

## Praktikumsversuch 1: RSA-Verschlüsselung

Hardwarenahe Programmierung / Angewandte Informatik  
Wintersemester 2017/18 · Prof. Dr. Peter Gerwinski

Aufgabe: Schreiben Sie ein Programm, das die Verschlüsselung nach Rivest, Shamir und Adleman (RSA) sowie die Schwierigkeiten beim Brechen („Knacken“) der Verschlüsselung demonstriert.

Schreiben Sie ein C-Programm (oder mehrere), das folgendes durchführt:

- **Schlüsselerzeugung**

Bestimmen Sie 3 Primzahlen  $p$ ,  $q$  und  $e$ , wobei  $e$  kleiner als  $(p-1) \cdot (q-1)$  und teilerfremd zu  $(p-1) \cdot (q-1)$  sei. (Dies ist z. B. der Fall, wenn  $e$  größer als  $p$  und  $q$  ist.)

Berechnen Sie  $N = p \cdot q$  sowie eine natürliche Zahl  $d$  mit der Eigenschaft:

$$(e \cdot d) \% ((p-1) \cdot (q-1)) = 1$$

(„ $x \% y$ “ wird „ $x$  modulo  $y$ “ gesprochen und steht für den Rest, der bei Division von  $x$  durch  $y$  verbleibt.

$N$  und  $e$  sind der *öffentliche Schlüssel*.  
 $p$ ,  $q$  und  $d$  sind der *geheime Schlüssel*.

- **Verschlüsselung**

Wählen Sie eine geheime Botschaft  $m$  (eine Zahl), die Sie verschlüsseln wollen.

$m$  muß teilerfremd zu  $N$  sein. (Dies ist z. B. der Fall, wenn  $m$  kleiner ist als  $N$  und es nicht gleich  $p$  oder gleich  $q$  ist.)

Schreiben Sie ein Programm, das aus  $m$  die verschlüsselte Nachricht  $c$  berechnet:

$$c = m^e \% N$$

Hinweis:

$$\begin{aligned} m^e \% N &= \underbrace{(m \cdot m \cdot \dots \cdot m)}_{e \text{ Faktoren}} \% N \\ &= \underbrace{\left( \dots \left( (m \cdot m) \% N \cdot m \right) \% N \cdot \dots \cdot m \right)}_{e \text{ Faktoren}} \% N \end{aligned}$$

Dies bedeutet: Multiplizieren Sie die Zahl  $m$   $e$ -mal mit sich selbst, wobei Sie *nach jeder Multiplikation* modulo  $N$  rechnen.

- **Entschlüsselung**

Rekonstruieren Sie aus der verschlüsselten Botschaft  $c$  wieder die geheime Botschaft  $m$ :

$$m = c^d \% N$$

- **Verschlüsselung brechen**

Rekonstruieren Sie aus der verschlüsselten Botschaft  $c$  wieder die geheime Botschaft  $m$ , *ohne* den geheimen Schlüssel zu kennen, d. h. Sie kennen nur  $N$  und  $e$ , nicht jedoch  $p$ ,  $q$  und  $d$ .

Hinweis:

Sie können z. B. versuchen  $N$  in seine Primfaktoren zu zerlegen. Auf diese Weise können Sie zunächst  $p$  und  $q$  berechnen und danach  $d$ .

Wenn Sie die Primzahlen groß genug wählen, sollte man unmittelbar erkennen, daß das Brechen der Verschlüsselung *wesentlich* länger dauert als das reguläre Ver- und Entschlüsseln. Auf diesem Schwierigkeitsunterschied beruht die Sicherheit der RSA-Verschlüsselung.

*Viel Erfolg!*

Stand: 19. Oktober 2016

Copyright © 2014, 2015, 2016, 2017 Peter Gerwinski  
Lizenz: CC-by-sa (Version 3.0) oder GNU GPL (Version 3 oder höher)

Sie können diese Praktikumsunterlagen einschließlich Quelltext herunterladen unter:  
<https://gitlab.cvh-server.de/pgerwinski/hp>