FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Fachbereich Elektrotechnik
und Informationstechnik

Prof. Georg Hoever

WS 2024/25
20.11.2024

Musterlösung zum 8. Übungsblatt zur Vorlesung Kryptologie

Übung 1

Mit $p = 31$ und $q = 17$ ergibt sich

$$n = p \cdot q = 527,$$
$$f = (p-1) \cdot (q-1) = 480.$$

480 hat als Primfaktoren die Zahlen 2, 3 und 5, wähle deshalb $e = 7$. Mit dem euklidischen Algorithmus kann man nun d berechnen:

k	r_k	q_k	x_k	
0	480		0	
1	7		1	+
2	4	68	68	−
3	3	1	69	+
4	1	1	137	−

$$\Rightarrow d = -\frac{1}{1} \cdot 137 \bmod 480 = 343.$$

e und n sind der öffentliche Teil des Schlüssels, d der private.

Übung 2

a) Mit $p = 29$ und $q = 31$ ergibt sich

$$f = (p-1) \cdot (q-1) = 840.$$

Damit kann man nun mit dem euklidischen Algorithmus d bestimmen:

k	r_k	q_k	x_k	
0	840		0	
1	23		1	+
2	12	36	36	−
3	11	1	37	+
4	1	1	73	−

$$\Rightarrow d = -\frac{1}{1} \cdot 73 \bmod 840 = 767.$$

b) Verschlüsselung : $c = m^e \bmod n$, $n = 29 \cdot 31 = 899$

$$\Rightarrow \; c \;=\; 6^{23} \bmod 899 \;=\; 863.$$

c) Entschlüsselung : $m = c^d \bmod n$

$$m \;=\; 863^{767} \bmod 899 \;=\; 6.$$

d) $c \;=\; 58^{23} \bmod 899 \;=\; 29, \qquad m \;=\; 29^{767} \bmod 899 \;=\; 58.$

e) Die Ver- und Entschlüsselung funktioniert nicht, da m größer ist als n.

Übung 3

Die Verschlüsselung ist die e-fache Addition von m in $\mathbb{Z}_{100}$, also modulo 100:

$$c \;=\; \underbrace{m + \ldots + m}_{e\text{-mal}} \bmod 100 \;=\; e \cdot m \bmod 100,$$

konkret: $c = 11 \cdot 31 \bmod 100 = 41$.

Für d muss gelten $ed = 1 \bmod 100$, d.h., d ist das multiplikative Inverse zu e modulo 100, konkret $11^{-1} = 91 \bmod 100$.

Die Entschlüsselung ist dann

$$m \;=\; \underbrace{c + \ldots + c}_{d\text{-mal}} \bmod 100 \;=\; d \cdot c \bmod 100,$$

konkret: $m = 91 \cdot 41 \bmod 100 = 31$.

(Es ist allgemein $d \cdot c = d \cdot (e \cdot m) = (d \cdot e) \cdot m = 1 \cdot m = 1 \bmod 100$.)

Übung 4

a) Der Entschlüsselungsexponent d wird mit dem Modul $\Phi(n) = (p-1)(q-1)(r-1)$ berechnet:

$$d \cdot e \;=\; 1 \bmod \Phi(n)$$

b) Bei gleicher Stellenanzahl von n sind bei RSA^3 die Primzahlen kleiner als beim normalen RSA, was das Faktorisieren (z.B. durch Probedivision) beschleunigt. Man muss nun zwar zwei Primfaktoren bestimmen, um n vollständig faktorisieren und damit $\Phi(n)$ bestimmen zu können, aber bei Annahme eines exponentiellen Aufwands zur Stellenanzahl erhält man beim üblichen RSA-Verfahren mit s-stelligem n, also p, q, die etwa $\frac{s}{2}$ Stellen haben, einen Aufwand von $\exp\left(\frac{s}{2}\right)$ im Vergleich zu $2 \cdot \exp\left(\frac{s}{3}\right)$ bei RSA^3.