Matheschülerzirkel
Universität Augsburg
Schuljahr 2014/2015
Klasse 6/7

Zirkelzettel vom 15. April 2015

Die *RSA-Verschlüsselung* ist ein Beispiel für ein *asymmetrisches Verschlüsselungsverfahren*: Während bei sogenannten *symmetrischen Verschlüsselungsverfahren* derselbe Schlüssel für das Ver- und Entschlüsseln verwendet wird, gibt es bei den asymmetrischen Verschlüsselungsverfahren zwei verschiedene Schlüssel: Einen *öffentlichen Schlüssel*, mit dem eine Nachricht verschlüsselt werden kann, und einen *privaten Schlüssel*, mit dem die verschlüsselte Nachricht wieder entschlüsselt wird.

Um zu verstehen, wie das RSA-Verfahren funktioniert, brauchen wir ein paar Vorkenntnisse über die *Eulersche ϕ-Funktion* und über die *Modulo-Rechnung*.

Die Eulersche ϕ-Funktion ist folgendermaßen definiert: Für eine beliebige natürliche Zahl n ist $\phi(n)$ die Anzahl der Zahlen, die zwischen 1 und n liegen, und die teilerfremd zu n sind. Zwei Zahlen sind dabei *teilerfremd*, falls ihr größter gemeinsamer Teiler gleich 1 ist. Ein paar Beispiele: $\phi(1) = 1$, denn die einzige Zahl zwischen 1 und 1 ist 1 selbst, und $\mathrm{ggT}(1,1) = 1$, also ist 1 teilerfremd zu sich selbst. $\phi(2) = 1$, denn $\mathrm{ggT}(1,2) = 1$, aber $\mathrm{ggT}(2,2) = 2 \neq 1$, also ist nur eine der beiden Zahlen teilerfremd zu 2. Weiter ist $\phi(3) = 2$, denn sowohl 1 als auch 2 sind teilerfremd zu 3, allerdings ist 3 nicht teilerfremd zu sich selbst. Interessanter ist $\phi(4) = 2$, denn hier sind nur 1 und 3 teilerfremd zu 4, aber 2 und 4 nicht. Der Wert von $\phi(5)$ ist 4, da alle Zahlen, die kleiner als 5 sind, keinen gemeinsamen Teiler mit 5 außer 1 hat (5 ist ja eine Primzahl, also sind die einzigen Teiler von 5 die Zahl 5 selbst und 1). Genauso ist $\phi(p) = p - 1$ für jede beliebige Primzahl.

Natürlich ist $\phi(n)$ auch gleich n minus die Anzahl der Zahlen, die *nicht* teilerfremd zu n sind, die also Vielfache eines Teilers von n sind. Wollen wir zum Beispiel $\phi(20)$ berechnen, dann können wir einfach 20 in seine Primfaktoren zerlegen ($20 = 2^2 \cdot 5$), und alle Vielfachen der Primfaktoren zählen: Das sind 2, 4, 5, 6, 8, 10, 12, 14, 15, 16, 18 und 20, also 12 Stück. Das heißt, $\phi(20) = 20 - 12 = 8$.

Für die RSA-Verschlüsselung ist der Fall wichtig, dass n das Produkt von zwei verschiedenen Primzahlen ist, also $n = p \cdot q$, wobei p und q zwei verschiedene Primzahlen sind. In diesem Fall sind die Teiler von n ja nur p und q, und deren Vielfache sind einerseits

$$1 \cdot p, 2 \cdot p, \ldots, (q-1) \cdot p \text{ und } q \cdot p$$

sowie andererseits

$$1 \cdot q, 2 \cdot q, \ldots, (p-1) \cdot q \text{ und } p \cdot q.$$

Dabei haben wir nur $p \cdot q$ doppelt gezählt: Ist nämlich $a \cdot p = b \cdot q$, dann ist q ein Teiler von $a \cdot p$. Da q kein Teiler von p ist (p ist ja eine Primzahl und $q \neq p$), muss q ein Teiler von a sein, also $a \geq q$.

Das heißt, die Anzahl der zu n nicht teilerfremden Zahlen ist gerade $q + p - 1$ und damit

$$\phi(n) = \phi(pq) = pq - (q + p - 1) = pq - q - p + 1 = (p - 1) \cdot (q - 1).$$

Das zweite wichtige Thema, das für das Verständnis von RSA wichtig ist, ist die *Modulo-Rechnung*. Die Idee dabei ist sehr einfach: $a \bmod n$ ist die Zahl, die bei der Division von a durch n als Rest bleibt. Ein paar Beispiele:

$$\begin{aligned} 15 \bmod 7 &= 1 \\ 15 \bmod 8 &= 7 \\ (3 \cdot 5) \bmod 4 &= 3 \\ (2^4) \bmod 7 &= 2 \end{aligned}$$

Die Modulo-Rechnung wird sehr angenehm durch die Tatsache, dass man Rechenoperationen und Modulo in einem gewissen Sinne „vertauschen“ darf. Konkret gelten folgende Rechenregeln:

$$\begin{aligned} (a + b) \bmod n &= ((a \bmod n) + (b \bmod n)) \bmod n \\ (a \cdot b) \bmod n &= ((a \bmod n) \cdot (b \bmod n)) \bmod n \\ (a^b) \bmod n &= ((a \bmod n)^b) \bmod n \end{aligned}$$

Beispielsweise können wir jetzt $(3109^2 + 329 \cdot 17) \bmod 3$ sehr einfach ausrechnen, ohne den Term in Klammern explizit berechnen zu müssen:

$$\begin{aligned} &(3109^2 + 329 \cdot 17) \bmod 3 \\ &= ((3109^2 \bmod 3) + ((329 \cdot 17) \bmod 3)) \bmod 3 \\ &= (((3109 \bmod 3)^2 \bmod 3) + (((329 \bmod 3) \cdot (17 \bmod 3)) \bmod 3)) \bmod 3 \\ &= ((1^2 \bmod 3) + ((2 \cdot 2) \bmod 3)) \bmod 3 \\ &= (1 + 1) \bmod 3 = 2 \end{aligned}$$

Der sogenannte *Satz von Euler* ist eine „Rechenregel“, die die ϕ-Funktion mit der Modulo-Rechnung verbindet. Er sagt aus, dass

$$a^{\phi(n)} \bmod n = 1$$

ist, wenn a und n teilerfremd sind. Ein Beispiel dafür ist

$$16^{352} \bmod 391 = 16^{16 \cdot 22} \bmod 391 = 16^{\phi(17 \cdot 23)} \bmod 391 = 16^{\phi(391)} \bmod 391 = 1.$$

Jetzt sind wir in der Lage, das RSA-Verfahren zu verstehen. Man wählt zwei verschiedene Primzahlen p und q und berechnet $n = p \cdot q$. Wichtig ist: Die Sicherheit des Algorithmus hängt davon ab, dass es schwierig ist, p und q aus n zurückzugewinnen. Man sollte p und q also sehr groß wählen. Als nächstes berechnet man $\phi(n) = \phi(pq) = (p-1)(q-1)$ und wählt eine Zahl e, die teilerfremd zu $\phi(n)$ ist. Man kann (ein Beispiel folgt) eine Zahl d mit $(d \cdot e) \bmod 336 = 1$ finden. Die Zahlen e und n werden veröffentlicht, und d (und $\phi(n)$) bleiben geheim.

Verschlüsselt werden können Zahlen, die kleiner als n sind. Dabei wird zu einer Zahl $m < n$ der Geheimtext durch die Formel

$$c = m^e \bmod n$$

berechnet. Die Entschlüsselung funktioniert ähnlich mit Hilfe des geheimen Schlüssels d, nämlich durch die Formel

$$m' = c^d \bmod n.$$

Natürlich muss man sich jetzt noch überlegen, ob das Verfahren korrekt arbeitet, ob also bei der Entschlüsselung wieder die ursprüngliche Nachricht herauskommt. In unseren Symbolen bedeutet das $m = m'$. Das ist tatsächlich der Fall:

$$\begin{aligned}
m' &= c^d \bmod n \\
&= (m^e \bmod n)^d \bmod n \\
&= ((m^e)^d) \bmod n \\
&= (m^{ed}) \bmod n \\
&= (m^{1+k\phi(n)}) \bmod n \\
&= (m \cdot (m^{\phi(n)})^k) \bmod n \\
&= ((m \bmod n) \cdot ((m^{\phi(n)} \bmod n)^k \bmod n)) \bmod n \\
&= ((m \bmod n) \cdot (1^k \bmod n)) \bmod n \\
&= ((m \bmod n) \cdot 1) \bmod n \\
&= (m \bmod n) \bmod n = m
\end{aligned}$$

In der fünften Zeile haben wir dabei verwendet, dass $ed \bmod \phi(n) = 1$ ja gerade bedeutet, dass $ed = 1 + k\phi(n)$ für eine natürliche Zahl k ist. Jetzt haben wir verstanden, warum der RSA-Algorithmus das richtige Ergebnis liefert. Jetzt noch ein konkretes Beispiel:

Wir nehmen als Primzahlen $p = 13$ und $q = 29$ (das ist kein sehr realistisches Beispiel, weil p und q nicht besonders groß ist). Dann ist $n = pq = 13 \cdot 29 = 377$, $\phi(n) = (p-1)(q-1) = 12 \cdot 28 = 336$. Wir nehmen $e = 17$ und überlegen uns, dass tatsächlich $\mathrm{ggT}(e, \phi(n)) = 1$ ist. Das kann man mit Hilfe des *euklidischen Algorithmus* überprüfen.

Dazu rechnet man

$$\begin{aligned} 336 &= 19 \cdot 17 + 13 \\ 17 &= 1 \cdot 13 + 4 \\ 13 &= 3 \cdot 4 + 1 \\ 4 &= 4 \cdot 1 \end{aligned}$$

Der größte gemeinsame Teiler muss auf jeden Fall die Zahlen teilen, die ganz rechts stehen (also $13, 4, 1$) – wenn zum Beispiel eine Zahl a die Zahlen 17 und 336 teilt, dann muss sie auch $13 = 336 - 19 \cdot 17$ teilen und so weiter. Also teilt ggT$(336, 17$ die 1, muss also selbst schon gleich 1 sein. Die Rechnung aus dem euklidischen Algorithmus kann man jetzt verwenden, um eine Zahl d mit $d \cdot e$ mod $336 = 1$ zu finden. Wir schreiben dazu die 1 aus der vorletzten Zeile etwas komplizierter als

$$\begin{aligned} 1 &= 13 - 3 \cdot 4 = 13 - 3 \cdot (17 - 1 \cdot 13) \\ &= 13 + 3 \cdot 13 - 3 \cdot 17 = 4 \cdot 13 - 3 \cdot 17 \\ &= 4 \cdot (336 - 19 \cdot 17) - 3 \cdot 17 = 4 \cdot 336 - 4 \cdot 19 \cdot 17 - 3 \cdot 17 \\ &= 4 \cdot 336 - (4 \cdot 19 + 3) \cdot 17 = 4 \cdot 336 - 79 \cdot 17 \\ &= 4 \cdot 336 - (336 - 257) \cdot 17 = (4 - 17) \cdot 336 + 257 \cdot 17 \\ &= -13 \cdot 336 + 257 \cdot 17. \end{aligned}$$

Daher ist $(257 \cdot 17)$ mod $336 = (1 + 13 \cdot 336)$ mod $336 = 1$ und wir können $d = 257$ nehmen. Angenommen, wir wollen jetzt die Zahl $m = 28 < 377$ verschlüsseln. Dann erhalten wir als Geheimtext

$$\begin{aligned} c &= m^e \text{ mod } 377 = 28^{17} \text{ mod } 377 \\ &= (28 \cdot (28^2)^8) \text{ mod } 377 \\ &= (28 \cdot (28^2 \text{ mod } 377)^8) \text{ mod } 377 \\ &= (28 \cdot (784 \text{ mod } 377)^8) \text{ mod } 377 \\ &= (28 \cdot 30^8) \text{ mod } 377 \\ &= (28 \cdot (30^2 \text{ mod } 377)^4) \text{ mod } 377 \\ &= (28 \cdot 146^4) \text{ mod } 377 \\ &= (28 \cdot (146^2 \text{ mod } 377)^2) \text{ mod } 377 \\ &= (28 \cdot 204^2) \text{ mod } 377 \\ &= 318 \end{aligned}$$

Die Zahl 28 wird also durch die Zahl 318 verschlüsselt. Auch die umgekehrte Rechnung funktioniert mit Hilfe eines Computers:

$$m = c^d \text{ mod } 377 = 318^{257} \text{ mod } 377 = \cdots = 28.$$

An diesen ganzen Rechnungen sieht man, dass RSA-Verschlüsselung einen höheren Rechenaufwand erfordert als die symmetrischen Verschlüsselungsverfahren, die wir bisher besprochen haben. Natürlich kann ein Computer diese ganzen Rechnungen in sehr schneller Zeit durchführen, aber das Verfahren ist trotzdem langsamer als unsere symmetrischen Verfahren. Daher wird RSA üblicherweise benutzt, um einen geheimen Schlüssel auszutauschen, und die tatsächliche Nachricht wird dann mit einem anderen Verfahren verschlüsselt.