Schnorr-Schemas. Auch hier sieht man, dass die Blendungsfunktion nichts verwendet, was nicht auch schon im normalen Signaturschema funktionieren würde. Das blinde Signaturschema hat also insgesamt die gleichen Anforderungen wie das Signaturschema, auf dem es basiert.

3.2.3 Das Nyberg-Rueppel Signaturschema

Das Nyberg-Rueppel Signaturschema basiert auf dem gleichnamigen Schema für digitale Signaturen. Dieses funktioniert wie folgt:
Der Signierer wählt zufällig $k \in \mathbb{Z}_q$ und berechnet r und s mit

$$\begin{aligned} r &= mg^k \mod p \\ s &= xr + k \mod q \end{aligned}$$

Das Paar (r, s) ist die Signatur der Nachricht $m \in \mathbb{Z}_q$. Da man aus der Signatur die Nachricht m berechnen kann, muss man diese selbst nicht mit verbreiten. Die Korrektheit der Signatur lässt sich feststellen, indem man die Gleichheit von

$$m = g^{-s} y^r r \mod p$$

überprüft. Daraus wurde das folgende blinde Signaturschema entwickelt [3]:

1. Der Signierer wählt $\tilde{k} \in \mathbb{Z}_q$, berechnet $\tilde{r} = g^{\tilde{k}} \mod p$ und sendet $\tilde{r}$ an den Signierer.

2. (a) Der Benutzer wählt zufällig $\alpha \in \mathbb{Z}$ und $\beta \in \mathbb{Z}_q^\star$, berechnet $r = mg^\alpha \tilde{r}^\beta \mod p$ und $\tilde{m} = r\beta^{-1} \mod q$.
   (b) Er überprüft, ob $\tilde{m} \in \mathbb{Z}_q^\star$. Wenn dies nicht der Fall ist, geht er wieder zu Schritt a), ansonsten sendet er $\tilde{m}$ an den Signierer.

3. Der Signierer berechnet $\tilde{s} = \tilde{m}x + \tilde{k} \mod q$ und schickt $\tilde{s}$ an den Benutzer.

4. Der Benutzer berechnet $s = \tilde{s}\beta + \alpha \mod q$. Die Signatur ist nun das Paar (r,s).

Die Korrektheit der Signatur überprüft man, indem man

$$g^{-s} y^r r = mg^{-\tilde{s}\beta - \alpha + xr + \tilde{k}\beta + \alpha} = mg^{-\tilde{m}x\beta - \tilde{k}\beta + xr + \tilde{k}\beta} = m(\mod p)$$

berechnet.

Anforderungen

Insgesamt basiert auch dieses Schema auf dem Problem, einen diskreten Logarithmus zu berechnen. Auch hier sind die Anforderungen des blinden Signaturschemas nicht höher als die des digitalen Signaturschemas. Die verwendeten Operationen und Schritte, die für die Blendung der vorausgesetzten Signatur verwendet werden, sind auch hier wegen der Voraussetzungen an die Signatur zulässig. Das blinde Signaturverfahren hat also keine zusätzlichen Voraussetzungen.

3.2.4 Das blinde Signaturschema von Okamoto

Die Voraussetzungen für das Signaturschema von Okamoto [18] sind zwei bilineare Gruppen $(\mathbb{G}_1, \mathbb{G}_2)$ mit den folgenden Eigenschaften:

1. $\mathbb{G}_1, \mathbb{G}_2$ sind zyklische Gruppen der Ordnung p, wobei p eine Primzahl ist. Der Fall $\mathbb{G}_1 = \mathbb{G}_2$ ist zulässig.
2. g_1, g_2 sind Erzeuger von $\mathbb{G}_1$ bzw. $\mathbb{G}_2$.
3. Es sei φ ein Isomorphismus von $\mathbb{G}_2$ nach $\mathbb{G}_1$ mit $\varphi(g_2) = g_1$
4. e sei eine nicht degenerierte, bilineare Funktion $e : \mathbb{G}_2 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$, wobei $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T|$.
   Nicht degeneriert bedeutet hierbei, dass $e(g_1, g_2)$ ein Erzeuger von $\mathbb{G}_T$ ist und bilinear, das heißt, dass für alle $u \in \mathbb{G}_1, v \in \mathbb{G}_2$ und $a, b \in \mathbb{Z}$ die Gleichheit $e(u^a, v^b) = e(u, v)^{ab}$ gilt.
5. e, φ und die Gruppenaktion in $\mathbb{G}_1, \mathbb{G}_2$ und $\mathbb{G}_T$ sollten effizient zu berechnen sein.

Für die Schlüsselerzeugung benötigt man zufällige Erzeuger $g_2, u_2, v_2 \in \mathbb{G}_2$ und setzt $g_1 = \phi(g_2), u_1 = \phi(u_2)$ und $v_1 = \phi(v_2)$. Man wählt danach zufällig $x \in \mathbb{Z}_p^\star$ und berechnet $w_2 = g_2^x \in \mathbb{G}_2$. Der öffentliche Schlüssel ist $(g_1, g_2, w_2, u_2, v_2)$, der geheime Schlüssel ist x. Okamoto setzt weiterhin voraus, dass die zu signierende Nachricht m in $\mathbb{Z}_p^\star$ liegt, gibt aber selbst an, dass man dieses Problem durch eine geeignete Hashfunktion umgehen kann. Das Schema ist deswegen von Bedeutung, weil es auch im Standardmodell sicher ist und das Random Oracle Modell nicht benötigt wird. Das Verfahren funktioniert wie folgt:

1. Der Benutzer überprüft, ob der öffentliche Schlüssel korrekt ist.
2. Der Benutzer wählt $s, t \in \mathbb{Z}_p^\star$, berechnet $X = g_1^{mt} u_1^t v_1^{st}$ und sendet X an den Signierer. Außerdem beweist der Benutzer zusätzlich, dass er $(mt \mod p, t, st \mod p)$ kennt, also X korrekt erzeugt hat:

   (a) Er wählt $a_1, a_2, a_3 \in \mathbb{Z}_p^\star$ und sendet $W = g_1^{a_1} u_1^{a_2} b_1^{a_3}$ an den Signierer.
   (b) Der Signierer wählt $\eta \in \mathbb{Z}_p^\star$ und sendet dies an den Benutzer.
   (c) Der Benutzer berechnet $b_1 = a_1 + \eta m t \mod p, b_2 = a_2 + \eta t \mod p, b_1 = a_1 + \eta s t \mod p$ und sendet (b_1, b_2, b_3) an den Signierer.
   (d) Der Signierer überprüft, ob $g_1^{b_1} u_1^{b_2} v_1^{b_3} = W X^\eta$.

3. Der Signierer wählt zufällig $r \in \mathbb{Z}_p^\star$. Sollte gelten $x + r = 0 \mod p$, so probiert er es erneut mit einem anderen, wieder zufälligen r. Außerdem wählt er $l \in \mathbb{Z}_p^\star$, berechnet $Y = (X v_1^l)^{1/(x+r)}$ und schickt (Y, r, l) an den Benutzer.

4. Der Benutzer wählt zufällig $f, \lambda \in \mathbb{Z}_p^\star$ und berechnet $\tau = (ft)^{-1}$, $\sigma = Y^t$, $\alpha = w_2^f g_2^{fr}$, $\beta = s + l/t \mod p$. Er berechnet $\mathbf{Test}(\alpha) = (U, V)$ mit $U = w_1^{1/f} g_1^\lambda$, $V = w_2^{f\lambda + r} g_2^{fr\lambda}$.

Die blinde Signatur ist dann $(\sigma, \alpha, \beta, \mathbf{Test}(\alpha))$. Sie kann überprüft werden, indem man berechnet:

$$e(\sigma, \alpha) = e(g_1, g_2^m u_2 v_2^\beta), \; e(U, \alpha) = e(w_1, w_2) \cdot e(g_1, V).$$

Okamoto beweist, dass dieses Schema perfekte Blindheit garantiert und unfälschbar ist, solange die 2SDH Annahme gilt[8].

Anforderungen

Die Anforderungen an das Schema macht Okamoto selbst besonders gut deutlich. Die Voraussetzungen sind die entsprechenden Gruppen und Isomorphismen. Offensichtlich muss weiterhin eine Version der starken Diffie-Hellman Annahme gelten, damit das Verfahren sicher ist.
Betrachteten wir die Anforderungen, die die blinde Signatur zusätzlich zur darunter liegenden digitalen Signatur hat, so stellen wir fest, dass zusätzlich zur Signaturerzeugung ein weiterer Schritt eingefügt wurde, der der Blendung dienen soll. In diesem Schritt führt der Benutzer den Beweis, dass er bestimmte Parameter kennt und sich somit korrekt verhält. Dieser Schritt stellt keine neuen Anforderungen an die blinde Signatur. Dies ist insofern nicht verwunderlich, da man auf dem Problem des diskreten Logarithmus viele verschiedene kryptographische Verfahren aufbauen kann. Das blinde Signaturverfahren hat dieselben Anforderungen wie die zugrunde liegende Signatur.

[8] Die 2SDH Behauptung wird in [18] erklärt. Für das weitere Verständnis dieser Arbeit ist es nicht unbedingt erforderlich, genau zu wissen, was diese Annahme besagt. Es reicht, sich vorzustellen, dass angenommen wird, dass gewisse Probleme schwer zu lösen sind und darauf kryptographische Verfahren konstruiert werden können.

3.3 Signaturen basierend auf der Sicherheit des Faktorisierungsproblems

3.3.1 Das Fiat-Shamir Signaturschema

Es gelten folgende Voraussetzungen: Es sei $N = pq$ ein Produkt aus zwei großen Primzahlen, $k \gg \log n$, wobei $n = \log N$. Der geheime Schlüssel des Signierers sei $S_i \in (\mathbb{Z}/N\mathbb{Z})^\star$ für $i \in 1, \ldots, k$, der öffentliche Schlüssel sei $V_i = S_i^2 \mod N$.

1. Der Signierer wählt $t \in (\mathbb{Z}/N\mathbb{Z})^\star$ und schickt $x = t^2 \mod N$ an den Benutzer.

2. Der Benutzer wählt $\beta \in 0, \ldots, 2^k - 1, \gamma_1 \ldots \gamma_k \in \{0,1\}^k$ und berechnet $\alpha = x\beta^2 \prod_{\gamma_i=1}^{k} V_i \mod N$ sowie $\epsilon_1 \ldots \epsilon_k = h(m, \alpha) \in \{0,1\}^k$. Dann schickt er $e = \gamma \oplus \epsilon$ an den Signierer.

3. Dieser schickt $y = t \prod_{i=1}^{k} {S_i}^{e_i} \mod N$ zurück.
   Es gilt $y^2 = x \prod_{i=1}^{k} {V_i}^{e_i} \mod N$.

4. Der Benutzer berechnet $\rho = y\beta \prod_{\gamma_i > e_i} V_i \mod N$.

Dann gilt $\alpha = \rho^2 \prod_{i=1}^{k} {V_i}^{-\epsilon_i} \mod N$.

Wie Pointcheval und Stern bewiesen haben [20], ist eine „one-more“-Fälschung im Fiat-Shamir Signaturschema auch unter parallelen Attacken im Random Oracle Modell äquivalent zum Faktorisierungsproblem.

Anforderungen

Vergleicht man die Schritte des blinden Signaturverfahrens mit dem normalen Signaturverfahren, das auf der Fiat-Shamir Identifikation beruht, so wird wiederum deutlich, dass das blinde Signaturschema keine zusätzlichen Anforderungen zum digitalen Signaturschema aufweist. Es hat somit die gleichen Anforderungen wie das Fiat-Shamir Signaturschema, nämlich einen großen Restklassenring über zusammengesetzten Zahlen und eine Hashfunktion.

3.4 Zusammenfassung

Alle hier betrachteten blinden Signaturen haben keine höheren Anforderungen als die zugrunde liegenden digitalen Signaturen. Das ist jedoch nicht verwunderlich, da man zum Beispiel auf der Sicherheit des diskreten Logarithmusproblems sehr viele kryptographische Verfahren aufbauen kann. Demnach fällt es hier nicht auf, ob tatsächlich ein Zero-Knowledge Beweis gebraucht wurde, der von den Rechenoperationen aber nicht ohne genauere Analyse von einer normalen Berechnung, die für die digitale Signatur

benötigt wird, zu unterscheiden ist. Auch wird dadurch deutlich, dass sich bestimmte Probleme hervorragend dazu eignen, kryptographische Verfahren zu sichern.
Dies bedeutet auch, dass es durchaus möglich ist, ein blindes Signaturschema so zu konstruieren, dass die Anforderungen im Vergleich zum digitalen Signaturschema nicht ansteigen. Es bliebe dann bei der Konstruktion erneut zu beweisen, dass die gewünschten Blindheitseigenschaften erfüllt sind. Es scheint aber nicht so, als könne man leicht ein allgemeines Resultat angeben, das die Sicherheit eines blinden Signaturschemas sicherstellt, da sich die Blendungsfunktionen von Verfahren zu Verfahren unterscheiden. In Kapitel 6 wird noch einmal auf die Vorzüge und die Nachteile dieser Eigenschaften eingegangen.

4 Blendung beliebiger digitaler Signaturen

Im Anschluss an die bisherige Betrachtung, kann man sich nun die Frage stellen, in welchem Fall man aus einer gegebenen digitalen Signatur eine blinde Signatur erstellen kann. Dies ist immer möglich, wenn man die Existenz von Einweg-Trapdoorpermutationen annimmt. Dazu wird im Folgenden das wichtige Resultat von Hazay, Katz, Koo und Lindell betrachtet.

Das blinde Signaturschema von Hazay, Katz, Koo und Lindell

Das Signaturschema von Hazay, Katz, Koo und Lindell [13] ist ein blindes Signaturschema, welches auch unter beliebig vielen sogar gleichzeitigen Aufrufen sicher ist. Es basiert auf einer vereinfachten und angepassten Version von Fischlins Signaturschema[9] und ist im Standardmodell sicher. Um das Schema verstehen zu können, bedarf es noch einiger Definitionen.
Zunächst definieren Lindell et al. ein ambiguous-Commitment-Schema. Dieses unterscheidet sich von einem normalen Commitment-Schema darin, dass das Commitment von einem Schlüssel abhängt, welcher entweder durch eine normale oder durch eine alternativen Schlüsselerzeugung generiert wird. Mit der normalen Schlüsselerzeugung ist das Schema vollständig sicher, gibt also keine Informationen über die festgelegten Informationen preis, während bei der alternativen Schlüsselerzeugung zusätzlich zum Schlüssel noch Informationen erzeugt werden, mit denen eine Extraktion der Informationen erfolgen kann. Solch ein Commitment Schema kann zum Beispiel auf der Basis von verschiedenen zahlentheoretischen Problemen realisiert werden [10].
Weiter definieren Lindell et al. ein ZAP, welcher ein 2 Runden basierter witness-indistinguishable Beweis ist. Dabei ist ein witness-indistinguishable Beweis eine Abschwächung von einem Zero-Knowledge-Beweis[10]. Ein ZAP kann mit der Hilfe von Trapdoor-Permutationen erzeugt werden.
Abschließend benötigen Lindell et al. noch ein cKZ, ein argument of knowledge, welches eine Variante eines concurrent zero-knowledge Beweises ist. Dabei geben sich Lindell et al. mit einem Argument zufrieden und fordern keinen kompletten Beweis. Das bedeutet, dass die soundness-Wahrscheinlichkeit nicht bedingungslos, sondern die Zuverlässigkeit nur bei einem betrügenden Angreifer gelten muss. Dieses Protokoll basiert auf einem Protokoll von Prabhakaran, Rosen und Sahai[22, 23].

Für das Signaturschema, wird zunächst Folgendes benötigt:
Sei $\Pi' = (\text{Gen}', \text{Sign}', \text{Vrf}')$ ein normales Signaturschema, cKZ ein ,,con-

[9] Fischlins Schema [11] erlaubt ebenfalls eine allgemeine Blendung, setzt aber noch ein weiteres Verschlüsselungsschema, welches auch mit Einweg-Trapdoorpermutationen implementierbar ist, und kollisionsfreie Hashfunktionen voraus.

[10] Für eine genauere Beschreibung siehe z.B. http://www.wikipedia.org/wiki/Witness-indistinguishable_proof

current"-Zero-Knowledge Protokoll und sei schließlich (ComGen, Extgen, Com*, Extract) ein ,,ambiguous"-Commitment-Schema. Des Weiteren wird ein ZAP benötigt. Der Algorithmus funktioniert nun wie folgt:

Schlüsselerzeugung:

1. $\mathrm{Gen}'(1^k)$ erzeugt die Schlüssel $(p_k{}', s_k{}')$.
2. $\mathrm{ComGen}(1^k)$ erzeugt p_{k_c}.
3. Der Signierer berechnet ρ als Verifier zu einer Nachricht in einem ZAP.

Daraus ergeben sich der öffentliche Schlüssel (p'_k, p_{k_c}, ρ) und der private Schlüssel (s'_k, z), wobei z der Zufallsparameter ist, der zur Erzeugung von p_{k_c} verwendet wurde.

Signieren einer Nachricht m:

1. $\mathfrak{U}$ berechnet $com = \mathrm{Com}(m)$ und sendet com an $\mathfrak{S}$.
2. $\mathfrak{U}$ und $\mathfrak{S}$ führen cZK aus, um festzustellen, ob p_{k_c} korrekt erzeugt wurde. Schlägt dieser Beweis fehl, so bricht $\mathfrak{U}$ ab. Entdeckt $\mathfrak{S}$ in cZK, dass $\mathfrak{U}$ betrügt, bricht $\mathfrak{S}$ das Signaturprotokoll ab.
3. $\mathfrak{S}$ wählt zufällig eine $nonce = 0, 1^k$, berechnet $\sigma' = \mathrm{sign}'_{s_k{}'}(com||nonce)$ und sendet $\sigma', nonce$ an $\mathfrak{U}$.
4. $\mathfrak{U}$ überprüft die Signatur und berechnet $C^* = \mathrm{Com}^*_{p_{k_c}}(com||nonce||\sigma')$ Er berechnet dann ein ZAP π für ρ für $(m, C^*, p_k{}', p_{k_c}) \in \mathrm{L}_2$ mit

$$\begin{aligned} \mathrm{L}_2 = \{(m, C^*, p_k{}', p_{k_c}) : \quad & \exists (w_1, w_2) : \\ & (com = \mathrm{Com}(m; w_1) \\ & \wedge C^* = \mathrm{Com}^*_{p_{k_c}}(com, nonce, \sigma'; w_2) \\ & \wedge \mathrm{Vrf}'_{p_k}(com||nonce||\sigma) = 1) \\ \bigvee \quad & p_{k_c} = \mathrm{ComGen}(1^k; w_1)\} \end{aligned}$$

.

Die Signatur ist dann (C^*, π).

Verifikation der Signatur:

Um zu überprüfen, ob die Signatur (C^*, π) gültig ist, muss π ein gültiger Beweis für $(m, C^*, p_k{}', p_{k_c}) \in \mathrm{L}_2$ sein.

Das blinde Signaturschema ist dann blind und unfälschbar, wenn Com perfekt bindend und nicht mit einem Computer zu brechen ist, wenn (ComGen, Extgen, Com*, Extract) ein ambiguous Commitment Schema ist, wenn das

ZAP witness-indistinguishable und einen vernachlässigbaren soundness-Fehler hat, wenn das cZK als argument of knowledge einen vernachlässigbaren knowledge-Fehler hat und das normale digitale Signaturschema existentiell unfälschbar unter adaptiven chosen-message Attacken ist. Wichtig ist dabei, dass für die Sicherheit explizit das verwendetet cZK-Protokoll gebraucht wurde. Einen Beweis mit einem beliebigen concurrent Zero-Knowledge Protokoll ist Lindell et al. nicht gelungen.

Anforderungen

Die Anforderungen dieses Schemas werden durch den Sicherheitsbeweis deutlich. Zunächst ist ein beliebiges existenziell unfälschbares digitales Signaturschema nötig. Das alleine ist eine bemerkenswerte Voraussetzung, zeigt sie doch, dass man aus einem beliebigen Signaturschema eine sichere blinde Signatur erzeugen kann. Die weiteren Anforderungen sind jedoch relativ hoch. Ein perfekt bindendes und rechnerisch sicheres [11] Commitment-Schema wird genauso benötigt wie ein weiteres ,,ambiguous"-Commitment-Schema. Zwar behaupten Lindell et al., dass man ein ambiguous Commitment-Schema mit der Hilfe von zahlentheoretischen Problemen realisieren kann, doch wäre für die Sicherheit in der Quantenkryptographie von Bedeutung, ob es auch Möglichkeiten gibt, diese so zu realisieren, dass die Schemata in der Quantenkryptographie sicher sind. Weiterhin wird eine spezielle Variante eines concurrent Zero-Knowledge Protokolls benötigt, die mit der Existenz von Einwegfunktionen auskommt. Eine weitere Voraussetzung ist ein ZAP, welches laut Lindell et al. auch mit Trapdoorpermutationen erzeugt werden kann. Es ist dem Autor nicht bekannt, ob man die Existenz eines ZAP auch auf andere Weise sicherstellen kann.
Daraus folgt, dass man mit Hilfe von Einweg-Trapdoorpermutationen und den beiden Commitment-Schemata aus einer existenziell unfälschbaren digitalen Signatur eine blinde Signatur erzeugen kann. Diese Signatur ist sogar im Standardmodell auch bei parallelen und gleichzeitigen Aufrufen sicher. Leider sind die Anforderungen nicht so gering, dass man daraus sofort ein quantenkryptographisch sicheres blindes Signaturschema herleiten könnte. Das Resultat lässt sich daher hauptsächlich für die Blendung digitaler Signaturen verwenden, die auf zahlentheoretischen Problemen beruhen.

[11] Das bedeutet, dass es mit normalen Computern nicht möglich ist, das Schema in praktikabler Zeit zu brechen.

5 Konstruktion blinder Signaturen aus Einweg-Trapdoorpermutationen

Nachdem in den vorherigen Abschnitten die Anforderungen einer blinden Signatur über einer allgemeinen digitale Signatur sowie die Anforderungen von einigen bestehenden Signaturen betrachtet wurden, stellt sich nun die Frage, wie weit man diese Anforderungen an eine blinde Signatur noch abschwächen kann, wenn man ein geeignetes Signaturschema zugrunde legt. Es sei daran erinnert, dass die im dritten Abschnitt vorgestellten blinden Signaturen keine weiteren Anforderungen außer denen der zugrunde liegenden digitalen Signatur hatten.
Dazu gibt es ein Resultat von Juels, Ostrovsky und Luby [15], welches auf einem Signaturschema von Naor und Yung [17] beruht. Dieses Signaturschema von Naor und Yung setzt ausschließlich Einweg-Trapdoorpermutation voraus. Man kann überraschenderweise zeigen, dass diese auch für die Existenz eines blinden Signaturschemas ausreichend sind.
Um das Schema etwas besser verstehen zu können, ist es nötig, einen Blick auf das Signaturschema von Naor und Yung zu werfen. Das Schema basiert auf der Idee des Markierens von Nachrichten. Eine Signatur ist immer ein gesamter Pfad, bei dem das letzte Element die Nachricht und das Element vorher markiert. Dies kann zum Beispiel mit Hashfunktionen realisiert werden, die auf Einwegpermutationen aufbauen. Weiterhin besteht die Möglichkeit, das ganze Verfahren mit einem Baum zu implementieren, wenn eine höhere Effizient gewünscht ist. Wichtig bei der Verifizierung ist, dass stets der gesamte Pfad überprüft werden muss. Die Konstruktion des Schemas führte zu diversen Problemen, da neue Signaturen vorherige Signaturen offen legten. Das Schema ist auf diese Weise für blinde Signaturen ohne Modifikation nicht geeignet. Dieses Problem haben Juels, Ostrovsky und Luby schließlich mit Hilfe eines Seed behoben.

Das Signaturschema von Juels, Ostrovsky und Luby

Das Signaturschema von Juels, Ostrovsky und Luby war das erste Signaturschema, das auch im Standardmodell sicher war und dessen Sicherheit nicht auf der Sicherheit zahlentheoretischer Probleme beruhte. Für die Sicherheit dieses Schemas im Random Oracle Modell ist die Existenz einer Einweg-Trapdoor-Funktion ausreichend.
Unglücklicherweise ist das Schema nach Aussage der Autoren sehr ineffizient. Doch kann man auf dieser Grundlage möglicherweise ein Schema konstruieren, das auch in der Quantenkryptographie sicher ist.
Die Idee dieses Schemas ist es, anstelle eines Blendungsschrittes ein Resultat der Theorie der UC-Protokolle zu verwenden. Dabei wird eine Berechnung auf eine Art durchgeführt, die es erlaubt, nur bestimmte Informationen an die Parteien weiterzugeben. Hier erfährt der Benutzer die Signatur, während

der Signierer keine Informationen erhält. Auf diese Weise wird die Blindheit der Signatur sichergestellt.
Folglich werden zwei grundlegende Resultate benutzt. Zum einen basiert das Schema wie bereits erwähnt auf dem digitalen Signaturschema von Naor und Yung, welches mit der Existenz von Einwegpermutationen alleine sicher gegen existentielle, adaptive chosen-message Attacken ist. Zum anderen benötigt das Schema das UC completeness Theorem für zwei Parteien [4]. Dieses besagt, dass man jede in polynomieller Zeit berechenbare Funktion mit zwei möglicherweise geheimen Argumenten mit Hilfe eines UC-composable Protokolls so berechnen kann, dass beide Parteien nur das Ergebnis der Berechnung und keine Informationen über die Argumente erhalten. Ferner soll das Verfahren einfach abzuändern sein, so dass nur eine Partei das Ergebnis erhält und die andere Partei keine Informationen bekommt. Auf dieser Grundlage konstruieren Juels et al. das folgende blinde Signaturverfahren.

Der Signierer veröffentlicht ein Commitment $c(s)$ seines pseudozufälligen Schlüssels s zusammen mit dem öffentlichen Schlüssel p_k und der Einwegpermutation, die für das Naor-Yung Signaturschema benötigt wird.
Um eine Signatur zu erstellen, berechnen Signierer und Benutzer mit einer sicheren 2-Teilnehmer UC-Berechnung die Signatur. Der Signierer steuert hierzu seine privaten Informationen, also s_k und s, sowie einige Informationen, die für die Sicherheit des Verfahrens gegen Fälschungen nötig sind, bei, während der Benutzer die Nachricht m, einige Zufallsparameter sowie extractable Commitments einfließen lässt. Die Berechnung liefert dann ein Blatt des Baumes, in das das Ergebnis der Pseudozufallsfunktion g angewendet auf die Nachricht mit dem Seed s eingetragen wird. Bei Erfolg erfährt der Benutzer dann den Signaturpfad $\delta(m)$. Wenn man das Verfahren im Common Reference String Modell benutzt, erhält man sogar ein Verfahren, welches von mehreren Parteien auch gleichzeitig verwendet werden kann.

Bei der Erarbeitung des Verfahrens beobachteten Juels et al. einige Probleme, die dafür sorgten, dass lediglich eine einfache Verbindung des Signaturschemas von Naor und Yung mit dem UC-completeness Theorem nicht ohne weitere Modifikationen sicher war. So behoben sie unter anderem das Problem, dass das Signaturschema von Naor und Yung nicht ohne weiteres zu blenden war. Dazu führten sie ein Commitment eines Seeds ein. Dieses Seed wird in der blinden Signatur verwendet, so dass das Schema vorherige signierte Nachrichten nicht mehr offen legt. Leider ist damit ein weiterer Bestandteil in das Verfahren eingeflossen, der die Verwendung in der Quantenkryptographie erschwert.

Anforderungen

Die Anforderungen dieses blinden Signaturschemas scheinen auf dem Papier sehr gering. Alleine die Existenz von Einweg-Trapdoor-Permutationen stellt sicher, dass dieses Schema sequenziell sicher ist. Betrachtet man das Schema im Common Reference String Modell, so ist es sogar parallel sicher. Bei genauerer Betrachtung wird jedoch deutlich, dass die Anforderungen nicht so gering sind, wie auf den ersten Blick angenommen. Ob echte Einweg-Trapdoor-Permutationen existieren, ist eine offene Frage. Weiter muss man überprüfen, ob es eine Möglichkeit gibt, eine Version eines Commitments zu verwenden, die in der Quantenkryptographie sicher ist, falls man das Signaturschema dort einsetzen möchte. Insgesamt ist es jedoch bemerkenswert zu sehen, dass blinde Signaturen alleine mit Einweg-Trapdoor-Permutationen existieren können. Es wurde außerdem gezeigt, dass blinde Signatur auch mit Hilfe von UC-Berechnungen durchgeführt werden können. Für die weitere Forschung ist es möglich, hier anzusetzen und diese Anforderungen noch weiter abzuschwächen. Vielleicht kann man auf diese Weise ein Schema erhalten, welches auch in der Quantenkryptographie sicher ist.

6 Ergebnisse

Nachdem in dieser Arbeit die Anforderungen blinder Signaturen auf verschiedene Arten untersucht wurden, werden die Ergebnisse noch einmal zusammengefasst und dann genauer analysiert. Dabei wird ein Augenmerk darauf gelegt, inwieweit sich daraus Schlüsse über die Anforderungen blinder Signaturen ergeben und ob man die Ergebnisse zur Konstruktion blinder Signaturen in der Quantenkryptographie verwenden kann.
Im dritten Kapitel wurden einige bestehende blinde Signaturen untersucht. Dabei stellte sich heraus, dass die Blendungsfunktion bei verschiedenen Signaturen zum Teil auch unterschiedlich war. Es wurde jeweils auf das zugrunde liegende digitale Signaturverfahren geachtet, so dass die Blendungsfunktion in das jeweilige Signaturverfahren scheinbar sehr gut integriert war. Die Anforderungen stiegen in diesem Fall nicht an.
Im nächsten Abschnitt wurde ein Resultat vorgestellt, welches die Blendung von digitalen Signaturen ermöglicht. Das Verfahren erlaubt es, jedes genügend sichere digitale Signaturverfahren zu blenden und stellt damit im Wesentlichen sicher, dass die blinde Signaturerzeugung unter gewissen Voraussetzungen nicht schwerer ist, als die Konstruktion einer digitalen Signatur.
Im letzten Abschnitt wurde schließlich eine Konstruktion eines blinden Signaturschemas vorgestellt, die alleine mit der Existenz von Einweg-Trapdoorpermutation sicherstellt, dass blinde Signaturen existieren.
In diesem Abschnitt wird nun erarbeitet, ob diese Resultate für die Erzeugung blinder Signaturen in der Quantenkryptographie verwendbar sind oder welche zusätzlichen Schritte noch gemacht werden müssen, um praktischere Anforderungen zu erhalten.

Die Betrachtung bestehender digitaler Signaturen

Da für die Existenz blinder Signaturen die Existenz eines digitalen Signaturschemas notwendig ist [16], liegt es nahe, geeignete digitale Signaturen zu blenden. Wie im dritten Kapitel gezeigt wurde, müssen blinde Signaturen keine zusätzlichen Anforderungen im Vergleich zum zugrunde liegenden Signaturschema besitzen. Klar ist, dass die Anforderungen des neu konstruierten Signaturschemas mindestens so hoch sein müssen, wie die eines zugrunde liegenden Signaturschemas. Man könnte sonst ein neues digitales Signaturschema aus dem konstruierten blinden Schema erzeugen, welches geringere Anforderungen besitzt. Daher kann es sinnvoll sein, eine blinde Signatur durch die geeignete Blendung einer digitalen Signatur zu konstruieren. Dies hat darüber hinaus noch weitere Vorteile.
Durch eine geeignete Integration des Blendungsschrittes muss sich die Anzahl der Interaktionen nicht erhöhen. Dies ist insbesondere dann von Nutzen, wenn das digitale Signaturverfahren nur zwei Runden benötigt. Würde das

blinde Signaturverfahren die gleiche Anzahl an Interaktionen benötigen, so wäre es auch in einem parallelen Umfeld sicher. Ein Beispiel dafür sehen wir in der Chaum-Signatur. Diese besteht ebenso wie die RSA-Signatur nur aus zwei Runden, setzt man voraus, dass die Nachricht des Benutzers erst zum Signierer gelangen muss.
Des Weiteren kann man durch die geeignete Blendung eines digitalen Signaturverfahrens auch die Implementierung erleichtern. Benötigt der Blendungsschritt keine von der digitalen Signatur unterschiedlichen Rechenoperationen, so kann die Implementierung in Hardware einfacher werden. Es müssen in diesem Fall keine weiteren Bauteile integriert werden, die eventuell neue Rechenoperationen des Blendungsschrittes zur Verfügung stellen. Daher kann es sinnvoll sein, eine geeignete Blendungsfunktion zu finden. Die betrachteten Beispiele besaßen jeweils Blendungsschritte, die keine neuen Rechenoperationen benötigten.
Bei einer integrierten Blendungsfunktion kann außerdem auf die Eigenheiten der digitalen Signatur eingegangen werden. Das bedeutet, dass eine Blendungsfunktion im Allgemeinen nicht in jedem Signaturverfahren verwendet werden kann. Es ist daher möglich, dass das Verfahren einfacher zu implementieren oder zu verstehen sein kann, wenn eine geeignete Blendungsfunktion zugrunde liegt. Dabei kann diese Blendungsfunktion auch Sachverhalte ausnutzten, die in der gewählten kryptographischen Umgebung vorhanden sind. Zusätzlich kann auch die Effizienz erhalten bleiben.
Leider hat der Ansatz, ein bestimmtes digitales Signaturverfahren zu blenden, auch Nachteile. Es ist daher nicht immer sinnvoll, eine geeignete Blendung eines digitalen Signaturverfahrens zu suchen.
Zum einen ist alleine das Problem, eine geeignete Blendungsfunktion zu finden, sehr schwierig. Für die Konstruktion der Blendungsfunktion ist es in jedem Fall nötig, das Signaturverfahren und dessen Eigenheiten genau zu verstehen. Auch ist es aufgrund der verschiedenen Verfahren nicht möglich, einen allgemeingültigen Algorithmus zur Konstruktion einer Blendungsfunktion anzugeben.
Ein weiteres Problem besteht schließlich darin, die Sicherheit des konstruierten Verfahrens zu beweisen. Es ist zwar möglich, dass sich die Analyse des neuen Verfahrens vereinfacht, wenn ein geeigneter Blendungsschritt eingeführt wird. In der Regel wird jedoch der umgekehrte Fall eintreten. Die Analyse des Verfahrens wird oft durch die Blendungsfunktion erschwert. Des Weiteren kann das Verfahren durch den Blendungsschritt nicht mehr parallel sicher sein. Der Sicherheitsbeweis muss in jedem Fall komplett neu erbracht werden.
Betrachtet man die Vor- und Nachteile insgesamt, so fällt auf, dass die Nachteile zwar zahlenmäßig geringer, aber von der Schwierigkeit wesentlich höher ausfallen. Klar ist in jedem Fall, dass die Analyse bestehender digitaler Signaturen sehr aufwendig ist, in Spezialfällen aber auch sehr vielversprechend sein kann. Hat man eine gute Idee wie eine geeignete Blendungsfunktion aus-

sehen kann oder hat man diese sogar schon gefunden, so ist dies einem allgemeinen Blendansatz, wie er später noch einmal genauer untersucht wird, vorzuziehen. Gerade im Hinblick auf die Voraussetzungen kann eine geeignete Blendungsfunktion ein nicht zu unterschätzender Vorteil sein. Es bleibt dennoch zu zeigen, dass das neu konstruierte Verfahren auch wirklich sicher ist. Diesen Schritt kann ein allgemeiner Ansatz deutlich vereinfachen. Ob dieser in jedem Fall sinnvoller ist, wird im folgenden Abschnitt untersucht.

Blendung beliebiger digitaler Signaturen

Wie im vierten Kapitel gezeigt wurde, ist es möglich, ein digitales Signaturschema, welches unter einem adaptive-chosen-message-Angriff sicher ist, allgemein zu blenden. Dies bedeutet, dass unter noch näher zu spezifizierenden Umständen das Problem blinde Signaturen zu finden nicht viel schwerer ist, als das Problem digitale Signaturen zu finden. Diese Aussage ist enorm wichtig, da sie zeigt, dass man jede digitale Signatur blenden kann. Eine genauere Analyse der digitalen Signatur ist in diesem Fall nicht mehr erforderlich. Der Sicherheitsbeweis des Verfahrens wurde bereits erbracht und so ist das Verfahren im Standardmodell sequenziell, im Common-Reference-String Modell sogar parallel und gleichzeitig sicher.
Ein weiterer Vorteil der allgemeinen Blendung ist, dass die Anforderungen des Blendungsschrittes klar definiert sind. Man benötigt das ZAP, cKZ und zwei Commitmentschemata. Das heißt, dass es nicht notwendig ist, sich den Blendungsschritt genauer anzuschauen, um herauszufinden, in welcher Umgebung dieser funktioniert. Des Weiteren ist von Anfang an voraussehbar, welche Komponenten zusätzlich zum digitalen Signaturverfahren benötigt werden.
Einschränkend ist zu bemerken, dass die Anforderungen relativ hoch sind und das Verfahren laut Aussage der Autoren recht ineffizient ist. Das bedeutet, dass sich dieses allgemeine Blendungsverfahren nicht dazu eignet effiziente blinde Signaturen zu konstruieren. Für die Konstruktion werden sehr spezielle Verfahren verwendet, die teilweise auch für die Sicherheit notwendig sind. Das kann dann zu einem Nachteil werden, wenn die Sicherheit der zu blendenden digitalen Signatur nicht auf zahlentheoretischen Problemen beruht. Durch die Anwendung des allgemeinen Blendungsverfahrens kann es zu einer Abhängigkeit von zahlentheoretischen Problemen kommen, welche gerade in der Quantenkryptographie problematisch ist. Da die Existenz einiger im Blendungsverfahren verwendeter Komponenten auf der Grundlage zahlentheoretischer Probleme beruht und es nicht klar ist, ob diese Komponenten auch anders realisiert werden können, können diese neue Anforderungen an das blinde Signaturschema nach sich ziehen.
Es ist also im Allgemeinen nicht sinnvoll, eine generelle Blendungsfunktion zu verwenden, da so nicht auf die Möglichkeiten des blinden Signaturverfahrens eingegangen werden kann. Weiter können die Anforderungen des

gesamten Schemas soweit ansteigen, dass sie in der geplanten Umgebung nicht mehr praktikabel sind.
Das generelle Blendungsverfahren kann dennoch eine wichtige Rolle spielen. Da gezeigt wurde, dass jedes digitale Signaturverfahren blendbar ist, kann nun versucht werden, die Anforderungen einer allgemeinen Blendungsfunktion noch zu verringern. Hierzu kann das bestehende Verfahren von Hazay, Katz, Koo und Lindell analysiert werden und es kann auf dieser Grundlage versucht werden Abschwächungen der Anforderungen zu erhalten. Insbesondere kann es viel versprechend sein, zu untersuchen, ob es neben zahlentheoretischen Problemen weitere Voraussetzungen gibt, unter denen die Existenz des Verfahrens sichergestellt ist. Würde man solche Voraussetzungen finden, ließe sich auf der Grundlage des vorgestellten Verfahrens eine quantenkryptographisch sichere Möglichkeit entwickeln, blinde Signaturen zu konstruieren.

Möglichkeiten der Blendung

Die vorgestellten Verfahren zeigen weiter, dass es mehrere Möglichkeiten gibt, digitale Signaturen zu blenden. Die Analyse der bestehenden Signaturen zeigt, dass es durchaus möglich ist, die Blendung mit einer geeigneten Blendungsfunktion so durchzuführen, dass sich das Blendungsverfahren gut in die digitale Signatur einpaßt. Das bedeutet, dass es sinnvoll sein kann, zu untersuchen, inwieweit ein Blendungsschritt auf den von der zugrunde liegenden digitalen Signatur vorgegebenen Problemen implementiert werden kann. Dies kann die schon bereits erwähnten Vorteile haben.
Ein weiteres Resultat der Arbeit von Juels, Ostrovsky und Luby ist es, dass es möglich ist, Ergebnisse aus der Theorie der UC-Protokolle zu verwenden und so eine Blendung zu versuchen. Dabei ist diese Idee insofern vielversprechend, da sie auch einen allgemeinen Ansatz einer Blendung darstellt. Die zusätzlichen Voraussetzungen der blinden Signatur ergeben sich bei dieser Variante schlussendlich aus denen des zu verwendeten Verfahrens, welches die sichere Berechnung implementiert.
Daneben gibt es jedoch zusätzlich die Möglichkeit, das Verfahren mit einer generellen Blendungsfunktion zu blenden. Natürlich hat dies die genannten Nachteile, man muss sich aber nicht auf die schwierige Suche nach einer Blendungsfunktion machen. Auch wenn dieser Ansatz so nicht im Allgemeinen funktioniert, so ist es doch wichtig, dass er existiert. Da es meist einfacher ist, bestehende Verfahren zu vereinfachen, kann nach der generellen Blendung versucht werden, die Funktion so abzuändern, dass sie besser zum zugrunde liegenden Signaturverfahren passt. Daraus folgt, dass es auf diese Weise möglich sein kann, eine gute Blendungsfunktion zu erhalten, indem man sich das generelle Verfahren zur Blendung als Startpunkt für weitere Verfeinerungen vornimmt.

Minimale Anforderungen blinder Signaturen

Wie bereits am Anfang der Arbeit erwähnt, gibt es mehrere Möglichkeiten die Anforderungen blinder Signaturen zu untersuchen. Im Folgenden werden diese Ergebnisse noch einmal genauer analysiert.

Die erste Betrachtungsweise ermöglicht es, Aussagen über die Anforderungen blinder Signaturen zusätzlich zu den Anforderungen der zugrunde liegenden digitalen Signatur zu treffen. Wie Kahl [16] gezeigt hat, ist die Existenz einer digitalen Signatur eine Voraussetzung für die Existenz einer blinden Signatur. Die Betrachtung bestehender blinder Signaturen führt nun zu der Erkenntnis, dass eine blinde Signatur keine zusätzlichen Voraussetzungen besitzen muss. Dies bedeutet aber auch, dass es bei einer quantenkryptographisch sicheren blinden Signatur nicht nur auf die Voraussetzungen des gesamten Schemas ankommt. Es kann genauso vielversprechend sein, den Versuch zu unternehmen, eine quantenkryptographisch sichere digitale Signatur zu blenden.
Aufgrund der Möglichkeit beliebige digitale Signaturverfahren zu blenden, können auch in diesem Fall die Anforderungen des Schemas betrachtet werden. Legt man die Anforderungen des verwendeten digitalen Signaturschemas zugrunde, so werden die Anforderungen des gesamten Schemas deutlich. Neben den offensichtlichen Anforderungen werden für die Blendung noch das ZAP, cKZ und die Commitment-Schemata benötigt. Diese Anforderungen sind dadurch nicht einfach zu erfüllen, dass sie zum Teil spezielle Ausprägungen kryptographischer Primitive sind. Es ist wichtig zu erwähnen, dass diese für den Sicherheitsbeweis zum Teil unbedingt benötigt werden. Auch ist zum Beispiel für das ,,ambiguous"-Commitment lediglich bekannt, dass es mit der Hilfe zahlentheoretischer Probleme konstruierbar ist. Ob es auch die Möglichkeit gibt, dieses Schema anders zu konstruieren, ist dem Autor nicht bekannt.
Im letzten Abschnitt wurde ein Resultat vorgestellt, dass die Konstruktion blinder Signaturen auf der Basis von Einweg-Trapdoorpermutationen erlaubt. Es ist bemerkenswert, dass man alleine mit der Voraussetzung der Existenz blinde Signaturen konstruieren kann, da sich so die Zahl benötigter kryptographischer Primitive auf ein einziges reduziert. Auf den zweiten Blick fällt jedoch auf, dass sowohl Commitments als auch Zero-Knowledge-Beweise benötigt werden. Da die Existenz von Einweg-Trapdoorpermutation keineswegs geklärt ist, sind diese Anforderungen schwierig zu verwenden[12]. Dass Einweg-Trapdoorpermutationen als Voraussetzung genannt werden, überrascht insofern nicht, als dass man auf der Basis dieser Funktionen relativ viele kryptographische Verfahren konstruieren kann. Das bedeutet, dass die Erkenntnis, dass Einweg-Trapdoorpermutationen ausreichen, zwar wich-

[12]Im nächsten Teil wird darauf im Hinblick auf die Post-Quantenkryptographie noch einmal näher eingegangen.

tig ist, aber dennoch erlaubt sie es nicht ohne weiteres blinde Signaturen zu konstruieren.

Verwendung der Ergebnisse in der Quantenkryptographie

Im folgenden Abschnitt wird nun analysiert, inwieweit die bisherigen Ergebnisse für blinde Signaturen in der Quantenkryptographie verwendbar sind. Des Weiteren wird untersucht, an welchen Stellen noch Arbeit investiert werden muss, um die Ergebnisse in der Quantenkryptographie benutzen zu können und wie diese Veränderungen aussehen müssten.

Klar ist, dass die Betrachtung bestehender blinder Signaturen für die Konstruktion blinder Signaturen in der Quantenkryptographie nur eingeschränkt von Bedeutung ist. Es lassen sich dennoch einige Ergebnisse formulieren.
Wie bereits festgestellt wurde, müssen blinde Signaturen bei geeigneter Blendungsfunktion keine zusätzlichen Anforderungen im Vergleich zum zugrunde liegenden digitalen Signaturschema erhalten. Dieses Resultat lässt sich eventuell auch in der Quantenkryptographie verwenden. Daher kann auch die genauere Analyse eines digitalen Signaturschemas in der Quantenkryptographie vielversprechend sein. Das Problem bleibt unglücklicherweise trotzdem schwierig, da es im Allgemeinen nicht ohne weiteres möglich ist, eine geeignete Blendungsfunktion zu finden.
Betrachtet man die allgemeine Blendung nach dem Schema von Hazay, Katz, Koo und Lindell, so stellt sich auch hier die Frage, inwieweit man dieses Resultat für die Konstruktion blinder Signaturen in der Quantenkryptographie verwenden kann. Die Anforderungen wurden bereits genannt: Das cKZ, ZAP, ein normales Commitment-Schema sowie das ,,ambiguous"-Commitment-Schema. Dabei geben die Autoren an, dass sowohl das ,,ambiguous"-Commitment-Schema auf der Basis zahlentheoretischer Probleme als auch das ZAP auf der Basis von Trapdoorpermutationen implementiert werden kann. Das erste Problem, welches im Blendungsverfahren von Lindell et al. auftritt, ist die Verwendung von Commitments. Da es in der Quantenkryptographie Probleme mit uneingeschränkt sicheren Commitments gibt, gilt es zu untersuchen, welche Art von Commitment hier benutzt werden kann, um die nötige Sicherheit zu garantieren. Auch die weiteren Anforderungen sind in der Post-Quantenkryptographie nicht einfach zu erfüllen, da zum einen die Existenz von Trapdoorpermutationen nicht geklärt ist und zum anderen zahlentheoretische Probleme wie RSA oder das DL-Problem mit Quantencomputern einfach zu brechen sind. Es kann versucht werden, quantenkryptographisch sichere Einweg-Trapdoorpermutationen zu finden, auch wenn dies für die Benutzbarkeit des Schemas vorerst von Nachteil sein kann[13]. Das bedeutet, dass dieses Resultat nicht ohne weiteres in der Quan-

[13]Wie in [10] genannt wird, kann man Trapdoorpermutationen auch im Kontext von Quantencomputern definieren. Hierbei ist es jedoch möglich, dass normale Computer noch

tenkryptographie verwendbar ist. Sollte es jedoch möglich sein, Einweg-Trapdoorpermutationen zu finden, die auch von Quantencomputern nicht gebrochen werden können, so findet sich in diesem Schema eine Möglichkeit, blinde Signaturen auch in der Quantenkryptographie zu realisieren.
Das Schema von Juels, Luby und Ostrovsky ist dann sicher, wenn man die Existenz von Einweg-Trapdoorpermutationen voraussetzt. Die Frage ist nun, ob das bei der Konstruktion blinder Signaturen in der Quantenkryptographie hilfreich ist. Die Frage ist nicht einfach zu beantworten, da die Existenz von Einweg-Trapdoorpermutationen noch nicht geklärt ist. Das hat zwei Auswirkungen. Auf der einen Seite lässt sich dieses Schema zur Zeit nicht anwenden, da noch nicht genau geklärt ist, ob es Einweg-Trapdoorpermutationen gibt, die in der Quantenkryptographie sicher sind. Auf der anderen Seite kann man das Schema sehr gut verwenden, wenn es quantenkryptographisch sichere Einweg-Trapdoorpermutationen gibt, da außer diesen Permutationen keiner weiteren Anforderungen an das Verfahren gestellt werden.
Schlussendlich lässt sich folgern, dass die Existenz von post-quantenkryptographisch sicheren Einweg-Trapdoorpermutationen eine Möglichkeit eröffnet, quantenkryptographisch sichere blinde Signaturen zu konstruieren.

nicht einmal die normale Berechnung oder auch die Umkehrung durchführen können.

7 Fazit

Wie man leicht erkennen kann, müssen die Anforderungen für blinde Signaturen auf den ersten Blick nicht sehr hoch sein. Die Existenz einer Einweg-Trapdoorpermutation ist ausreichend. Doch das ist trügerisch, denn bisher ist es ein offenes Problem, ob diese Einweg-Trapdoorpermutationen überhaupt existieren. Die bekannten Kandidaten beruhen auf zahlentheoretischen Problemen und sind in der Quantenkryptographie nicht sicher. Unter diesem Blickwinkel ist es ebenfalls fraglich, ob man mit Hilfe dieses Resultats quantenkryptographisch sichere blinde Verfahren erzeugen kann. Es ist daher schwierig, alleine über die Minimalanforderungen einen Weg zu suchen, auf dem man solch sichere blinde Signaturen erhält.
Die Betrachtung digitaler Signaturverfahren und der Versuch, diese mit Hilfe einer geeigneten Blendungsfunktion aus einer blinden Signatur zu erzeugen, kann wesentlich viel versprechender sein, da dabei auf die Eigenheiten des Verfahrens eingegangen werden kann. Zwar können die Anforderungen des blinden Signaturverfahrens schnell ansteigen, was aber kein Problem darstellt, für den Fall dass das Signaturverfahren diese Anforderungen ohnehin voraussetzt. Es ist dabei nicht ohne eine genauere Analyse der zugrunde liegenden digitalen Signatur möglich, praktische Minimalanforderungen der blinden Signatur zusätzlich zu den Anforderungen der digitalen Signatur zu nennen.
Trotzdem ist es möglich, ein allgemeines Verfahren zur Blendung eines digitalen Signaturverfahrens anzugeben. Die Anforderungen an diese Blendung, nämlich die Existenz von Einweg-Trapdoorpermutationen und bestimmten Commitment-Schemata, ist nicht ohne weiteres in der Quantenkryptographie geklärt. Bemerkenswert ist jedoch, dass es ein allgemeines Verfahren zur Blendung gibt. Gelingt es nun, die Anforderungen dieses Verfahrens auch in der Quantenkryptographie zu erfüllen oder das Verfahren so abzuändern, dass es auch in der Quantenkryptographie sicher ist, kann man aus einem beliebigen digitalen Signaturverfahren eine blinde Signatur erstellen. Dies bedeutet auch, dass dann die Möglichkeit besteht, auf einem quantenkryptographisch sicheren digitalen Signaturverfahren ein ebenso sicheres blindes Signaturverfahren zu erzeugen. Das Schema von Juels, Ostrovsky und Luby eröffnet weiterhin die Möglichkeit in der Post-Quantenkryptographie sichere blinde Signaturen zu konstruieren, wenn die Existenz von Einweg-Trapdoorpermutationen dort sichergestellt ist.

Literatur

[1] Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In *CRYPTO '00: Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, pages 271–286, London, UK, 2000. Springer-Verlag.

[2] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The power of rsa inversion oracles and the security of chaum's rsa-based blind signature scheme. In *FC '01: Proceedings of the 5th International Conference on Financial Cryptography*, pages 319–338, London, UK, 2002. Springer-Verlag.

[3] Jan L. Camenisch, Jean-Marc Piveteau, and Markus A. Stadler. Blind signatures based on the discrete logarithm problem. *Lecture Notes in Computer Science*, 950:428–432, 1995.

[4] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multiparty secure computation, 2003.

[5] Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 13(1):143–202, 2000.

[6] Ran Canetti and Marc Fischlin. Universally composable commitments. Cryptology ePrint Archive, Report 2001/055, 2001.

[7] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. pages 209–218, 1998.

[8] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *CRYPTO '88: Proceedings on Advances in cryptology*, pages 319–327, New York, NY, USA, 1990. Springer-Verlag New York, Inc.

[9] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[10] Paul Dumais, Dominic Mayers, and Louis Salvail. Perfectly concealing quantum bit commitment from any quantum one-way permutation. *Lecture Notes in Computer Science*, 1807:300+, 2000.

[11] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 60–77. Springer, 2006.

[12] Oded Goldreich. *Foundations of Cryptography*, volume Basic Tools. Cambridge University Press, 2001.

[13] Carmit Hazay, Jonathan Katz, Chiu-Yuen Koo, and Yehuda Lindell. Concurrently-secure blind signatures without random oracles or setup assumptions. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 323–341. Springer, 2007.

[14] Debasish Jena, Sanjay Kumar Jena, and Banshidhar Majhi. A novel blind signature scheme based on nyberg-rueppel signature scheme and applying in off-line digital cash. In *ICIT '07: Proceedings of the 10th International Conference on Information Technology*, pages 19–22, Washington, DC, USA, 2007. IEEE Computer Society.

[15] Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In *CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, pages 150–164, London, UK, 1997. Springer-Verlag.

[16] Benjamin Kahl. Blinde signaturen und post-quantum-kryptographie. Bachelorarbeit, Technische Universität Darmstadt, 2007.

[17] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *STOC '89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 33–43, New York, NY, USA, 1989. ACM Press.

[18] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. Cryptology ePrint Archive, Report 2006/102, 2006.

[19] David Pointcheval and Jacques Stern. Provably secure blind signature schemes. In *ASIACRYPT: Advances in Cryptology – ASIACRYPT: International Conference on the Theory and Application of Cryptology*. LNCS, Springer-Verlag, 1996.

[20] David Pointcheval and Jacques Stern. New blind signatures equivalent to factorization (extended abstract). In *CCS '97: Proceedings of the 4th ACM conference on Computer and communications security*, pages 92–99, New York, NY, USA, 1997. ACM Press.

[21] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 13(3):361–396, 2000.

[22] M. Prabhakaran and A. Sahai. Concurrent zero knowledge proofs with logarithmic round-complexity, 2002. Electronic Colloqium on Computational Complexity, ECCC.

[23] M. Prabhakaran and A. Sahai. New notions of security: Achieving universal composability without trusted setup. In *STOC '04: Proceedings of the 36th annual ACM symposium on Theory of computing*, pages 242–251, New York, NY, USA, 2004. ACM Press.

[24] Markus A. Stadler, Jean-Marc Piveteau, and Jan L. Camenisch. Fair blind signatures. *Lecture Notes in Computer Science*, 921:209+, 1995.