Vorgelegt an der TU Braunschweig
Institut für Theoretische Informatik

Wie man ein Geheimnis verrät: Theorie und Anwendung von Ringsignaturen

Seminar zur Kryptologie
Professor Dr. Dietmar Wätjen

von
Jan Oliver Ringert
Braunschweig
17. Januar 2007

Abstract

Ringsignaturen bieten besondere Möglichkeiten bei der Signatur von Nachrichten. Mit einer Ringsignatur existieren mehrere mögliche Unterzeichner, von denen mindestens einer die Unterschrift erzeugt hat. Es ist nicht möglich, unter den Unterzeichnern denjenigen zu finden, der die Unterschrift tatsächlich erzeugt hat. Die Menge der möglichen Erzeuger der Unterschrift ist Bestandteil der Signatur und es ist gewährleistet, dass der tatsächliche Unterzeichner sich darin befindet.

Inhaltsverzeichnis

1 Einführung

Im Folgenden wird ein Vorschlag von Rivest, Shamir und Tauman vorgestellt, welcher den Begriff "Ringsignatur" definiert. Dieses Verfahren wurde 2001 zum ersten Mal veröffentlicht und in einer überarbeiteten und erweiterten Form 2006 ein weiteres Mal publiziert.
Rivest et al. (RST06) stellen weiterhin ein Verfahren vor, mit dem Ringsignaturen erstellt werden können. Dabei besteht eine Ringsignatur aus einer Menge von möglichen Unterzeichnern, die mit ihren öffentlichen Schlüsseln angegeben werden sowie weiteren Daten, die die Signatur ausmachen. Es kann bewiesen werden, dass mindestens einer der möglichen Unterzeichner die Unterschrift erstellt hat.
Der wahre Unterzeichner benötigt zur Erzeugung der Signatur nur die öffentlichen Transformationen (Schlüssel) der genannten möglichen Unterzeichner. Die möglichen Unterzeichner müssen demnach Schlüssel zu einem Signaturverfahren wie z.B. dem RSA oder Rabin Verfahren besitzen, deren öffentliche Komponenten zugänglich sind.
Das Ringsignaturverfahren hat den Vorteil gegenüber bekannten Möglichkeiten, dass es sehr effizient ist. Es wird bei effizient gewähltem Signaturverfahren pro Mitglied des Ringes nur eine modulare Multiplikation und eine symmetrische Verschlüsselung hinzugefügt. Dem wahren Unterzeichner, welcher der Erzeuger der Ringsignatur ist, wird eine uneingeschränkte Anonymität durch das Verfahren gewährleistet. Es wurde bewiesen, dass das Verfahren im Sinne des Zufälligen Orakel Modells als sicher gilt.

2 Anwendung

Es sind für die Ringsignaturen zwei Anwendungen von besonderer Bedeutung. Die erste und hauptsächliche Verwendung ist das anonyme Herausgeben von Informationen aus einer Gruppe. Die zweite Anwendung ist das Erstellen von Unterschriften, die nur von einem bestimmten Empfänger verifiziert werden können.

2.1 Geheimnisse verraten

Die Motivation der Ringsignatur ist, dass eine Person aus einer Gruppe von Leuten ein Geheimnis nach Außen dringen lassen möchte. Die Einschränkungen sind:

1. Der Verräter muss diese Aufgabe ohne die Hilfe der anderen Mitglieder seiner Gruppe schaffen.
2. Er muss anonym innerhalb der Gruppe bleiben.
3. Es soll nachgewiesen werden können, dass das Geheimnis von mindestens einem Mitglied aus der Gruppe unterzeichnet wurde.

Ein motivierendes Beispiel für die Notwendigkeit eines solchen Verfahrens findet sich unter anderem in "Leaking the secret of snow white" (Rin06). Neben Ringsignaturen existieren verschiedene Verfahren, mit denen Gruppensignaturen erzeugt werden können. Die Gruppensignatur nach (CV91) erfüllt den Punkt 3 vollständig, da der Empfänger (und jeder Weitere) die Signatur auf Echtheit prüfen kann. Der Punkt 1 wird nicht komplett erfüllt, da zwar jedes Mitglied der Gruppe berechtigt ist, im Namen der Gruppe alleine zu unterzeichnen, das Verfahren jedoch erst einmal eingerichtet werden muss. Zur Einrichtung des Verfahrens werden verschiedene Instanzen und die Zustimmungen aller Mitglieder benötigt. Der Punkts 2 wird insofern erfüllt, als dass es normalen Personen nicht möglich ist, herauszufinden, welches Mitglied der Gruppe die Unterschrift erzeugt hat. Es ist jedoch dem Gruppenmanager, der zur Einrichtung des Verfahrens benötigt wird, möglich, die Anonymität eines Gruppenmitglieds aufzuheben. Er kann feststellen und beweisen, wer der Unterzeichner ist.
Die Anwendung eines Anonymisierers (GRS99) im Internet deckt Punkt 1 ab, da der Verräter die Nachricht ohne weitere Hilfe senden kann. Weiterhin wird der Punkt 2 erfüllt, da jegliche Identifikationsinformationen entfernt werden. Es ist jedoch nicht möglich, den wichtigen Punkt 3 zu erfüllen, da jeder Anonymisierer sämtliche Arten von Signaturen entfernen muss.
Die im Obigen gestellten Anforderungen werden von der vorgeschlagenen Ringsignatur wie folgt erfüllt:

1. Der Verräter muss lediglich die öffentlichen Transformationen der anderen möglichen Unterzeichner kennen. Mit dieser Kenntnis kann er die Signatur einer Nachricht ohne weitere Hilfe erzeugen. Das Verfahren ist vorbereitungsfrei.
2. Der Verräter besitzt eine uneingeschränkte Anonymität.
3. Jeder Empfänger kann die Gültigkeit der Signatur effizient überprüfen. Es ist berechnungsmäßig sicher, dass sie von einem Mitglied aus der angegebenen Menge erzeugt wurde.

2.2 Signatur mit festgelegtem Prüfer

Mit einer Ringsignatur ist es möglich Nachrichten zu signieren, so dass nur ein bestimmter Empfänger die Echtheit der Unterschrift prüfen kann. Dabei ist es dem Empfänger nicht möglich, eine dritte Partei von der Echtheit der Signatur zu überzeugen.
Eine Signatur mit festgelegtem Prüfer ist dann sinnvoll, wenn eine Partei ein Dokument so unterzeichnen möchte, dass eine zweite Partei von der Echtheit der Signatur überzeugt ist, diese

aber keiner dritten Partei gegenüber beweisen kann. Ein konkretes Beispiel wäre das Aushandeln von Verträgen. Dabei übermitteln sich zwei Parteien gegenseitige Vorschläge, an die sie sich aber noch nicht binden wollen. Sie möchten ihre Unterschriften also nur für den jeweiligen Partner prüfbar machen.
Mögliche Lösungen für dieses Problem sind interaktive Zero Knowledge Protokolle, die sehr aufwendig sind, aber durch ihre Konstruktion einen Beweis gegenüber einer dritten Partei verhindern. Eine andere Möglichkeit ist das Verwenden eines MAC-Verfahrens mit symmetrischem Schlüssel wie z.B. in (Sti95). Wenn Partei A eine Nachricht mit einem MAC authentifiziert und an Partei B schickt, weiß B, dass die Nachricht nur von A kommen konnte. Später kann B gegenüber Dritten nicht beweisen, dass sie von A kam, da ja auch B den MAC erzeugen kann. Zum Einsatz dieses Verfahrens müssen die beiden Parteien vorher auf einem sicheren und geheimen Weg einen gemeinsamen Schlüssel vereinbaren. Diese Vereinbarung ist unter Umständen sehr aufwendig.
Eine Abhilfe, die keinerlei Vorbereitung benötigt, wird durch Ringsignaturen geschaffen. Partei A sendet ihre Nachrichten einfach authentifiziert mit einer Ringsignatur an B. Als mögliche Unterzeichner werden in dem Ring nur A und B genannt. Die weitere Argumentation ist identisch zu der obigen bzgl. MACs.

3 Grundlegende Definitionen und Begriffe

In diesem Abschnitt werden wichtige Definitionen und Begriffe angegeben, die in dieser Arbeit verwendet werden. Einige der angeführten Definitionen entsprechen nicht vollständig einer allgemeinen Form. Sie sind bezogen auf das dargestellte Thema vereinfacht angegeben. Es werden zuerst in der Arbeit verwendete Begriffe geklärt, die sich auf kryptografische Grundlagen beziehen. Weiterhin werden die von Rivest, Shamir und Tauman gemachten Definitionen bezüglich Ringsignaturen angegeben.

3.1 Allgemeine Definitionen

Berechnungsmäßige Sicherheit/Anonymität

Berechnungsmäßig sicher bedeutet, dass die Sicherheit darauf beruht, dass für ihre Kompromittierung ein sehr hoher Rechenaufwand notwendig ist (in der Regel ein exponentieller Aufwand). Beispielsweise gilt ein Verschlüsselungsverfahren (z.B. DES) als berechnungssicher, da zum Brechen der Chiffre ein exponentieller Zeitaufwand nötig ist.

Uneingeschränkte Sicherheit/Anonymität

Uneingeschränkte Sicherheit gibt an, dass es keine Möglichkeit gibt, das Verfahren zu kompromittieren. Es ist dabei unwichtig, welche Menge an Rechen- bzw. Speicherkapazität der Angreifer zur Verfügung hat. Weiterhin kann der Angreifer beliebige Mengen an Chiffre und Klartexten besitzen. Ein Beispiel für eine Chiffre mit uneingeschränkter Sicherheit ist das One-Time-Pad, bei dem der Schlüssel zufällig gewählt wird und mindestens die Länge des Klartextes besitzt.

Modell des Zufälligen Orakels

In dem Modell des Zufälligen Orakels (MP93) existiert ein Orakel, auf das alle Teilnehmer und Angreifer Zugriff haben. Das Orakel ist eine Abbildung von einer Eingabe auf eine vollständig zufällige Ausgabe. Das Orakel wird angegeben als eine Funktion RO mit der folgenden Signatur.

$$RO : \{0,1\}^* \rightarrow \{0,1\}^k$$

Die Länge der Ein- und Ausgabe können abhängig von einem Sicherheitsparameter sein. Es ist in diesem Modell gewährleistet, dass alle Anfragen an das Orakel mit gleicher Eingabe die gleiche Ausgabe produzieren.
Es ist eine gängige Praxis, Verfahren und Protokolle im Modell des zufälligen Orakels als sicher zu beweisen. In der Realität ist ein zufälliges Orakel jedoch nicht implementierbar und wird häufig durch Hashfunktionen ersetzt (CGH98).
Es ist gezeigt worden, das unter gewissen Bedingungen ein als in diesem Modell sicher bewiesenes Verfahren bei seiner Implementierung sofort unsicher wird (CGH98).

Modell der Idealen Verschlüsselung

Das Modell der Idealen Verschlüsselung kann als Einschränkung des Zufälligen Orakels gesehen werden. Es existiert eine öffentlich zugängliche Funktion E:

$$E : \{0,1\}^m \times \{0,1\}^l \rightarrow \{0,1\}^l$$

Wobei m die Bitlänge des Schlüssels und l die Bitlänge der Ein- und Ausgabe ist.
Die Funktion E ist für einen festen ersten Parameter eine Bijektion. Die Abbildung geschieht wie im Modell des Zufälligen Orakels vollständig zufällig. Es existiert weiterhin

eine Umkehrabbildung

$$E^{-1} : \{0,1\}^m \times \{0,1\}^l \to \{0,1\}^l$$

für E und E^{-1} gilt $\forall k, x : E^{-1}(k, E(k, x)) = x$.
Das Modell der idealen Verschlüsselung ist im Modell des zufälligen Orakels einfach implementierbar.
Dieses Modell wird, wie das des zufälligen Orakels, genutzt, um die Korrektheit von Protokollen und Verfahren zu beweisen.
Es ist gezeigt worden, das unter gewissen Bedingungen ein als in diesem Modell sicher bewiesenes Verfahren bei seiner Implementierung sofort unsicher wird (Bla05).

Hashfunktion

Eine Hashfunktion ist eine Funktion h, die Eingaben x von beliebiger Länge auf Ausgaben $h(x)$ einer festen Länge abbildet. Diese Abbildung ist bei gegebenen x und h leicht berechenbar. Für eine stark kollisionsfreie Hashfunktion ist es berechnungsmäßig praktisch unmöglich zwei x_1, x_2 zu finden mit $h(x_1) = h(x_2)$ (Wät03).
Stark kollisionsfreie Hashfunktionen können durch ein Zufälliges Orakel simuliert werden (MP93). Teilweise ist es auch möglich Verfahren, die im Zufälligen Orakel Modell als sicher bewiesen wurden durch eine geeignete Hashfunktion zu implementieren (CGH98).

Symmetrische Verschlüsselungsfunktion

Eine symmetrische Verschlüsselungsfunktion E_k bildet Eingaben aus einem Klartextraum M in einen Chiffretextraum C ab (Wät03). Wenn $C = M$ gilt, handelt es sich bei E_k um eine bijektive Abbildung. E_k entsteht im Allgemeinen aus der Definition eines Verschlüsselungsalgorithmus und einem Schlüssel k. Es ist möglich, E_k in dem zufälligen Orakel Modell als Orakel darzustellen, das Anfragen nach $E_k(x)$ und $E_k^{-1}(x)$ beantwortet. Es soll weiterhin die Bijektivität gelten, also Existenz von $E_k^{-1}(x)$ mit $E_k^{-1}(E_k(x)) = x$.
Das Modell der Idealen Verschlüsselung stellt solche Funktionen E_k und E_k^{-1} bereit.

3.2 Definitionen des Verfahrens

Unterzeichner

Der Unterzeichner ist die Person, welche die Unterschrift erstellt. Es handelt sich dabei um die Person, die das Geheimnis verraten möchte.

Mögliche Unterzeichner

Als mögliche Unterzeichner werden die Mitglieder der Gruppe angegeben, die der Unterzeichner erstellt. Er selbst ist auch ein möglicher Unterzeichner. Alle möglichen Unterzeichner bilden den Ring. Die Menge der möglichen Unterzeichner wird über deren öffentliche Schlüssel als Teil der Signatur mit angegeben.

Ringsignaturfunktionen

- $ring\text{-}sign(m, P_1, P_2, ..., P_r, s, S_s) =: \sigma$ produziert eine Signatur für die Nachricht m. Es sind P_1 bis P_r die öffentlichen Schlüssel der möglichen Unterzeichner und S_s der private Schlüssel des Unterzeichners sowie s seine Position in der Aufzählung der öffentlichen Schlüssel.
- $ring\text{-}verify(m, \sigma)$ überprüft für eine Nachricht m, ob σ eine gültige Signatur dieser Nachricht ist. Die Signatur σ enthält dabei die öffentlichen Schlüssel der möglichen Unterzeichner.

Vorbereitungsfreies Verfahren

Das vorgestellte Verfahren für Ringsignaturen ist vorbereitungsfrei. Das bedeutet, es ist nicht nötig, Schritte durchzuführen, die andere Mitglieder des Ringes aktiv einbeziehen. Der Unterzeichner benötigt lediglich den Zugriff auf die öffentlichen Schlüssel der Personen, die er in den Kreis der möglichen Unterzeichner aufnehmen möchte.

4 Vorbereitungen für Ringsignaturen

Bevor die Erzeugung von Ringsignaturen vorgestellt wird, sind einige Konstruktionen nötig, um bestehende Signaturverfahren verwenden zu können. Nach der Modifikation der Signaturfunktionen wird die Familie der Kombinationsfunktionen vorgestellt. Diese bildet die Grundlage für das Erzeugen von Ringsignaturen.

4.1 Definitionsbereichserweiterung

Für das Erzeugen einer Ringsignatur wird ein bestehendes Signaturverfahren benötigt. In dem Vorschlag von Rivest et al. müssen alle Mitglieder des Ringes das selbe Signaturverfahren verwenden. Die betrachteten Signaturverfahren sind das RSA-Signaturverfahren und das Rabin-Signaturverfahren es handelt sich dabei um Verfahren, die der Idee von Diffie und Hellman (DH76) entsprechen, öffentliche und private Schlüssel zu verwenden. Beide Verfahren ermöglichen es den individuellen Nutzern, die Parameter für ihre Signatur mit gewissem Spielraum zu wählen.
Bei beiden Verfahren ist die Bitlänge der Ein- und Ausgabe abhängig von der des Modulus. Damit die Signaturen aller möglichen Unterzeichner in dem Ring genutzt werden können, wird ein gemeinsamer Definitionsbereich der Funktionen benötigt.
Im Folgenden wird vorgestellt, wie die verschiedenen öffentlichen Transformationen f_i der möglichen Unterzeichner i über $\mathbb{Z}_{n_i}$ auf einen gemeinsamen Definitionsbereich $\mathbb{Z}_{2^b}$ erweitert werden können. Es ist dabei b ungefähr um 160 größer als die Bitlänge des größten in der Signatur genutzten n_i.
Es findet eine Erweiterung der $f_i : \mathbb{Z}_{n_i} \to \mathbb{Z}_{n_i}$ zu $g_i : \mathbb{Z}_{2^b} \to \mathbb{Z}_{2^b}$ statt. Die Eingabe $m \in \mathbb{Z}_{2^b}$ wird dabei zu q und r zerlegt, so dass gilt $m = qn_i + r$ mit $r = m \ mod \ n_i$.
Die Erweiterung lautet:

$$g_i(x) = \begin{cases} qn_i + f_i(r) & \text{wenn } (q+1)n_i \leq 2^b \\ x & \text{sonst} \end{cases}$$

Die obige Fallunterscheidung ist nötig, da $f_i(r)$ bis zu $n_i - 1$ groß werden kann. Die Summe in der oberen Zeile ergäbe $qn_i + n_i - 1(q+1)n_i - 1$ was größer als 2^b sein kann. Wir betrachten nun, in welchen Fällen die Eingabe nicht modifiziert wird:

> Die Länge von x ist maximal b Bit. Die Länge des n_i sei c Bit. Es folgt eine Länge von q als $(b-c)$ Bit. Der Fall $(q+1)n_i > 2^b$ ist gleichbedeutend damit, dass die Bitlänge von $(q+1)n_i$ $b+1$ beträgt. Es folgt eine Bitlänge von $(b-c)+1$ für $q+1$. Der einzige Fall, in dem dies auftritt ist, dass q in Binärdarstellung nur aus Einsen besteht. Wählt man die Bitlänge b um 160 größer als die des größten n_i, dann bedeutet dies, eine Wahrscheinlichkeit von maximal $\frac{1}{2^{160}}$. Also tritt bei genügend großem b der zweite Fall nur mit vernachlässigbarer Wahrscheinlichkeit auf.

4.2 Kombinationsfunktion

Grundlage der Ringsignaturen ist eine Kombinationsfunktion.

Definition 4.1. *Eine Kombinationsfunktion besitzt die Form:*

$$C : \{0,1\}^l \times \{0,1\}^b \times (\{0,1\}^b)^r \longrightarrow \{0,1\}^b$$

Dabei ist l die Bitlänge des Schlüssels einer symmetrischen Verschlüsselung und b die Bitlänge der Ausgabe der erweiterten Signaturfunktionen.

Damit lässt sich C schreiben als:

$$C_{k,v}(y_1, y_2, ..., y_r) \mapsto z$$

Es ist k ein Schlüssel für eine symmetrische Verschlüsselung, v ein Initialisierungsvektor, y_i verschiedene Werte aus dem selben Definitionsbereich und z die Ausgabe der Kombinationsfunktion. Die Familie der für Ringsignaturen gültigen Kombinationsfunktionen besteht aus allen Funktionen, welche die nachfolgenden Kriterien erfüllen:

Bijektion für alle y_s
Bei festen Parametern v, k und $y_{i\neq s}$ ist die neue Funktion $h(y_s) = z$ eine Bijektion.

Effizientes Bestimmen von y_s
Aus der Gleichung $C_{k,v}(y_1, y_2, ..., y_r) = z$ muss jedes y_s (für alle anderen Werte fest gewählt) effizient bestimmbar sein. Das y_s ist durch die vorhergegangene Forderung eindeutig bestimmt.

Unlösbarkeit von $C_{k,v}(g_1(x_1), g_2(x_2), ..., g_r(x_r)) = z$
Es soll nicht möglich sein, eine Lösung $x_1, ..., x_r$ von

$$C_{k,v}(g_1(x_1), g_2(x_2), ..., g_r(x_r)) = z$$

für feste Parameter k, v und z zu finden. Die Gleichung soll unlösbar sein, wenn zu keinem $g_s(x)$ ein $g_s^{-1}(x)$ bekannt ist. Ist ein $g_s^{-1}(x)$ bekannt, folgt aus der vorhergegangenen Forderung sofort die effiziente Berechenbarkeit einer Lösung $x_1, ..., x_r$.

Um den Begriff der Kombinationsfunktion zu verdeutlichen, werden nun zwei verschiedene Funktionen vorgestellt. Die erste ist sehr intuitiv, erfüllt jedoch nicht allen Kriterien. Die zweite Kombinationsfunktion ist die in (RST06) vorgestellte Funktion.

4.2.1 Unzureichende Kombinationsfunktion

Bei den Kombinationsfunktionen handelt es sich um spezielle Kompressionsfunktionen. Von mehreren Eingaben wird auf eine einzige Ausgabe abgebildet. Ein wichtiger Punkt ist dabei, dass es sich bei $C_{k,v}$ immer noch um eine Bijektion für einzelne Parameter handeln soll.

Definition 4.2. *Eine erste Kombinationsfunktion ist:*

$$C_{k,v} = v \oplus y_1 \oplus ... \oplus y_r$$

Dabei ist $\oplus$ eine bitweise XOR-Verknüpfung der Parameter.
Die obigen Kriterien werden wie folgt erfüllt:

- $C_{k,v}$ ist klar eine Bijektion für jedes einzelne y_s. Wegen der Kommutativität und Assoziativität lassen sich die $y_{i\neq s}$ mit v zu einem Wert $b = v \oplus y_{i\neq s}$ zusammenfassen. Die neue Funktion $y_s \mapsto y_s \oplus b$ ist durch die Eigenschaften von $\oplus$ eine Bijektion.
- Zu der obigen Konstruktion $y_s \mapsto y_s \oplus b$ existiert eine inverse Abbildung $z \mapsto z \oplus b$. Somit lässt sich die Gleichung $C_{k,v} = y_s \oplus b = z$ nach jedem y_s auflösen: $y_s = z \oplus b$.
- Die letzte Forderung wird nicht erfüllt. Es lässt sich bei großem r mit nicht vernachlässigbarer Wahrscheinlichkeit eine Lösung für die Kombinationsfunktion finden (siehe (RST06)).

4.2.2 Konkrete Kombinationsfunktion

Die Kombinationsfunktion, die in (RST06) vorgeschlagen wurde nutzt neben der XOR-Verknüpfung auch noch eine symmetrische Verschlüsselung in jeder Komponente.

Definition 4.3. *Eine für Ringsignaturen mögliche Kombinationsfunktion lautet:*

$$C_{k,v}(y_1, y_2, ..., y_r) = E_k(y_r \oplus E_k(y_{r-1} \oplus E_k(y_{r-2} \oplus E_k(... \oplus E_k(y_1 \oplus v)...)))) = z$$

Wenn man für die y_i jeweils die öffentlichen Signaturfunktionen $g_i(x_i)$ einsetzt erhält man die folgende Darstellung.

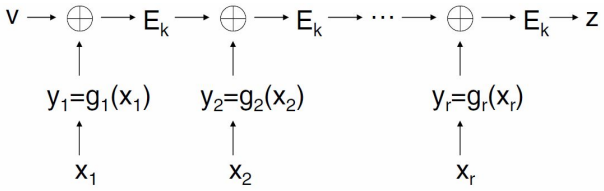

Abbildung 1: Illustration der Kombinationsfunktion aus (RST06).

Im Folgenden soll gezeigt werden, dass diese Funktion die geforderten Eigenschaften besitzt.

Satz 4.1. *$C_{k,v}$ ist eine Bijektion für jedes einzelne y_s.*

Beweis. Für jedes y_s lässt sich die obige Vorschrift umformen. Die Verknüpfung aller y_i mit $i < s$ ist die "rechte Seite" der Kombinationsfunktion und kann geschrieben werden als

$$b_{rs} := E_k(y_{s-1} \oplus E_k(... \oplus E_k(y_1 \oplus v)...)).$$

Die "linke Seite" bis y_s ist die Verknüpfung der y_i mit $i > s$, die ebenfalls zusammengefasst wird als

$$b_{ls}(\xi) := E_k(y_r \oplus E_k(y_{r-1} \oplus E_k(... \oplus E_k(y_{s+1} \oplus \xi))..)).$$

Es ergibt sich aus der Gleichung $C_{k,v}(y_1, y_2, ..., y_r) = z$ die äquivalente Gleichung

$$z = b_{ls}(E_k(y_s \oplus b_{rs})) = \Phi_s(y_s).$$

Dabei ist b_{rs} ein fester Wert. Es gilt noch zu zeigen, dass $\Phi_r(y_s)$ eine Bijektion für y_s ist.
Da $(\circ \oplus b_l)$ und $E_k(\circ)$ jeweils Bijektionen sind, bleibt zu zeigen, dass es sich bei $b_{ls}(\xi)$ auch um eine Bijektion handelt. Siehe Lemma 4.1.
Der Fall $s = r$ ist durch diesen Beweis ebenfalls behandelt. In diesem Fall ist $b_{rs} = v$ und die Induktion im Beweis von Lemma 4.1 läuft bis r.

□

Lemma 4.1. *$b_{ls}(\xi) := E_k(y_r \oplus E_k(y_{r-1} \oplus E_k(... \oplus E_k(y_{s+1} \oplus \xi))..))$ ist eine Bijektion.*

Beweis. Die Indizes der y_i lassen sich wie folgt neu nummerieren:

$$b_l(\xi) := E_k(y_n \oplus E_k(y_{n-1} \oplus E_k(... \oplus E_k(y_1 \oplus \xi))..)).$$

Mit natürlicher Induktion über n lässt sich die Behauptung einfach zeigen, da es sich bei $(y_i \oplus \circ)$ und E_k um Bijektionen handelt.

□

Satz 4.2. *Jedes y_s ist aus der Gleichung $C_{k,v}(y_1, y_2, ..., y_r) = z$ effizient bestimmbar.*

Beweis. Wie im vorhergegangenen Beweis ergibt sich die "rechte Seite" der Gleichung zu b_{rs}. Die "linke Seite" der Gleichung bis y_s lautet:

$$z = E_k(y_r \oplus E_k(y_{r-1} \oplus E_k(... \oplus E_k(y_{s+1} \oplus \xi)))).$$

Diese lässt sich mit E_k^{-1} der Umkehrfunktion zu E_k und der Kommutativität sowie Assoziativität von $\oplus$ äquivalent umformen zu

$$\xi = y_{s+1} \oplus E_k^{-1}(... \oplus E_k^{-1}(y_r \oplus E_k^{-1}(z))...).$$

Für ξ gilt außerdem

$$\xi = E_k(y_r \oplus b_{rs}).$$

Damit lässt sich $C_{k,v}(y_1, y_2, ..., y_r) = z$ auflösen zu

$$y_s = E_k^{-1}(\xi) \oplus b_{rs}.$$

□

Satz 4.3. *Es ist nur mit vernachlässigbarer Wahrscheinlichkeit möglich, eine Lösung $x_1, ..., x_r$ für*

$$C_{k,v}(g_1(x_1), g_2(x_2), ..., g_r(x_r)) = z$$

mit gegebenem k, v und z zu finden, wenn kein $g_s^{-1}(x)$ bekannt ist.

Beweis. Der Beweis ist analog zu dem der Sicherheit des Verfahrens, die auf der Kombinationsfunktion basiert. Für weitere Informationen siehe Korollar 5.1.

□

4.3 Kombinationsfunktion als Ring

Der Name Ringsignatur kommt daher, dass die Kombinationsfunktion in dem Verfahren zu einem Ring geschlossen wird. Die nötige Forderung, um die Kombinationsfunktion zu einem Ring zu biegen, ist $z = v$. Damit ergibt sich:

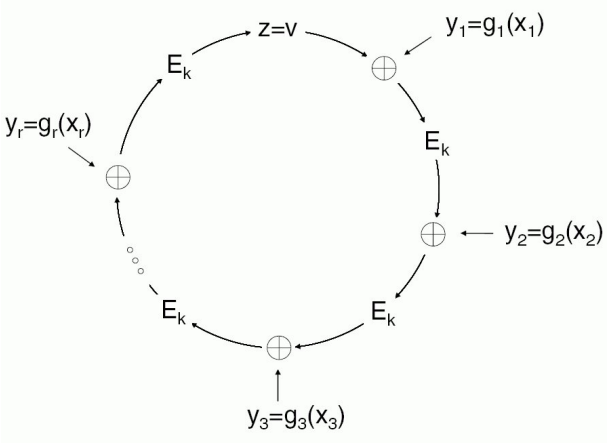

Abbildung 2: Die Kombinationsfunktion als Ring (RST06).

Es entsteht genau dann ein Ring, wenn die Signatur korrekt ist.

5 Verfahren unter Verwendung von RSA

Bei dem RSA Signaturverfahren handelt es sich wohl um das bekannteste und am meisten genutzte Signaturverfahren. Es existieren im Internet verschiedene Schüsselserver, auf denen öffentliche Schlüssel bereitgestellt werden können (Wät03). Dieses Kapitel erläutert kurz das RSA Signaturverfahren und geht dann auf die konkrete Erzeugung von Ringsignaturen unter Verwendung von RSA ein.

5.1 RSA Signaturverfahren

Das RSA Signaturverfahren mit öffentlichem Schlüssel (n, e) und privatem Schlüssel d wird in (RSA78) beschrieben. Es folgt ein kurzer Überblick über das Verfahren und die hier verwendete Notation.
Die Sicherheit des Verfahrens beruht auf der Annahme, dass es berechnungsmäßig praktisch unmöglich ist, das Produkt zwei großer Primzahlen zu faktorisieren. Es werden zwei große Primzahlen p und q gewählt, welche den Modulus $n = pq$ bestimmen. p und q sind geheim und werden nur zum Erzeugen der Schlüssel benötigt.
Der öffentliche Schlüssel zur Prüfung der Signatur ist ein zufällig gewähltes e für das gilt

$$1 < e < \varphi(n)$$

$$ggT(e, \varphi(n)) = 1.$$

Mit diesen Einschränkungen ist gesichert, dass ein $d = e^{-1}$ existiert, welches den geheimen Schlüssel darstellt. Die Berechnung von d kann nur mit dem Wissen von p und q stattfinden, da hierzu die Eulerfunktion $\varphi(n) = (p-1)(q-1)$ berechnet werden muss. Der private Schlüssel lautet

$$d = e^{-1} mod\ \varphi(n).$$

Damit ergeben sich die beiden Funktionen

$$f^{-1}(M) = M^d mod\ n = \sigma$$

$$f(\sigma) = \sigma^e\ mod\ n = M^{ed}\ mod\ n = M.$$

Es ist dabei M eine Nachricht, die mit der privaten Funktion $f^{-1}(m)$ signiert wird. Mit der öffentlichen Funktion $f(\sigma)$ kann die Signatur überprüft werden.

5.2 Ringsignatur

Gegeben ist eine Nachricht m, die signiert werden soll. Die möglichen Unterzeichner sind $A_1, A_2, ..., A_r$. Der Erzeuger der Signatur ist A_s mit $1 \leq s \leq r$. Zu jedem A_i bezeichnet P_i die Informationen, die für die Verifikation einer Signatur nötig sind. Dabei ist $g_i(x)$ die nach Abschnitt 4.1 erweiterte öffentliche Transformation des RSA Verfahrens. Weiterhin bezeichnet S_i die privaten Schlüsselinformationen, die benötigt werden, um $g_i^{-1}(y)$ zu bestimmen.

5.2.1 Erzeugen einer Ringsignatur

Der Unterzeichner A_s besorgt sich die öffentlichen Transformationen $P_i, i \neq s$ und erzeugt eine gültige Signatur für die Nachricht m.

1. Der symmetrische Schlüssel k für die Verschlüsselungsfunktion wird aus der zu signierenden Nachricht erzeugt.

$$k = h(m)$$

2. A_s wählt einen zufälligen Initialisierungsvektor $v \in \{0,1\}^b$, der den Ring zusammenführen soll (durch $z = v$).

3. Für jeden möglichen Unterzeichner generiert A_s die $r-1$ zufälligen Werte $x_i \in \{0,1\}^b, i \neq s$. Somit ergeben sich die y_i als
$$y_i = g_i(x_i), i \neq s.$$

4. A_s löst die Kombinationsfunktion mit den gegebenen Werten nach y_s auf. y_s ist nach Satz 4.2 bestimmbar aus der Gleichung:
$$C_{k,v}(y_1, y_2, ..., y_r) = v.$$

5. Mit der A_s bekannten, privaten Funktion $g_s^{-1}(y)$ erhält A_s
$$x_s = g_s^{-1}(y_s).$$

6. Die Ringsignatur der Nachricht m ist nun vollständig bestimmt und wird zusammengesetzt als:
$$(P_1, P_2, ..., P_r; v; x_1, x_2, ..., x_r).$$

5.2.2 Verifizieren einer Ringsignatur

Zum Verifizieren einer Ringsignatur erhält der Prüfer die Nachricht m und die korrespondierende Signatur σ. Er überprüft die Korrektheit der Signatur $\sigma = (P_1, P_2, ..., P_r; v; x_1, x_2, ..., x_r)$.

1. Der Prüfer wendet für alle $x_i, i = 1, 2, ..., r$ die Einwegfunktionen $g_i(x)$ an und erhält jeweils
$$y_i = g_i(x_i).$$

2. Durch Anwendung der Hashfunktion h erhält er den symmetrischen Schlüssel
$$k = h(m).$$

3. Mit einer Überprüfung der Gleichung
$$C_{k,v}(y_1, y_2, ..., y_r) = v$$
entscheidet der Prüfer, ob er die Signatur akzeptiert.

5.3 Sicherheit des Verfahrens

Die Sicherheit des Verfahrens besteht aus zwei Teilen. Als erstes muss gezeigt werden, dass die Anonymität des Erzeugers einer Signatur gewahrt wird. Der zweite Punkt ist die Korrektheit des Verfahrens. Die Korrektheit des Verfahrens bedeutet, dass es einem Angreifer außerhalb des Ringes nicht möglich ist, eine gültige Signatur zu erzeugen.

Satz 5.1. *Der Erzeuger einer Ringsignatur besitzt uneingeschränkte Anonymität unter den möglichen Unterzeichnern.*

Beweis. Es ist zu zeigen, dass alle Lösungen der Gleichung
$$C_{k,v}(y_1, y_2, ..., y_r) = v$$
für festes k und v mit gleicher Wahrscheinlichkeit auftreten und nicht vom Unterzeichner abhängen. Da $y_i \in \{0,1\}^b$ gilt, gibt es für jedes y_i (bis auf das letzte festzulegende y_i) genau 2^b

Möglichkeiten. Das r-te y_i ist durch Satz 4.1 eindeutig bestimmt. Es existieren also

$$L_{gesamt} = (2^l)^{r-1}$$

mögliche, gleich wahrscheinliche Lösungen.
Beim Erstellen von Signaturen werden in Schritt 3 die Werte für $r-1$ x_i festgelegt, diese Werte werden zufällig aus dem Bereich $\{0,1\}^l$ gewählt. Es ergeben sich demnach

$$L_{erzeugbar} = (2^l)^{r-1}$$

gleich wahrscheinliche Lösungen beim Erzeugen der Signatur. Diese Anzahl hängt nicht von dem Parameter s ab.
Da $L_{gesamt} = L_{erzeugbar}$ ist es auch mit uneingeschränkter Rechenzeit nicht möglich den Erzeuger aus der Signatur zu bestimmen.

□

Satz 5.2. *Das vorgestellte Verfahren ist im Modell des zufälligen Orakels berechnungsmäßig sicher gegen Angriffe mit adaptiv wählbaren Klartext.*

Anders formuliert liefert dieser Satz:

Korollar 5.1. *Es ist nur mit vernachlässigbarer Wahrscheinlichkeit möglich eine Ringsignatur für die Teilnehmer $A_1, A_2, ..., A_r$ zu erzeugen, ohne mindestens ein S_i bzw. $g_i^{-1}(x)$ zu kennen.*

Beweis. Die Korrektheit des Verfahrens wird im Modell des zufälligen Orakels bewiesen. Da die berechnungsmäßige Sicherheit bewiesen wird, verfügen alle Algorithmen nur über durch ein Polynom beschränkte Rechen- und Speicherkapazität.
Annahme: Es existiert ein Algorithmus A, der aus der Eingabe $g_1, g_2, ..., g_r$ eine gültige Ringsignatur für eine Nachricht m erzeugen kann.
Der Algorithmus A kann beliebige Anfragen an die Orakel E_k, E_k^{-1} und h stellen. Weiterhin hat er Zugriff auf ein Orakel, das für Nachrichten m' gültige Ringsignaturen σ' erzeugt. Dabei muss für die am Ende von A ausgegebene Signatur gelten $m \neq m' \; \forall m'$.
Nach Lemma 5.1 existiert ein Algorithmus A', der mit Hilfe von A eine gültige Ringsignatur für eine Nachricht m erzeugen kann, jedoch keine Zugriffe mehr auf das Ringsignatur-Orakel benötigt.
Weiterhin existiert mit Lemma 5.2 ein Algorithmus B, der mit Hilfe von Algorithmus A' ein $g_i^{-1}(y)$ für beliebiges y berechnen kann. Dies ist ein Widerspruch zu der Korrektheit des verwendeten Signaturverfahrens. Damit kann kein Algorithmus A mit den obigen Eigenschaften existieren.
Der Algorithmus A aus der Annahme ist in der Abbildung 3 illustriert.
Auf die Funktionen, die in den Sternen dargestellt sind, hat der Algorithmus gemäß der Theorie des Modells der Idealen Verschlüsselung und des Zufälligen Orakels Zugriff.

□

Lemma 5.1. *Es existiert ein Algorithmus A', der bei Existenz von A eine gültige Ringsignatur für eine Nachricht m erzeugen kann. A und A' benötigen hierzu keinen Zugriff auf andere Ringsignaturen.*

Beweis. Algorithmus A' lässt sich aus A erzeugen, indem er A als Blackbox verwendet und die Antworten der Orakel an A modifiziert. Diese Konstruktion folgt der Illustration in Abbildung 4. Die beiden Algorithmen benötigen keinen Zugriff mehr auf das Ringsignatur-Orakel. A benötigt jedoch diesen Zugriff. Anfragen von A werden durch A' mit einem zufälligen Vektor $(v, x_1, x_2, ..., x_r)$ beantwortet. Später modifiziert A' die Antworten auf Anfragen von A an die Orakel E_k und E_k^{-1} so, dass A jede gelieferte Signatur korrekt verifizieren kann. Diese Aufgabe wird durch die Funktionen F_k und F_k^{-1} erfüllt, welche die folgende Definition besitzen:

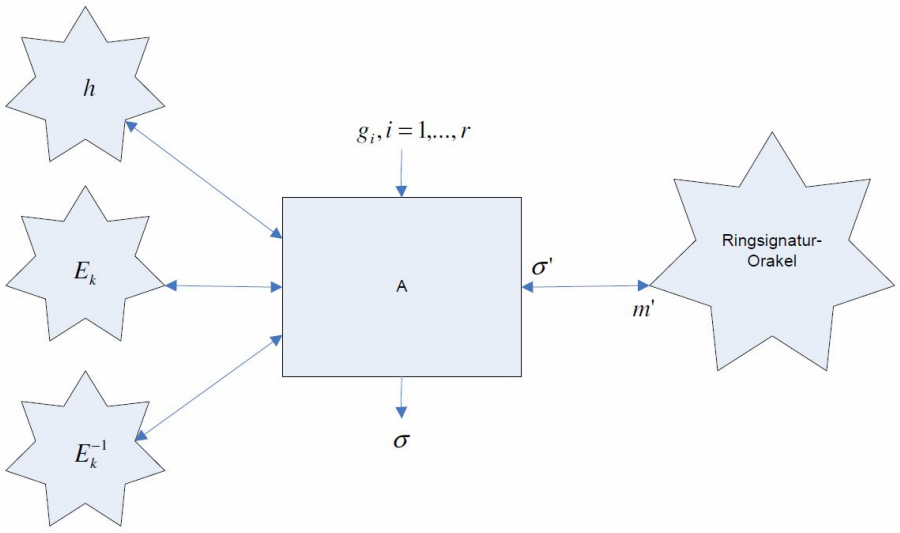

Abbildung 3: Algorithmus A, der gültige Ringsignaturen erzeugen kann.

$$F_k(y) = \begin{cases} z_{m,i+1} & \text{wenn } k = h(m), y = z_{m,i} \oplus g_{i+1}(x_{m,i+1}) \\ E_k(y) & \text{sonst} \end{cases}$$

$$F_k^{-1}(y) = \begin{cases} z_{m,i} \oplus g_{i+1}(x_{m,i+1}) & \text{wenn } k = h(m), y = z_{m,i+1} \\ E_k^{-1}(y) & \text{sonst} \end{cases}$$

Dabei gibt der zusätzliche Index m an, dass für jede signierte Nachricht m zusätzlich zu den $x_{m,i}$ aus der Signatur auch noch die $z_{m,i}$ generiert werden. Diese $z_{m,i}$ mit $i = 1, ..., r$ werden zufällig aus $\{0,1\}^b$ gewählt. F_k und F_k^{-1} benötigen auch ein $z_{m,0}$. Dieses ist vergleichbar mit dem Initialisierungsvektor der Kombinationsfunktion und wird $z_{m,0} := z_{m,r}$ gesetzt.
Nach Konstruktion schließt sich der "Ring" immer an einem $z_{m,i}$ für beliebiges i. Die Funktionen F_k und F_k^{-1} sorgen dafür, dass der Prüfer von beiden Richtungen, auf die er den Ring traversieren könnte, den selben Wert $z_{m,i}$ erhält.
Der Algorithmus A kann diesen "Betrug" nur mit vernachlässigbarer Wahrscheinlichkeit feststellen. A müsste schon vorher eine Anfrage der Form $E_{h(m)}(x_i)$ gestellt haben, mit einem x_i aus der Signatur. Da die x_i jedoch von A' zufällig aus allen Worten über $\{0,1\}^b$ gewählt werden, tritt dieser Fall nur mit vernachlässigbarer Wahrscheinlichkeit auf. Die Antworten von A' erscheinen A genau so zufällig wie die der Orakel.

□

Lemma 5.2. *Es existiert ein Algorithmus B, der bei Existenz von A' ein $g_i^{-1}(y)$ für beliebiges y berechnen kann.*

Beweis. Der Algorithmus B erhält als zusätzliche Eingabe ein y, für das ein $g_i^{-1}(y)$ bestimmt werden soll. Dabei ist g_i eine Funktion aus der Eingabe von A und A'. B benutzt den Algorithmus A' als eine Blackbox wie in der Abbildung 5 gezeigt. Dabei werden die Anfragen von A' an das Orakel E_k und E_k^{-1} abgefangen und die Antworten simuliert durch G_k und G_k^{-1} zurückgegeben. A kann die gefälschte Ringsignatur auf drei verschiedene Weisen aufbauen:

1. A baut den Ring im Uhrzeigersinn auf. Dies bedeutet, dass die zur Konstruktion des Ringes relevanten Anfragen nur an das Orakel E_k gehen.

2. A baut den Ring gegen Uhrzeigersinn auf. Dieser Fall ist analog zu dem obigen, so dass die zur Konstruktion des Ringes relevanten Anfragen nur an das Orakel E_k^{-1} gehen.

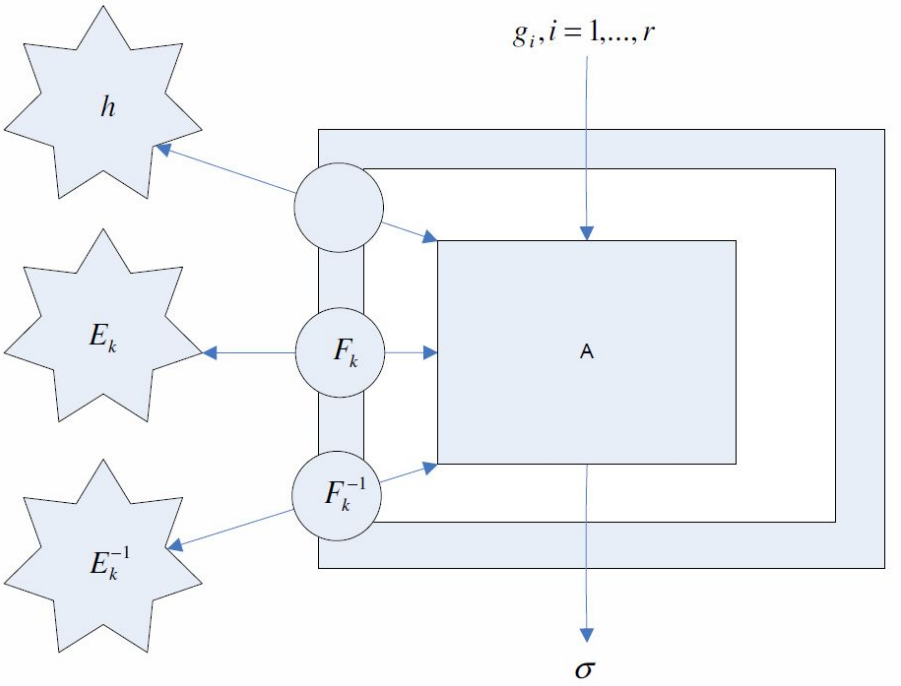

Abbildung 4: Algorithmus A', der durch A gültige Ringsignaturen erzeugen kann.

3. A berechnet einen Teil des Ringes im Uhrzeigersinn, den anderen gegen den Uhrzeigersinn. Dies ist durch Umformungen der Kombinationsfunktion wie im Beweis zu Satz 4.2 möglich.

Dieser Beweis bezieht sich nur auf den ersten Fall, deckt so aber auch den zweiten (analogen) Fall ab. Der dritte Fall findet sich in (RST06).
Es wird für das Erzeugen des gefälschten Ringes im Uhrzeigersinn durch A das Orakel E_k durch eine Funktion G_k des Algorithmus B ersetzt.

$$G_k(w) = \begin{cases} E_k^{-1}(r_j) \oplus y & \text{wenn } k = h(m^*), w = g_{j-1}(x_{j-1}) \oplus r_{j-2} \\ & \text{mit nicht vernachlässigbarer Wahrscheinlichkeit} \\ & m^*, j \text{ eindeutig bestimmt} \\ E_k(w) & \text{sonst} \end{cases}$$

Die Konstruktion von Algorithmus A' leitet die Anfragen von A an E_k weiter, wenn sie sich nicht auf eine von B gefälschte Signatur beziehen. Insbesondere werden alle Anfragen weitergeleitet, die A macht, um die Ringsignatur zu erstellen. Genau diese Anfragen werden von G_k abgefangen. Dabei muss der Algorithmus B "raten", welche der Nachrichten m diejenige ist, für die eine Ringsignatur erstellt werden soll. Dies ist mit nicht vernachlässigbarer Wahrscheinlichkeit möglich, da die Anzahl der möglichen Nachrichten durch ein Polynom nach oben beschränkt ist. Die ausgewählte Nachricht wird bezeichnet mit m^*.
Anfragen von A an G_k haben für die Elemente des Ringes die Form

$$r_i = G_k(g_i(x_i) \oplus r_{i-1}).$$

Dabei ist r_{i-1} der Teil des Ringes, der schon vorher berechnet wurde. Wenn A am Anfang des Ringes startet ist $r_0 = v$ der Initialisierungsvektor.
Der Punkt an dem der Ring geschlossen wird wird mit r_j bezeichnet. Wird mit dem Initialisierungsvektor begonnen, so wird der Ring bei $r_j = z = v$ geschlossen. Weiterhin ist r_{j-1} berechnet. Damit der Ring geschlossen werden kann, muss A den Wert von $g_j(x_j)$ entsprechend anpassen. Der genaue Wert von j ist dem Algorithmus B nicht bekannt. B kann die Anfrage auch nicht

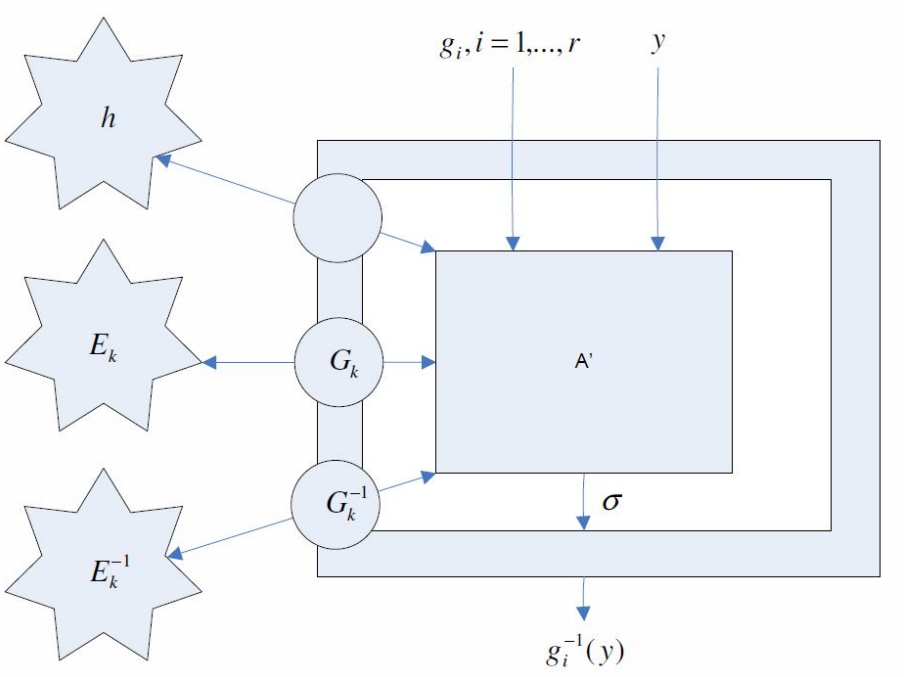

Abbildung 5: Algorithmus B, der durch A' ein $g_i(x)$ invertieren kann.

erkennen, da ihm selbst x_l nicht bekannt ist. Der Zeitpunkt der Anfrage muss von B "geraten" werden. Da die Anzahl der Anfragen nur durch ein Polynom beschränkt ist, gelingt das "Raten" mit nicht vernachlässigbarer Wahrscheinlichkeit.
B fälscht durch G_k die Berechnung von r_{j-1} als

$$r_{j-1} = G_k(g_{j-1}(x_{j-1}) \oplus r_{j-2}) = y \oplus E_k^{-1}(r_j).$$

Damit bleibt A zum Schließen des Ringes nichts anderes übrig, als die Gleichung

$$r_j = E_k(g_j(x_j) \oplus r_{j-1})$$

also

$$r_j = E_k(g_j(x_j) \oplus y \oplus E_k^{-1}(r_j))$$

für x_j zu lösen. Es ergibt sich

$$\begin{aligned} E_k^{-1}(r_j) &= g_j(x_j) \oplus y \oplus E_k^{-1}(r_j) \\ E_k^{-1}(r_j) \oplus E_k^{-1}(r_j) \oplus y &= g_j(x_j) \\ y &= g_j(x_j) \end{aligned}$$

und daraus als einzige mögliche Lösung

$$x_j = g_j^{-1}(y).$$

B kann demnach mit nicht vernachlässigbarer Wahrscheinlichkeit eine der Funktionen g_i invertieren. Die gesuchte Lösung liefert A' als Teil der Ringsignatur, die alle x_i enthält.

□

6 Verfahren mit Rabin Signatur

Das RSA Signaturverfahren ist weit verbreitet und es wird pro Signatur eine modulare Exponentiation benötigt. Ein weiteres Verfahren, das auf einer ähnlichen Sicherheitsannahme beruht, ist das Rabin Signaturverfahren (Rab79). Dieses ist im Gegensatz zum RSA Verfahren effizienter, da zur Prüfung der Signatur nur eine modulare Multiplikation benötigt wird. Das Erstellen einer einfachen Signatur ist ungefähr so komplex wie beim RSA Verfahren.
Für das Rabin Verfahren treffen jedoch nicht alle Annahmen zu, die auch für das RSA Verfahren gelten. Aus diesem Grund muss im zweiten Teil dieses Abschnitts bewiesen werden, dass trotzdem eine uneingeschränkte Anonymität beibehalten wird.

6.1 Rabin Signaturverfahren

Für das Rabin Signaturverfahren hat der private Schlüssel die Form (p, q). Bei p und q handelt es sich um ungefähr gleich große Primzahlen, die multipliziert den Modulus $n = pq$ ergeben. Wenn eine Signatur erstellt werden soll, wird die private Funktion

$$f^{-1}(x) = \sqrt{(x)}\ mod\ n = \sigma, (fr)x \in R_n$$

verwendet. Dabei bezeichnet R_n die Menge der quadratischen Reste modulo n. Zur effizienten Bestimmung einer Quadratwurzel muss die Faktorisierung von $n = pq$ bekannt sein. Die öffentliche Transformation zur Prüfung einer Signatur ist

$$f(x) = \sigma^2\ mod\ n = \sqrt{(x)}^2\ mod\ n = \sigma.$$

Die gegebenen Funktionen beziehen sich auf eine leichte Vereinfachung des Rabin Verfahrens. Dabei wird vor dem Signieren keine Redundanzfunktion auf die Eingabe angewendet (beide Möglichkeiten finden sich in (Wät03)).

6.2 Modifikation der Signaturerzeugung

Der Unterschied zum RSA Verfahren besteht darin, dass das Rabin Verfahren keine Bijektion über $\mathbb{Z}_n$ ist. Nur die Zahlen aus R_n besitzen eine Quadratwurzel in $\mathbb{Z}_n$.
Es kann bei der Erzeugung einer Ringsignatur im vorletzten Schritt vorkommen, dass

$$x_s = g_s^{-1}(y_s)$$

nicht definiert ist. In diesem Fall muss A_s die letzte zufällige Wahl des x_i mit $i = s + 1$ oder $i = s - 1$ erneut treffen. So verändert A_s das y_s, bis gilt $y_s \in R_n$. Die Wahrscheinlichkeit für $y_s \in R_n$ liegt nach Lemma 6.1 ungefähr bei $\frac{1}{4}$.

6.3 Beibehaltung der uneingeschränkten Anonymität

Der Beweis für die uneingeschränkte Anonymität aus Satz 5.1 beruht auf der Annahme, dass alle Lösungen der Kombinationsfunktion gleichwahrscheinlich auftreten.

Lemma 6.1. *Die Anzahl der quadratischen Reste modulo $n = pq$ mit p und q große Primzahlen beträgt ungefähr $\frac{n}{4}$.*

Beweis. Die Menge der quadratischen Reste modulo n bezeichnet mit R_n ist isomorph zu der Menge $R_p \times R_q$. Für die Menge R_p mit p prim gilt

$$|R_p| = \frac{p-1}{2}.$$

Damit ergibt sich

$$\begin{aligned} |R_n| &= |R_p \times R_q| \\ &= \tfrac{p-1}{2} * \tfrac{q-1}{2} \\ &= \tfrac{pq-p-q+1}{4} \\ &\approx \tfrac{pq}{4} \\ &= \tfrac{n}{4} \end{aligned}$$

□

Die weitere Gültigkeit von Satz 5.1 für das Rabin Verfahren ist mit Satz 6.1 gezeigt. Dabei bezeichnet das verwendete x'_i den Teil von x_i, der sich auf den Bereich aus n_i bezieht. Die Erweiterung aus Definition 4.1 ist unabhängig davon möglich. Der relevante Teil des folgenden Satzes bezieht sich jedoch auf die eigentlichen Signaturfunktionen $f^-1(x)$.

Satz 6.1. *Die Wahrscheinlichkeit der Wahl eines $x'_{i\neq s}$ ist sehr nahe an der für die Wahl des x'_s.*

Beweis. Der Erzeuger der Ringsignatur A_s wählt die $x'_{i\neq s}$ weiterhin zufällig aus der Menge $\mathbb{Z}_{n_i}$. Es ergibt sich somit

$$\text{Wahrscheinlichkeit(Wahl von } x'_{i\neq s}) = \frac{1}{n_i}.$$

Zum Berechnen des passenden x'_s löst A_s die Gleichung $f_s^{-1}(y'_s) = \sqrt{(}y'_s)\ mod\ n$ für x'_s. Es existieren nach Lemma 6.1 ca. $\frac{n_s}{4}$ gleich wahrscheinliche Möglichkeiten für y'_s. Für $\sqrt{(}y'_s)\ mod\ n$ erhält A_s jeweils 4 Möglichkeiten (modulare Quadratwurzeln), aus denen A_s zufällig eine auswählt. Damit ergibt sich die Wahrscheinlichkeit für die Wahl von x'_s zu

$$\text{Wahrscheinlichkeit(Wahl von } x'_s) \approx \frac{4}{n_s} * \frac{1}{4} = \frac{1}{n_s}.$$

□

7 Effizienzbetrachtung und Bewertung

Das vorgestellte Verfahren zum Erzeugen von Ringsignaturen ist sehr effizient gegenüber anderen Möglichkeiten (vgl. (RST06)). Beim Erzeugen einer Signatur wird $(r-1)$ mal eine Funktion g_i ausgewertet und einmal die Funktion g_s^{-1}. Es sind r Anwendungen von E_k oder E_k^{-1} nötig, um den Ring zu schließen.
Zum Erstellen einer Signatur mit dem RSA Verfahren werden demnach nur r modulare Exponentiationen durchgeführt. Es empfiehlt sich jedoch (wenn möglich) das Rabin Verfahren zu verwenden, da hier $r-1$ mal eine Exponentiation durch eine modulare Multiplikation ersetzt werden kann.
In der Praxis ist es jedoch wahrscheinlicher, dass Ringsignaturen mit dem RSA Verfahren eingesetzt werden. Der Hauptgrund liegt darin, dass es sich bei der Erzeugung von Ringsignaturen um ein Verfahren handelt, in dem nicht die Mithilfe anderer möglicher Unterzeichner benötigt wird. Der Erzeuger muss also auf die veröffentlichten Schlüsselinformationen der anderen zugreifen. Dabei handelt es sich meistens um RSA Schlüssel.
Weiterhin von Interesse ist der verbrauchte Speicherplatz. Eine Signatur nach dem beschriebenen Schema hat eine von der Größe des Ringes linear abhängige Größe. Es gibt jedoch Verfahren, die es schaffen, eine konstante Größe zu verwenden (DKNS04).

Durch die vielseitige Einsetzbarkeit des hier vorgestellten Schemas, sollte es genug Möglichkeiten für eine Verwendung im Alltag geben. Nach dem Erscheinen einer ersten Fassung von (RST06) im Jahr 2001 sind viele weitere Arbeiten zum Thema Ringsignaturen entstanden. Dabei handelt es sich vor allem um Erweiterungen auf andere Signaturverfahren und Vereinfachungen der Kombinationsfunktion.
Der Charakter des Verfahrens - als Möglichkeit Geheimnisse und Informationen zu verraten - macht es jedoch fraglich, dass höhere Instanzen dieses Verfahren unterstützen. Falls die Pressefreiheit und das Recht der freien Meinungsäußerung nicht mehr gewahrt werden sollten, schaffen die Ringsignaturen kombiniert mit anderen Verfahren einen nötigen Ausgleich.

Literatur

[Bla05] John Black. *The Ideal-Cipher Model, Revisited: An Uninstantiable Blockcipher-Based Hash Function.* Cryptology ePrint Archive, Report 2005/210 . 2005.

[CGH98] Ran Canetti, Oded Goldreich, Shai Halevi. *The Random Oracle Methodology, Revisited.* 1998.

[CV91] David Chaum and Eugene Van Heyst. *Group signatures.* In D.W. Davies, editor, *Advances in Cryptology / Eurocrypt '91*, pages 257-265, Berlin, 1991. Springer-Verlag. Lecture Notes in Computer Science No. 547.

[DH76] W. Diffe and M. E. Hellman. *New directions in cryptography. IEEE Trans. Inform. Theory*, IT-22:644-654, November 1976.

[DKNS04] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup *Anonymous Identification in Ad-Hoc Groups.* In *em EUROCRYPT 2004, LNCS 3027*, pages 609-626. Springer-Verlag, 2004.

[GRS99] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. *Onion Routing for Anonymous and Private Internet Connections.* In *Communications of the ACM 42(2)*, pages 39-41, 1999.

[MP93] M. Bellare and P. Rogaway. *Random Oracles are Practical: a Paradigm for Designing Efficient Protocols.* In *ACM Conference on Computer and Communications Security* 1993, pages 62-73.

[Rab79] M. Rabin. *Digitalized signatures as intractable as factorization. Technical Report MIT/LCS/TR-212*, MIT Laboratory for Computer Science, January 1979.

[Rin06] Jan Oliver Ringert. *Leaking the secret of snow white: How to explain ring signatures to your children.* TU-Braunschweig 2006. `http://www.iti.cs.tu-bs.de/TI-INFO/waetjen/Seminar0607/howtoexplain.pdf`.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. *A method for obtaining digital signatures and public-key cryptosystems.* Communications of the ACM, 21(2):120-126, 1978.

[RST06] Ronald L. Rivest, Adi Shamir, and Yael Tauman. *How to Leak a Secret: Theory and Applications of Ring Signatures.* In O. Goldreich, A. Rosenberg, and A. Selman, editors, *Essays in Theoretical Computer Science: in Memory of Shimon Even*, volume 3895 of LNCS Festschrift. Springer- Verlag, 2006.

[Sti95] D.R. Stinson. *Cryptography - Theory and Practice.* CRC Press, Boca Raton. 1995.

[Wät03] Dietmar Wätjen. *Kryptographie: Grundlagen, Algorithmen, Protokolle.* 1. Auflage, Heidelberg: Spektrum Akad. Verl., 2003.