

## Übungen zur Vorlesung „Betriebssysteme“

### Blatt 9

#### Aufgabe 35

Bei dem in der Vorlesung vorgestellten RSA-Verschlüsselungsverfahren werden der öffentliche Schlüssel (*public key*) und der geheime Schlüssel (*secret key*) nach dem folgenden Algorithmus bestimmt.

1. Wähle große ( $> 100$  Stellen) Primzahlen  $p, q$ .
2.  $n := p \cdot q$ .
3. Wähle  $e < n$  mit  $\text{ggT}(e, \varphi(n)) = 1$ . (\*  $\varphi(n) = (p-1)(q-1)$  \*)
4. Berechne  $d \in \mathbb{N}$  mit  $e \cdot d = 1 \pmod{\varphi(n)}$ . (\* Erweiterter Euklidischer Algorithmus \*)
5. public key :=  $(e, n)$ ; secret key :=  $(d, n)$ .
  - a) Führen Sie die Berechnung eines öffentlichen und eines entsprechenden geheimen Schlüssels für die Wahl  $p = 127, q = 139$  von Hand durch.
  - b) Verschlüsseln Sie eine beliebige (Teil-)Nachricht, und prüfen Sie durch Entschlüsselung Ihres chiffrierten Textes, ob Sie korrekt gearbeitet haben. Benutzen Sie den in der Vorlesung vorgestellten Algorithmus zur modularen Exponentiation.
  - c) Die Sicherheit des RSA-Verfahrens basiert unter anderem darauf, daß es bis heute unbekannt ist, wie man  $\sqrt[n]{c} \pmod{n}$  berechnen kann. Gibt es Fälle (etwa durch die Wahl von  $p, q, e$  oder die Art der Zerlegung der zu verschlüsselnden Nachricht in Blöcke), in denen die Entschlüsselung bei Kenntnis des öffentlichen Schlüssels „leicht“ ist?
  - d)\* Beweisen Sie, daß man einen Block  $b$  einer Nachricht, der mittels  $c = b^e \pmod{n}$  verschlüsselt wird, durch  $b = c^d \pmod{n}$  entschlüsseln kann.  
Hinweis: Zeigen Sie  $\forall b < n : b^{ed} \equiv b \pmod{n}$ .

#### Aufgabe 36

Der Aufwand zur Entschlüsselung einer mit dem RSA-Verfahren chiffrierten Nachricht ohne Kenntnis des geheimen Schlüssels entspricht dem Aufwand zur Faktorisierung natürlicher Zahlen. Eine naheliegende Idee besteht darin, die zu faktorisierende Zahl  $M$  nacheinander durch die aufeinanderfolgenden Primzahlen  $2, 3, 5, 7, 11, \dots$  zu dividieren bis die kleinste Primzahl  $p$  gefunden wird, die Teiler von  $M$  ist. Dann wird dieses Verfahren rekursiv auf  $M/p$  angewendet. Somit erhält man nacheinander die Primfaktoren  $p_1 \leq p_2 \leq \dots \leq p_t$ .

- a) Formulieren Sie einen Algorithmus, der diese Idee umsetzt. Achten Sie dabei insbesondere auf geeignete Abbruchbedingungen.
- b) Führen Sie Ihren Algorithmus von Hand für die Zahlen 510510, 425253, 99983 durch.
- c) \* Wieviele Divisionen benötigt der Algorithmus? Geben Sie alternative (effizientere) Faktorisierungsalgorithmen an.

#### Aufgabe 37

- a) Beschreiben Sie die einzelnen Schritte, die bei einem entfernten Unterprogrammaufruf (*RPC, Remote Procedure Call*) ausgeführt werden. Erklären Sie insbesondere die Begriffe *Client-Stub* und *Server-Stub*.
- b) Was versteht man unter *dynamischem Binden*?

### Aufgabe 38

In RPC-Systemen können folgende Arten von Fehlern auftreten:

- Der Client kann den Server nicht lokalisieren.
- Die Anfragenachricht vom Client an den Server geht verloren.
- Die Antwortnachricht vom Server an den Client geht verloren.
- Der Server fällt nach dem Empfang einer Nachricht aus.
- Der Client fällt nach dem Senden einer Nachricht aus.

Beschreiben Sie für diese Fehlerarten jeweils, wodurch Sie ausgelöst werden könnten, und geben Sie geeignete Methoden zur Behebung der Fehler an.