Inhalt

Aus den Artikeln „Symmetrie vs. Asymmetrie – Teil I“ und „Symmetrie vs. Asymmetrie – Teil II“ wissen wir bereits, dass wir es bei **RSA** mit einer asymmetrischen Verschlüsselung zu tun haben. Im Gegensatz zu symmetrischen Verfahren, bei denen zum Verschlüsseln dasselbe „Passwort“ wie beim Entschlüsseln verwendet wird, haben wir es hier mit einem Privaten Schlüssel und einem Öffentlichen Schlüssel zu tun. Auch die mathematische Grundlage haben wir in „Primzahlen – Bausteine der natürlichen Zahlen“ ganz kurz angerissen und bereits gelernt, dass *RSA* im Prinzip auf dem ***Fundamentalsatz der Arithmetik*** basiert.

Jede natürliche Zahl lässt sich als Produkt von endlich vielen Primzahlen darstellen. Ordnet man die Faktoren der Größe nach, ist die Darstellung eindeutig.

Große Zahlen sind leicht miteinander zu multiplizieren, aber schwer in die Primzahlfaktoren zu zerlegen – RSA macht sich dies zu nutze.

Geschichtlicher Hintergrund *

Whitfield „Whit“ Diffie und Martin Hellmann, zwei Kryptologen allererster Klasse, veröffentlichten 1976 ihre ***Theorie zur Public-Key- Kryptografie*** (*asymmetrische Verschlüsselung*). Im Kern geht es eben genau darum, dass man zum Ver- und Entschlüsseln ein Schlüsselpaar verwendet, wobei die beiden Schlüssel wechselseitig nicht auseinander berechnet werden können. Drei renommierte Mathematiker am MIT hatten starke Zweifel an dieser Theorie und versuchten, diese zu widerlegen. Bei diversen Verfahren gelang dies, allerdings stießen sie bei ihrer Suche auf eine Methode, die allen Angriffsversuchen widerstand.

Es handelte sich um die Mathematiker **R**ivest, **S**hamir und **A**dleman. Die Anfangsbuchstaben ihrer Namen bilden den Namen dieses bekannten asymmetrischen Verschlüsselungsverfahrens – RSA. Am 21. September 2000 erlosch das Patent. Das Verfahren ist seitdem frei nutzbar.

Öffentlicher und Privater Schlüssel *

Der **Öffentlicher und der Privater Schlüssel** sind natürlich elementar für das

Verfahren. Wie werden die beiden Schlüssel erzeugt?

1. Zunächst wählt man ZWEI Primzahlen, die in der Praxis möglichst groß sein sollen und unterschiedlich sein müssen. Die beiden Primzahlen werden gemeinhin mit p und q bezeichnet.
2. Diese beiden Primzahlen werden miteinander multipliziert und man erhält den s.g. **RSA-Modul**, meist mit N bezeichnet.
   N = q * q

3. Jetzt wird es leider ein wenig mathematisch, aber keine Angst. Spätestens beim Beispiel wird alles klar. Man muss für dieses N jetzt die s.g. Eulersche Phi-Funktion berechnen, wobei
   Phi(N) = (q-1) * (p -1)
4. Jetzt müssen wir eine zu Phi(N) teilerfremde Zahl e – den Verschlüsselungsexponenten – suchen, die größer als 1 und kleiner als Phi(N) selbst sein muss. Als Formel ausgedrückt:
   *1 < e < Phi(N)***e** wird meist recht klein gewählt. Üblich ist der Wert **65537** (2^18 +1).**e und N** bilden zusammen den **Öffentlichen Schlüssel**, den Sie, wie der Name bereits vermittelt, nicht geheim halten müssen.

5. Jetzt wird es doch etwas heikel, was die Mathematik angeht. Um letztlich unseren Privaten Schlüssel zu erzeugen, ist weitergehendes Know-How nötig. Man benötigt den Entschlüsselungsexponenten d (*d für decryption*).Lassen Sie die folgende Formel zunächst einfach einmal so stehen:**e * d** + k * **Phi(N)** = ggT(e,Phi(N))Wir kennen **e**, welches wir entsprechend der Anweisung oben gewählt haben, und wir kennen **Phi(N)**. Was wir nicht kennen, sind **d** (*unser gesuchter Entschlüsselungsexponent*) und den Wert **k**.

   Anm:
   ggT (*gelegentlich auch gcd*) steht für größter gemeinsamer Teiler.

   Um an dieser Stelle weiter zu kommen, brauchen wir den erweiterten Euklidschen Algorithmus. Mit diesem können wir dann **d** und k bestimmen, wobei k nicht weiter benötigt wird.

   Auf der nachfolgenden Seite können Sie sich die Werte berechnen lassen:
   http://www.saar.de/~awa/infsek1/Berlekamp.htm

   Die Bezeichnungen sind etwas anders. a entspricht unserem e, b unserem Phi(N), a‘ wäre das gesuchte d, b‘ das nicht weiter benötigte k.

   d und N bilden zusammen unseren **Privaten Schlüssel**.

Schlüsselerzeugung am Beispiel *

Wir arbeiten einfach die obige Liste ab.

1. Wir suchen uns also **zwei Primzahlen p und q**. Für p nehmen wir 13 und für q nehmen wir 17.
2. Wir rechnen **N** aus und erhalten.
   N = 13 * 17 = 221
3. Wir berechnen **Phi(N) = (p-1)*(q-1)** = (13-1) * (17-1) = 12 * 16 = 192
4. Jetzt wählen wir ein zu 192 teilerfremdes e. Da bietet sich doch eine 11 an. Denn 192 und 11 haben als einzigen gemeinsamen Teiler die 1.
5. Mit unserem Wert für e und Phi(N) gehen wir in den erweiterten Euklidschen Algorithmus.
   ggT(e,Phi(N))= e * d + k * Phi(N)
   Bekannte Werte einsetzen:
   ggT(11,192) = 11 * d + k * 192
   Wir lassen uns hier die Werte berechnen
   http://www.saar.de/~awa/infsek1/Berlekamp.htm

   (*setzen Sie für a unser e und für b unser Phi(N) ein*)
   und erhalten:
   11 * 35 + (-2) * 192 = 1
   Das gesuchte d ist also 35, den Wert für k brauchen wir nicht,

Damit hätten wir jetzt die beiden Schlüssel.

Der **Öffentliche Schlüssel** besteht aus **e und N**, also aus **11 und 221**,

der **Private Schlüssel** aus **d und N**, also aus **35 und 221**.

Damit haben wir bereits ein großes Stück arbeitet geleistet. Mit diesem Vorwiesen widmen wir uns im nächsten Teil dann der eigentlichen Ver- und Entschlüsselung von Daten, die natürlich die oben berechneten Werte ausgiebig nutzt.

