Das RSA-Kryptosystem

Jannis Ihrig

16. Februar 2011

1 Asymmetrische Verschlüsslungsverfahren

Traditionelle symmetrische Verschlüsslungsverfahren (private-key cryptosystems) nutzen sowohl für das Verschlüsseln wie auch für das Entschlüsseln einer Nachricht den selben Schlüssel. Daraus ergeben sich zwei Probleme:

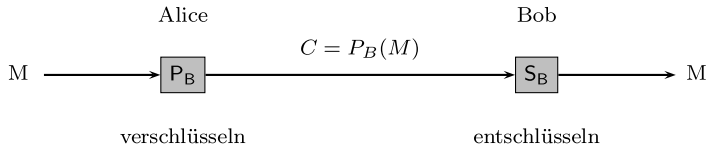
1. Für den Austausch des Schlüssels muss ein sicherer Kanal existieren.
2. Bei n Teilnehmern werden $O(n^2)$ Schlüssel benötigt.

Bei asymmetrischen Verschlüsslungsverfahren (public-key cryptosystems) besitzt jeder Teilnehmer sowohl einen öffentlichen als auch einen privaten Schlüssel (P, S). P wird veröffentlicht, so dass er jedem Teilnehmer zugänglich ist, während S nur dem jeweiligen Besitzer zugänglich sein sollte. Ein Schlüsselpaar P, S spezifiziert ein korrespondierendes Paar injektiver Funktionen $P()$, $S()$, welche auf eine Nachricht M angewandt werden können. Sei z.B. $M \in \mathbb{B}^*$, so bilden $P()$, $S()$ Funktionen von $\mathbb{B}^*$ nach $\mathbb{B}^*$. Dabei gilt für $P()$ und $S()$

$$M = P(S(M)) \tag{1}$$
$$M = S(P(M)) \tag{2}$$

1.1 Verschlüsselung

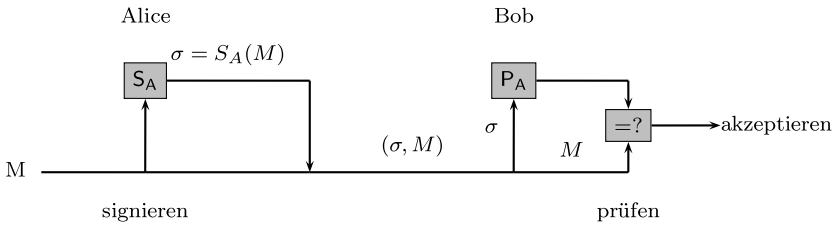
Angenommen Alice und Bob möchten verschlüsselt kommunizieren, wobei Alice die Schlüssel P_A, S_A und Bob die Schlüssel P_B, S_B besitzt. Die Verschlüsselung einer Nachricht M mittels asymmetrischer Verfahren funktioniert nach folgendem Schema:



Druch die Verschlüsselung kann die Vertraulichkeit und Integrität der Nachricht erreicht werden.

1.2 Signatur

Zusätzlich können Nachrichten mit einigen asymmetrischen Verschlüsselungsverfahren signiert werden. Durch Signaturen kann die Authentizität und Verbindlichkeit einer Nachricht erreicht werden. Das signieren einer Nachricht M funktioniert wie folgt:



Aus dem Schema wird ersichtlich, dass die signierte Nachricht nur unverschlüsselt übermittelt wird. Erstellt Alice jedoch zunächst eine Signatur und verschlüsselt anschliessend Signatur und Nachricht, so können beide Verfahren kombiniert werden.

2 Theoretische Voraussezungen

Die Theoretische Informatik liefert wichtige Grundlagen für moderne Verschlüsselungstheorien. So ermöglicht die Annahme der Existenz der im folgenden beschriebenen trap-door Funktionen erst die Entwicklung von asymmetrischen Verschlüsslungsverfahren.

2.1 Definition

Gegeben sei eine Funktionsschar $\{f_i\}, i \in \Sigma^*$. Diese wird durch die Funktion $f : \Sigma^* \times \Sigma^* \to \Sigma^*$ dargestellt, wobei für jedes $i, w \in \Sigma^*$ gilt: $f(i, w) = f_i(w)$.
Die Funktion f sei als längenerhaltend, wenn alle Funktionen f_i längenerhaltend sind.

2.2 Definition

Eine ***probabilistische TM*** M ist eine nichtdeterministische TM, die bei jedem Schritt mit einem Münzwurf zwischen zwei Ausführmöglichkeiten entscheidet. Verbildlicht, spannen die möglichen Wege, die M bei einer Berechnung wählen kann, einen Baum auf. Die Wahrscheinlichkeit, dass M einen Ast b des Baum wählt ist:

$$Pr[b] = 2^{-k}$$

Demnach ist die Wahrscheinlichkeit, dass M ein Wort $w \in \sigma$ akzeptiert:

$$Pr[\text{M akzept. w}] = \Sigma_{\text{b akzept.}} Pr[b]$$

2.3 Definition

Eine ***trap-door Funktion*** ist eine längenerhaltende Funktion $f : \Sigma^* \times \Sigma^* \to \Sigma^*$ mit einer probabilistischen TM G und einer Hilfsfunktion $h : \Sigma^* \times \Sigma^* \to \Sigma^*$. Für f, G und h muss gelten:

1. Die Funktionen f und h sind in polynomieller Zeit berechenbar.
2. Für jede probabilistische TM E, jedes k und ein genügend großes n, gilt für eine beliebige Ausgabe $\langle i, t\rangle$ von G für die Eingabe 1^n und ein beliebiges Wort $w \in \Sigma^*$ folgendes:

   $$\Pr[E(i, f(i, w)) = y, \text{ wobei } f(i, y) = f(i, w)] \leq n^{-k}$$

3. Für jedes n, jedes w mit $|w| = n$ und jede Ausgabe $\langle i, t\rangle$ von G, welche bei beliebiger Eingabe mit einer Wahrscheinlichkeit größer 0 ausgegeben wird, gilt:

   $$h(t, f(i, w)) = y, \text{ wobei } f(i, y) = f(i, w)$$

   Die TM G dient ausschliesslich zum Generieren des Indexes i der Funktion f sowie t, sodass f_i mit Hilfe von h effektiv invertiert werden kann.

3 RSA

Im Folgenden wird genauer auf das RSA-Kryptosystem eingegangen. Um die Korrektheit von RSA zu zeigen, werden einige mathematische Grundlagen benötigt. Auf die Beweise für die hier benutzen Sätze wird jedoch aus Zeit- und Komplexitätsgründen verzichtet.

3.1 Definition

Die ***Eulersche ϕ -Funktion*** gibt für eine Zahl $n \in \mathbb{Z}$ an, wie viele Zahlen $a \in \mathbb{Z}^*$ mit $a \leq n$ zu ihr teilerfremd sind.

$$\phi(n) = n \prod_{p|n} (1 - \frac{1}{p})$$

Ist p eine Primzahl, so ist p zu allen Zahlen $a \leq n$ teilerfremd und es gilt

$$\phi(p) = p - 1$$

Da ϕ multiplikativ ist, gilt für eine Zahl $n = m \cdot k$ zudem

$$\phi(n) = \phi(m) \cdot \phi(k)$$

3.2 Theorem

Für jedes $n > 1$, hat die Gleichung $ax \equiv 1 \pmod{n}$ in $\mathbb{Z}_n$ eine eindeutige Lösung modulo n, falls $ggT(a, n) = 1$. Ansonsten hat sie keine Lösung.

3.3 Schlüsselgenerierung

Im RSA-Kryptosystem erstellt sich jeder Teilnehmer ein Schlüsselpaar (S, P) nach folgendem Algorithmus:

1. Wähle zwei große Primzahlen p und q mit $p \neq q$.
2. Berechne n durch $n = p \cdot q$.
3. Wähle eine ungerade Zahl e, welche zu $\phi(n)$ teilerfremd ist. Nach der Definition von ϕ entspricht $\phi(n) = (p-1) \cdot (q-1)$.
4. Berechne d als das multiplikative Inverse von $e \pmod{\phi(n)}$. Nach dem Theorem aus 3.2 existiert d und ist eindeutig bestimmt.
5. Gib das Paar $P = (e, n)$ als öffentlichen Schlüssel zurück.
6. Gib das Paar $S = (d, n)$ als privaten Schlüssel zurück.

Im Folgenden ist $M \in \mathbb{Z}_n$. Alle Rechnungen finden dementsprechend ebenfalls in $\mathbb{Z}_n$ statt. Der generierte öffentliche Schlüssel spezifiziert bei RSA die folgende Funktion, welche auf M angewandt wird.

$$P(M) = M^e \pmod{n} \tag{3}$$

Der private Schlüssel S spezifiziert die folgende Funktion, die auf eine verschlüsselte Nachricht C angewandt wird.

$$S(C) = C^d \pmod{n} \tag{4}$$

Zum Signieren wird wie im Abschnitt 1 beschrieben $S()$ auf eine Nachricht angewandt und die Signatur mit $P()$ überprüft.

3.4 Theorem (Kleiner Satz von Fermat)

Ist p eine Primzahl, so gilt für alle $a \in \mathbb{Z}_p^*$:

$$a^{p-1} \equiv 1 \pmod{p}$$

3.5 Korollar (nach Chinesischem Restsatz)

Sind $n_1, \ldots, n_k$ paarweise teilerfremd und $n = n_1, \ldots, n_k$, so gilt für alle Zahlen $x, a \in \mathbb{Z}$

$$x \equiv a \pmod{n_i}$$

für $1 \leq i \leq k$, genau dann wenn

$$x \equiv a \pmod{n}$$

Dieses Korollar folgt direkt aus dem konstruktiven Beweis des Chinesischen Restsatzes.

3.6 Theorem (Korrektheit von RSA)

Die den Schlüsseln S und P entsprechenden Funktionen (3) und (4) sind zueinander invers und erfüllen somit die im Abschnitt 1 aufgestellten Gleichungen (1) und (2).

Beweis

Zu Zeigen ist P(S(M)) = S(P(M)). Nach den Gleichungen (3) und (4) gilt:

$$P(S(M)) = S(P(M)) = M^{ed} \pmod{n}.$$

Wir werden zunächst $M^{ed} \equiv M \pmod{p}$ und $M^{ed} \equiv M \pmod{q}$ beweisen. Daraus ergibt sich unter Anwendung des Korollars von 3.5 der Beweis für $M^{ed} \equiv M \pmod{n}$.

e und d bilden multiplikative Inverse modulo $\phi(n) = (p-1)(q-1)$. Danach gilt für alle $k \in N$:

$$ed = 1 + k(p-1)(q-1)$$

Im Fall $M \not\equiv 0 \pmod{p}$ gilt

$$\begin{aligned} M^{ed} &\equiv M(M^{p-1})^{k(q-1)} \pmod{p} \\ &\equiv M(1)^{k(q-1)} \pmod{p} \quad \textit{nach Kleinem Satz von Fermat} \\ &\equiv M \end{aligned}$$

Im Fall $M \equiv 0 \pmod{p}$ ist augenscheinlich

$$M^{ed} \equiv M \pmod{p}$$

Der Beweis für $M^{ed} \equiv M \pmod{q}$ erfolgt analog. Nach Korollar 3.5 gilt nun

$$M^{ed} \equiv M \pmod{n}$$

für alle Nachrichten M.

3.7 Laufzeit

Die für das Ver- und Entschlüsseln benötigte Zeit hängt von der Methode zum Exponenzieren der Nachricht M ab. Im Folgenden wird $a^b \mod n$ berechnet, b sei der Exponent in Binärdarstellung.

```
1  MODULAR-EXPONENTIATION(a,b,n)
2    d = 1
3    for i = b.length downto 0
4      d = (d*d) mod n
5      if b[i] == 1
6        d = (d*a) mod n
7    return d
```

Ist i die Bitlänge des Exponenten b, so benötigt der gegebene Algorithmus zum modularen Exponentieren $O(i)$ arithmetische Operationen. Durch das Anwenden von mod n auf das Ergebnis jeder Operation, besitzt d durchgehend eine Bitlänge von $\log n$. Demnach sind $P()$ und $S()$ in polynomieller Zeit auf ein M anzuwenden.

3.8 RSA als trap-door Funktion

Nach vorheriger Definition besteht eine trap-door Funktion aus einer Funktion f, einer probabilistischen TM G und einer Hilfsfunktion h. Im Fall von RSA generiert G nach oben gegebenen Algorithmus die Ausgabe $((n,e),d)$. Der Index der Funktion f besteht aus n und e. f ist somit:

$$f((n,e),w) = w^e \mod n.$$

Die Hilfsfunktion h zum invertieren von f lautet:

$$h(d,x) = x^d \mod n.$$

3.9 Sicherheit

Im heutigen Ansatz ist ein sicheres Verschlüsselungsverfahren eines, dessen Verschlüsselung nicht gebrochen weden kann, solange die in der Nachricht enthaltene Information einen Wert besitzt. Die Sicherheit von RSA beruht vor allem auf der Schwierigkeit, große Zahlen effizient zu faktorisieren. Es ist ungewiss, ob ein effizienter Algorithmus für dieses Problem existiert, alle bis jetzt bekannten Verfahren besitzen exponentielle Laufzeiten. Aus diesem Grund kann bei einer genügenden Schlüssellänge (heute ca. 2048 Bit) von einer ausreichenden Sicherheit ausgegangen werden. Zudem ist es bis jetzt nicht möglich, die Sicherheit eines Verschlüsselungsalgorithmus nachzuweisen. Im Fall von RSA bedeutet dies, dass kein anderer Ansatz zum Brechen der Verschlüsselung, als das Faktorisieren von n bekannt ist.