Abschnitt: CRT-RSA

Erinnerung:

- Ist (N, e) ein öffentlicher RSA-Schlüssel, so wandelt man den Klartext (nach einem vereinbarten Verfahren) in eine Folge von Zahlen a aus $\{0, 1, \ldots, N-1\}$ um und berechnet

$$b = a^e \bmod N.$$

  Der Chiffretext besteht aus den Zahlen b.
- Ist (N, d) der zu (N, e) gehörige private Schlüssel, so erhält man die Klartextzahlen a durch Berechnung von

$$a = b^d \bmod N.$$

- Es spricht nichts dagegen, e relativ klein zu wählen. Dann wird aber d relativ groß sein, denn es gilt der Zusammenhang

$$ed \equiv 1 \bmod (p-1)(q-1) \quad \text{mit} \quad N = pq.$$

- Eine Möglichkeit, mit stellenweise kleineren Zahlen bei der Entschlüsselung auszukommen, bietet der chinesische Restsatz. Wir stellen eine Möglichkeit vor. (CRT wie Chinese Remainder Theorem.)

Lemma

Seien p, q verschiedene ungerade Primzahlen, $N = pq$ und $e \in \mathbb{N}_{>1}$ mit $\mathrm{ggT}(e, (p-1)(q-1)) = 1$.

- *e ist invertierbar modulo $p-1$ und modulo $q-1$. Seien d_p und d_q entsprechende Inversen, d.h.*

$$ed_p \equiv 1 \bmod (p-1) \quad \text{und} \quad ed_q \equiv 1 \bmod (q-1).$$

- *Wegen $p \neq q$ ist p invertierbar modulo q. Sei p_{inv} ein Inverses von p modulo q, d.h.*

$$p_{\mathrm{inv}} \cdot p \equiv 1 \bmod q.$$

- *Ist $a \in \{0, 1, \ldots, N-1\}$ und $b = a^e \bmod N$, berechnet man*

$$a_p = b^{d_p} \bmod p \quad \text{und} \quad a_q = b^{d_q} \bmod q,$$

so gilt

$$a = a_p + p \cdot \Big((p_{\mathrm{inv}} \cdot (a_q - a_p)) \bmod q \Big).$$

(Beachte: $(p_{\mathrm{inv}} \cdot (a_q - a_p)) \bmod q$ ist eine Zahl aus

Beweis:

- Aus $\mathrm{ggT}(e, (p-1)(q-1)) = 1$ folgt natürlich $\mathrm{ggT}(e, p-1) = \mathrm{ggT}(e, q-1) = 1$ und damit die Invertierbarkeit von e modulo $p-1$ und modulo $q-1$. Aus $p \neq q$ folgt $\mathrm{ggT}(p, q) = 1$, also ist p invertierbar modulo q. Dies rechtfertigt die eingeführten Bezeichnungen d_p, d_q und p_{inv}.
- *Behauptung:* $a_p \equiv a \bmod p$.
  Wegen $ed_p \equiv 1 \bmod p$ gibt es ein $k \in \mathbb{N}_0$ mit $ed_p = 1 + k_p(p-1)$. Es folgt modulo p

$$a_p \equiv b^{d_p} \equiv (a^e)^{d_p} \equiv a^{ed_p} \equiv a^{1+k_p(p-1)} = a \cdot (a^{p-1})^{k_p} \equiv a \bmod p,$$

  wobei die letzte Gleichung für $a \equiv 0 \bmod p$ trivial ist, für $a \not\equiv 0 \bmod p$ aus dem kleinen Satz von Fermat ($a^{p-1} \equiv 1 \bmod p$) folgt. Damit ist die Behauptung bewiesen.
- *Behauptung:* $a_q \equiv a \bmod q$.
  Dies folgt mit dem gleichen Argument wie die Gleichung für p.
- Sei

$$\tilde{a} = a_p + p \cdot \Big((p_{\mathrm{inv}} \cdot (a_q - a_p)) \bmod q \Big).$$

  Wir müssen zeigen, dass die Gleichheit $\tilde{a} = a$ (in $\mathbb{Z}$) gilt.

- Aus $a_p \in \{0, 1, \ldots, p-1\}$ und $(p_{\text{inv}} \cdot (a_q - a_p)) \bmod q \in \{0, 1, \ldots, q-1\}$ folgt

$$\begin{aligned} 0 \leq \tilde{a} &= a_p + p \cdot \Big((p_{\text{inv}} \cdot (a_q - a_p)) \bmod q \Big) \leq \\ &\leq (p-1) + p \cdot (q-1) = pq - 1 = N - 1, \end{aligned}$$

  also $\tilde{a} \in \{0, 1, \ldots, N-1\}$.
- Modulo p gilt $\tilde{a} \equiv a_p \equiv a \bmod p$, wobei wir $a_p \equiv a \bmod p$ benutzt haben.
- Es gibt $\lambda, \mu \in \mathbb{Z}$ mit

$$(p_{\text{inv}} \cdot (a_q - a_p)) \bmod q = p_{\text{inv}} \cdot (a_q - a_p) + \lambda q \quad \text{und}$$

$$p \cdot p_{\text{inv}} = 1 + \mu q.$$

  Damit ergibt sich

$$\begin{aligned} \tilde{a} &= a_p + p \cdot \Big(p_{\text{inv}} \cdot (a_q - a_p) + \lambda q \Big) = a_p + (1 + \mu q) \cdot (a_q - a_p) + \lambda pq = \\ &= a_q + q \cdot (\mu(a_q - a_p) + \lambda p) \end{aligned}$$

  Es folgt mit $a_q \equiv a \bmod q$ dann

$$\tilde{a} \equiv a_q \equiv a \bmod q.$$

- Aus $\tilde{a} \equiv a \bmod p$ und $\tilde{a} \equiv a \bmod q$ folgt $\tilde{a} \equiv a \bmod N$. Wegen

Definition

Ist (N, e) ein öffentlicher RSA-Schlüssel mit $N = pq$, ist d_p ein Inverses von e modulo $p-1$, d_q ein Inverses von e modulo $q-1$ und p_{inv} ein Inverses von p modulo q, also

$$ed_p \equiv 1 \bmod (p-1), \quad ed_q \equiv 1 \bmod (q-1), \quad p \cdot p_{\text{inv}} \equiv 1 \bmod q,$$

so nennt man das Tupel

$$(p, q, d_p, d_q, p_{\text{inv}})$$

den zu (N, e) gehörigen privaten CRT-RSA-Schlüssel.

Entschlüsselung mit einem CRT-RSA-Schlüssel: Wurde $a \in \{0, 1 \ldots, N-1\}$ mit dem RSA-Schlüssel (N, e) zu $b = a^e \bmod N$ verschlüsselt, so erhält man a zurück aus b mit dem zugehörigen privaten CRT-RSA-Schlüssel $(p, q, d_p, d_q, p_{\text{inv}})$ durch die Gleichungen

$$a_p = b^{d_p} \bmod p, \quad a_q = b^{d_q} \bmod q,$$

$$a = a_p + p \cdot \Big((p_{\text{inv}} \cdot (a_q - a_p)) \bmod q \Big).$$

Bemerkungen:

- Mit Python kann die Entschlüsselung so aussehen:

```
a_p=pow(b,d_p,p)
a_q=pow(b,d_q,q)
a=a_p+p*((p_inv*(a_q-a_p))%q)
```

- Da man p_{inv} (einfach) aus p und q berechnen kann, müsste man es natürlich nicht in den Schlüssel aufnehmen.
- Im BSI-Dokument TR-02102-1 (vom 2.2.2024) wird das Tupel $(n, d, p, q, d_p, d_q, q_{\text{inv}})$ als privater CRT-RSA-Schlüssel bezeichnet (S.35).
- Die hier benutzte Variante des chinesischen Restsatzes findet sich als ein Satz in Abschnitt 15 (Varianten des chinesischen Restsatzes und eine Anwendung) im Kapitel „Grundeigenschaften der Ringe $\mathbb{Z}$ und $\mathbb{Z}/n\mathbb{Z}$“ des Vorlesungsskripts. Im „Handbook of Applied Cryptography“ von Menezes, van Oorschot und Vanstone ist das Verfahren unter dem Namen Garner-Algorithmus zu finden.

Beispiel: Das Beispiel soll nur einen optischen Eindruck von der Größe der Zahlen geben. $e = 65537$ ist klein. N und d haben jeweils 160 Dezimalstellen. Dagegen haben die Zahlen $p, q, d_p, d_q, p_{\mathrm{inv}}$ nur 80 oder 79 Dezimalstellen.

$$
\begin{aligned}
N &= 311884423959661626331706085618488835004395939956584679806460852039977050532040186605979970840196358722057502299012470788257395814235047763051991217629216542170765537 \\
e &= 65537 \\
d &= 266203685055091811552908662546760279727114455792780167218728460890981220572520306532340228402724863173630933734923209563749603013001776431165417677555943350481765323402284027248631736309337349232095637496030130017764311654176775559433504817 \\
p &= 49623337115637218710663148740331092793884556519090452761775654307780041870554613 \\
q &= 62850352694515007782308391547025712455815841148754944406482115778100354149248639 \\
d_p &= 56621956291977832601177811965789693137108612486039703641081883960751812427790001 \\
d_q &= 41780998761645106490254813228089006432435551693361103849080768697140518016087331 \\
p_{\mathrm{inv}} &= 19721418075718714147457445760104110794805740784757635939268653253818102587155 80
\end{aligned}
$$