DIE RSA-VERSCHLÜSSELUNG

VON
SEBASTIAN HALLMANN

Quellen: CrypTool-Skript Prof. Bernhard Esslinger (www.cryptool.org),
RSA-Verschlüsselung von Johannes Becker, Wikipedia

Gliederung

- RSA-Verschlüsselung
  - Entwicklung
  - Algorithmus
    - Erweiterter euklidischer Algorithmus
  - Beispiele

RSA-Verschlüsselung

Entwicklung

RSA wurde 1977 von Ronald L. **R**ivest, Adi **S**hamir und Leonard **A**delman entwickelt.

Es ist ein asymmetrisches Verschlüsselungsverfahren, welches sowohl zum Verschlüsseln als auch zum Signieren geeignet ist.

Im Jahre 1976 legten W. Diffie und M. Hellman, durch Herausgabe Ihres Buches „New Directions in Cryptography“ einen Grundstein in der modernen Kryptographie.
Aus diesem Material wurden zum ersten Mal Ideen zu einem Public-Key-Verfahren entwickelt, jedoch noch keine Lösungsansätze.

Erst 1978 gelang es den drei Forschern den wohl bekanntesten und bis heute sichersten Public-Key-Algorithmus vorzustellen.

Das Verfahren basiert auf dem aktuellen Wissensstand, dass die Faktorisierung einer großen Zahl, also ihre Zerlegung in Primfaktoren, eine sehr aufwendige Angelegenheit ist, während das Erzeugen einer Zahl durch Multiplikation zweier Primzahlen recht einfach ist.

Algorithmus

Man kann die Einzelschritte zur Durchführung des RSA-Verfahrens folgendermaßen beschreiben. Schritt 1 bis 3 sind die Schlüsselerzeugung, Schritt 4 und 5 sind die Verschlüsselung und 6, 7 die Entschlüsselung.

1. Wähle zufällig zwei verschiedene Primzahlen p und q und multipliziere diese (n = p * q). Der Wert n wird als RSA-Modul bezeichnet.

2. Wähle ein beliebiges $e \in \{2,..,n\text{-}1\}$, so dass e teilerfremd zur Eulerschen φ-Funktion von n ist.
   φ-Funktion J(n) = (p-1) * (q-1), danach kann man p und q „wegwerfen“.

3. Berechne $d \in \{1,...,n\text{-}1\}$ aus der multiplikativen Inversen zu e modulo J(n) oder mit dem erweiterten euklidischen Algorithmus, so das gilt: $e * d \equiv 1(\text{ mod } J(n))$

   → (n, e) ist der öffentliche Schlüssel
   → (n, d) ist der private Schlüssel

4. Zum Verschlüsseln wird die als Zahl dargestellte Nachricht in Teile aufgebrochen, so dass jede Teilzahl kleiner als n ist.

5. Verschlüsselung des Klartextes (bzw. seiner Teilstücke)

6. Zum Entschlüsseln wird das als Zahl dargestellte Chiffrat in Teile aufgebrochen, so dass jede Teilzahl kleiner als n ist.

7. Entschlüsselung des Chiffretextes (bzw. seiner Teilstücke)

Erweiterter euklidischer Algorithmus

Ist ein Algorithmus aus dem mathematischen Teilgebiet der Zahlentheorie.
Er berechnet neben dem größten gemeinsamen Teiler ggT(e, J(n)) zweier natürlicher Zahlen e und n noch zwei ganze Zahlen s und t, die die folgende Gleichung erfüllen.

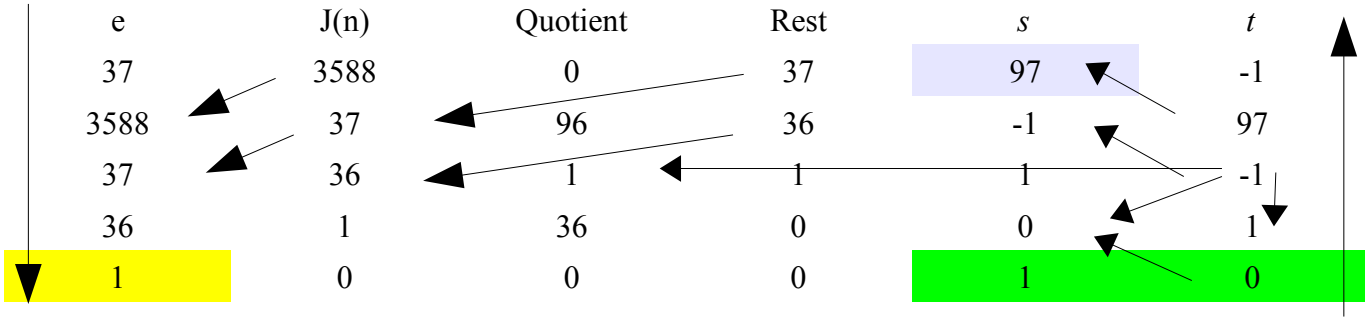
$$ggT(e, J(n)) = s * e + t * J(n)$$

Die multiplikative Inverse von e modulo J(n) kann nur bestimmt werden, wenn der ggT(e, J(n)) = 1 ist.
Dies bedeutet, dass e und J(n) teilerfremd zu einander sind, d.h. e und J(n) haben keine gemeinsamen Primfaktoren.
Die multiplikative Inverse ist diejenige Zahl (d), welche bei der Multiplikation das Ergebnis 1 liefert.

$$e * d \equiv 1 \pmod{J(n)} \rightarrow e * d \bmod J(n) = 1 \qquad 1 \bmod J(n) = 1$$

Um die multiplikative Inverse bestimmen zu können muss zu erst der ggT berechnet werden. Danach kann die Inverse (s) bestimmt werden.

Beispiel:



Der ggT berechnet sich durch die ganzzahlige Division mit Rest von e / J(n) = Quotient.
Dabei wird J(n) zu e und der *Rest* zu J(n) bis J(n) = Null ist.
In der linken unteren Ecke steht der ggT.
Nun kann die Inverse bestimmt werden. Dazu wird in die unterste Zeile bei s eine 1 und bei t eine 0 eingetragen. Jetzt arbeitet man die Tabelle von unten nach oben durch.
Dabei wird t_{alt} immer zu s und t ergibt sich aus $s_{alt} - q * t_{alt}$.
Dieser Schritt wird solange wiederholt bis die Tabelle ausgefüllt ist.
In der ersten Zeile unter s findet man den gesuchten Wert d.

Beispiele

RSA mit kleinen Primzahlen und mit einer Zahl als Nachricht

1. Die gewählten Primzahlen seien p = 5 und q = 11.
   Also ist n (p*q) = 55 und J(n) = (p – 1) * (q – 1) = 40.

2. e = 7 (muss teilerfremd zu 40 sein).

3. d = 23 (da 23 * 7 ≡ 161 ≡ 1 (mod 40))

   → Öffentlicher Schlüssel des Senders: (55, 7),
   → Privater Schlüssel des Empfängers: (55, 23).

4. Nachricht sei „nur“ die Zahl M = 2.

5. Verschlüsseln: $C \equiv 2^7 \equiv 18 \pmod{55}$.

6. Chiffrat ist die Zahl C = 18.

7. Entschlüsseln: $M \equiv 18^{23} \equiv 2 \pmod{55}$. → M = 2

RSA mit etwas größeren Zahlen und einem Text aus Großbuchstaben

1. Die gewählten Primzahlen seien p = 47 und q = 79.
   Also ist n (p*q) = 3713 und J(n) = (p – 1) * (q – 1) = 3588.

2. e = 37 (muss teilerfremd zu 3588 sein).

3. d = 97 (da 97 * 37 ≡ 3589 ≡ 1 (mod 3588))

   → Öffentlicher Schlüssel des Senders: (3713, 37),
   → Privater Schlüssel des Empfängers: (3713, 97).

4. Nachricht sei der Text M = ATTACK AT DAWN.

5. Verschlüsseln: Nach Zeichensatz (Blank 00, A=01,...,Z=26)

Text:	A	T	T	A	C	K		A	T		D	A	W	N
Zahl:	01	20	20	01	03	11	00	01	20	00	04	01	23	14

   Diese 28-stellige Zahl wird auf Grund der Bedingung, dass ein „Teil“ nicht größer sein darf als *n* in 4-stellige Teile zerlegt.
   (0120 2001 0311 0001 2000 0401 2314) sind alle kleiner als 3713

   Jeder der 7 Teile wird mittels $C \equiv M^{37} \pmod{3713}$ verschlüsselt.

6. Das Chiffrat ist die Zahlen C = 1404293235360001328422802235.

7. Entschlüsseln: C wird wieder in 4-stellige Teile zerlegt.
   (1404 2932 3536 0001 3284 2280 2235)

   Jeder der einzelnen Teile wird mittels $M \equiv C^{97} \pmod{3713}$ entschlüsselt:

   0120 2001 0311 0001 2000 0401 2314
   A T T A C K A T D A W N

Bei den Werten ist es für Kryptoanalytiker einfach, aus den öffentlichen Parametern n = 3713 und e = 37 die geheimen Werte zu finden, indem er offenlegt, dass 3713 = 47 * 79 ist.

Wenn *n* eine 768-Bit-Zahl ist, bestehen dafür – nach heutigen Kenntnissen – wenig Chancen.