

# Das RSA-Verfahren

9. März 2023

## Inhaltsverzeichnis

<b>1</b>	<b>Asymmetrische Verschlüsselung</b>	<b>2</b>
1.1	Das Problem der symmetrischen Verschlüsselung . . . . .	2
1.2	Was bedeutet asymmetrisch? - Die Idee des RSA-Verfahrens . . . . .	2
1.3	Zusatz: Erste Versuche . . . . .	4
1.3.1	Versuch 1: Modulare Addition . . . . .	4
1.3.2	Versuch 2: Modulare Multiplikation . . . . .	4
1.3.3	Versuch 3: Modulares Potenzieren . . . . .	4
<b>2</b>	<b>Ver-und Entschlüsseln von Nachrichten</b>	<b>5</b>
<b>3</b>	<b>Herstellung eines Schlüsselpaars</b>	<b>5</b>
<b>4</b>	<b>Angriffsmöglichkeiten</b>	<b>6</b>
4.1	Primfaktorzerlegung von $n$ . . . . .	6
4.2	Digitale Signatur . . . . .	7

# 1 Asymmetrische Verschlüsselung

## 1.1 Das Problem der symmetrischen Verschlüsselung

Alice und Bob arbeiten an einem streng geheimen Projekt. Da beide in verschiedenen Städten wohnen, sind sie sich noch nie begegnet. Alice und Bob müssen nun aber wichtige Nachrichten austauschen. Beide haben Angst, dass Oscar versuchen könnte ihre Nachrichten abzufangen. Sie müssen deshalb verschlüsseln.

Leider hat Alice nicht ein einziges Mal Zeit, sich persönlich mit Bob zu treffen. Aus diesem Grund sendet sie alle ihre Nachrichten mit der Post. Eine symmetrische Verschlüsselung der Nachrichten ist dabei keine gute Idee. Warum?

- ☐ Alice müsste Bob einmal den geheimen Schlüssel des verwendeten Verfahren mitteilen. Wird der Postbote dabei von Oscar abgefangen, kann Oscar alle weiteren Nachrichten entschlüsseln.
- ☐ Alle symmetrischen Verfahren sind einfach zu knacken.
- ☐ Da sich symmetrische Verfahren nur für sehr kurze Nachrichten eignen, müssen die beiden viel zu viele Nachrichten hin und her schicken.

Aus diesem Grund verwendet Alice eine asymmetrische Verschlüsselung, das RSA-Verfahren. Das RSA-Verfahren wird in vielen Bereichen des alltäglichen Lebens verwendet. Davon wissen aber nur die wenigsten. Es kommt unter Anderem zum Einsatz bei

- Apps zum Chatten (Telegram, Threema, usw.),
- jedem Öffnen einer sicheren Website (<https://...>) im Internet,
- Bankgeschäften oder
- der Fernwartung von Computern.

Im letzten Kapitel des Skripts wird genauer auf die Einsatzgebiete eingegangen.

## 1.2 Was bedeutet asymmetrisch? - Die Idee des RSA-Verfahrens

In diesem Kapitel wollen wir verstehen, warum das RSA-Verfahren als asymmetrisch bezeichnet wird. Dazu reicht es zunächst aus, wenn wir uns nur den Ablauf des Nachrichtenaustauschs anschauen. Öffne dazu die Website

<https://mx3030.github.io/rsa/>



oder verwende den QR-Code.

Bei den symmetrischen Verfahren mussten beide Parteien alle Informationen (den geheimen Schlüssel) kennen. RSA ist nun ein asymmetrisches Verfahren, weil Alice über Informationen verfügt, die Bob nicht kennt und umgekehrt. Bei diesen geheimen Informationen handelt es sich um **private** Schlüssel, die zum Entschlüsseln der erhaltenen Nachrichten verwendet werden.


Wir wollen uns jetzt anschauen, wie Alice eine Nachricht an Bob sendet.

## Alice sendet eine Nachricht an Bob

Drücke auf den Button von **Bob** um die Perspektive von Bob einzunehmen.

**Bob** ↻


öffentlicher Schlüssel



n= 33

e= 17

privater Schlüssel



n= 33

d= 13

Durch Drücken von **Bob** baut sich Bob ein Schlüsselpaar (Kapitel 3) bestehend aus einem **öffentlichen** und einem **privaten** Schlüssel.


Der **öffentliche** Schlüssel besteht aus zwei Zahlen  $(n, e)$  und ist frei zugänglich für alle Personen.

Der **private** Schlüssel besteht aus zwei Zahlen  $(n, d)$  und ist ein Geheimnis von Bob.

Drücke auf den Button von **Alice** um die Perspektive von Alice einzunehmen.

**Alice** ↻


privater Schlüssel



n= 143

d= 113


öffentlicher Schlüssel



n= 143


e= 17

Internet und Oscar



**Bob** ↻

öffentlicher Schlüssel



n= 55

e= 17

Hallo Bob


Schlüssel anwenden >>

Alice verfasst die Nachricht „Hallo Bob“. Sie holt sich den **öffentlichen** Schlüssel von Bob und verschlüsselt ihre Nachricht durch Drücken von **Schlüssel anwenden** (Kapitel 2). Durch Drücken von **>>** sendet sie die verschlüsselte Nachricht zu Bob.

Wechsle in die Perspektive von **Bob**.

**Bob** ↻


öffentlicher Schlüssel



n= 55

e= 17

privater Schlüssel



n= 55

d= 33

allezeg

<< Schlüssel anwenden

Nur Bob kann die Nachricht entschlüsseln, da er im Besitz des **privaten** Schlüssels ist (Kapitel 2). Wähle den privaten Schlüssel aus und drücke auf **Schlüssel anwenden**.

### Aufgabe

Sende eine Antwort von Bob an Alice. Beschreibe den Ablauf.

---

---

---

### Aufgabe

Alice baut sich ein Schlüsselpaar. Bob baut sich kein Schlüsselpaar. Welche der beiden Aussagen ist richtig?

- |  |  |
|--|--|
| <input type="checkbox"/> Alice kann eine verschlüsselte Nachricht an Bob senden. | <input type="checkbox"/> Bob kann eine verschlüsselte Nachricht an Alice senden. |
|--|--|

### Aufgabe

Welche Schlüssel kennt Alice?

- |                                       |                                     |   |   |
|---------------------------------------|-------------------------------------|---|---|
| <input type="checkbox"/> privat Alice | <input type="checkbox"/> privat Bob | <input type="checkbox"/> öffentlich Alice | <input type="checkbox"/> öffentlich Bob |
|---------------------------------------|-------------------------------------|---|---|

Welche Schlüssel kennt Bob?

- |                                       |                                     |   |   |
|---------------------------------------|-------------------------------------|---|---|
| <input type="checkbox"/> privat Alice | <input type="checkbox"/> privat Bob | <input type="checkbox"/> öffentlich Alice | <input type="checkbox"/> öffentlich Bob |
|---------------------------------------|-------------------------------------|---|---|

Welche Schlüssel kennt Oscar?

- |                                       |                                     |   |   |
|---------------------------------------|-------------------------------------|---|---|
| <input type="checkbox"/> privat Alice | <input type="checkbox"/> privat Bob | <input type="checkbox"/> öffentlich Alice | <input type="checkbox"/> öffentlich Bob |
|---------------------------------------|-------------------------------------|---|---|

Mit welchem Schlüssel verschlüsselt Bob eine Nachricht an Alice?

- |                                       |                                     |   |   |
|---------------------------------------|-------------------------------------|---|---|
| <input type="checkbox"/> privat Alice | <input type="checkbox"/> privat Bob | <input type="checkbox"/> öffentlich Alice | <input type="checkbox"/> öffentlich Bob |
|---------------------------------------|-------------------------------------|---|---|

Mit welchem Schlüssel entschlüsselt Alice eine Nachricht von Bob?

- |                                       |                                     |   |   |
|---------------------------------------|-------------------------------------|---|---|
| <input type="checkbox"/> privat Alice | <input type="checkbox"/> privat Bob | <input type="checkbox"/> öffentlich Alice | <input type="checkbox"/> öffentlich Bob |
|---------------------------------------|-------------------------------------|---|---|

## 1.3 Zusatz: Erste Versuche

In Kapitel 2 und 3 wird gezeigt, wie die Idee von öffentlichen und privaten Schlüsseln beim RSA-Verfahren umgesetzt wird. In diesem Zusatzkapitel sollen einige Beobachtungen beschrieben werden, die erklären warum das RSA-Verfahren auf diese Art und Weise funktionieren muss.

### 1.3.1 Versuch 1: Modulare Addition

### 1.3.2 Versuch 2: Modulare Multiplikation

### 1.3.3 Versuch 3: Modulares Potenzieren

## 2 Ver-und Entschlüsseln von Nachrichten

Wir wollen nun verstehen, was beim Ver-und Entschlüsseln der Nachrichten passiert. Was läuft also im Hintergrund ab, wenn auf Schlüssel anwenden gedrückt wird.

Zunächst machen wir uns klar, dass man jede Nachricht in eine Folge von Zahlenwerte übersetzten kann. Genauers dazu findest du im Zusatzkasten am Ende des Kapitels.

Wie werden nun also Zahlen mit dem RSA-Verfahren verschlüsselt?


### Aufgabe

Berechne die folgenden Kongruenzen mit der Methode der schnellen Exponentiation.

### Aufgabe

Verwende einen Schlüssel mit  $n = 11$  und schicke die Zahl 12. Was beobachtest du? Was muss beim Senden von Nachrichten also bachtet werden?

## 3 Herstellung eines Schlüsselpaars

Damit das Ver-und Entschlüsseln auf diese Art und Weise funktioniert, muss dass eigene Schlüsselpaar nach einer festgelegten Methode gebaut werden. Bei Drücken auf  passiert genau das.

### Aufgabe

Konstruiere dein eigenes Schlüsselpaar. Verwende keine Primzahlen  $> 30$ .

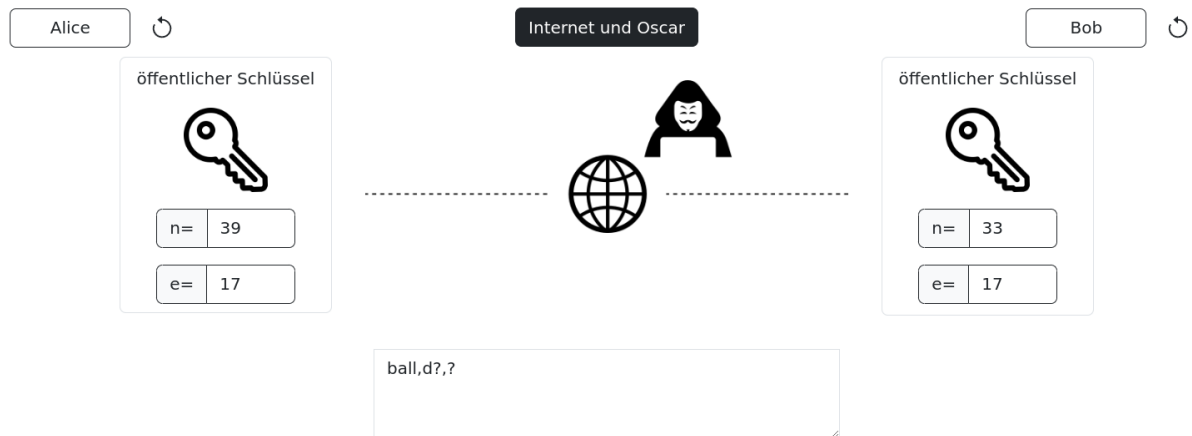
Hefte deinen öffentlichen Schlüssel mit Namen an die Tafel.

Nehme dir einen freien öffentlichen Schlüssel von der Tafel und schicke eine verschlüsselte Zahlenfolge (max. 5 Zahlen).

Gebe dem Inhaber des öffentlichen Schlüssels deine Nachricht.

Entschlüssele die Nachricht, die du erhalten hast.

## 4 Angriffsmöglichkeiten

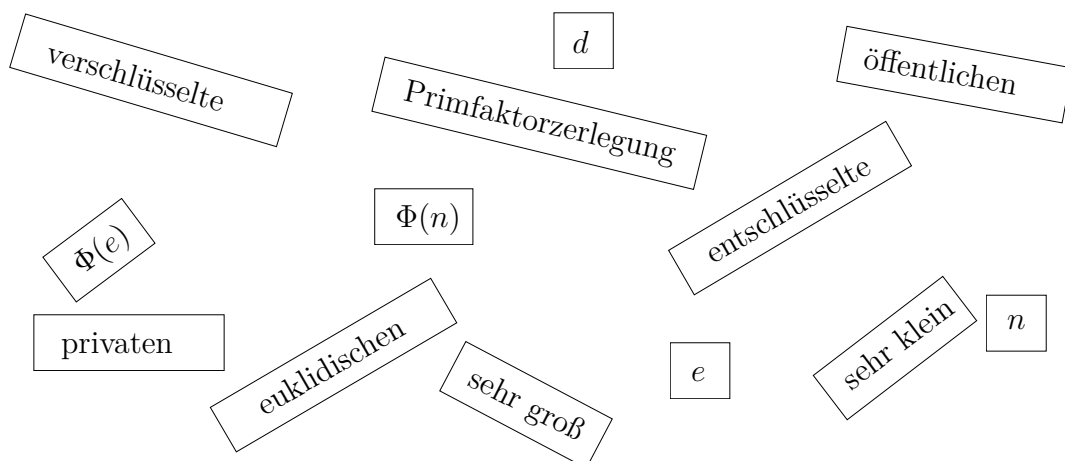


In diesem Abschnitt wollen wir die Perspektive von Oscar einnehmen. Drücke auf den Button **Oscar** um seine Situation grafisch darzustellen.

### 4.1 Primfaktorzerlegung von $n$

Fülle den Lückentext mit den Begriffen. Nicht alle Begriffe können verwendet werden.

Oscar kann nur an die \_\_\_\_\_ Nachricht gelangen und kennt den \_\_\_\_\_ Schlüssel. Um die Nachricht zu entschlüsseln benötigt er aber den \_\_\_\_\_ Schlüssel, genauer die Zahl \_\_\_\_\_. Diese ist das Inverse zu \_\_\_\_\_ modulo \_\_\_\_\_. Das lässt sich leicht mit dem erweiterten \_\_\_\_\_ Algorithmus berechnen, wenn  $\Phi(n)$  bekannt ist. Wenn  $n$  \_\_\_\_\_ ist, dauert es unfassbar lange  $\Phi(n)$  zu bestimmen. Nur wenn man die \_\_\_\_\_ von  $n$  kennt, gilt sofort  $\Phi(n) = (p - 1) \cdot (q - 1)$ .



Zusammenfassung: Angriff durch Primfaktorzerlegung

$$\begin{aligned}n &= p \cdot q \\ \Phi(n) &= (p - 1) \cdot (q - 1) \\ d^{-1} &\equiv e \pmod{\Phi(n)} \\ x &\equiv y^d \pmod{n}\end{aligned}$$

### Aufgabe

Hacke den privaten Schlüssel zu einem öffentlichen Schlüssel. Verwende das Primzahlsieb.

Um sich gegen diesen Angriff zu verteidigen wählt man ein sehr großes  $n$ . Ein RSA-Schlüssel in der Praxis besteht in der Regel aus 2048 Bit. Es ist dann sehr einfach  $n$  selber zu konstruieren, aber unmöglich die Zerlegung von  $n$  als Außenstehender zu finden. Außer man benötigt die Zerlegung erst in 100 Jahren.

## 4.2 Digitale Signatur

Ein weiterer Angriff könnte durch „Social Engineering“ (soziale Manipulation) erfolgen. Dabei würde Oscar so tun, als wäre er Bob. Wenn Alice eine Anfrage an Bob schickt um den öffentlichen Schlüssel zu holen, könnte Oscar die Anfrage abfangen und stattdessen seinen öffentlichen Schlüssel an Alice senden. Wenn Alice die verschlüsselte Nachricht an Bob schickt, kann Oscar diese Nachricht entschlüsseln.

Um dieses Problem auszuschließen verwendet man in der Praxis digitale Signaturen.