ETH zürich


# RSA

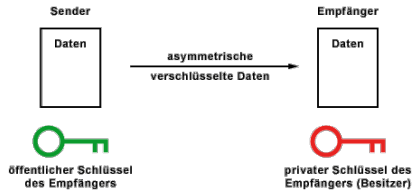
**Patrick Amrein**
**Alessandro Lägeler**
**Markus Schwagenscheidt**

# Symmetrische und asymmetrische Kryptoverfahren

**Symmetrische Verfahren:** Sender und Empfänger haben den **gleichen, geheimen Schlüssel**.

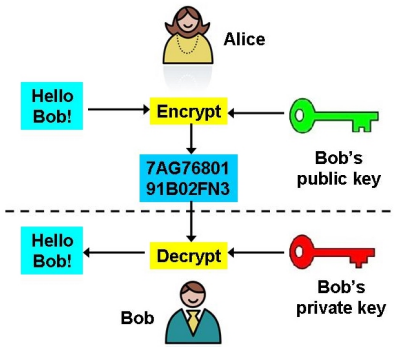

**Sicherheitsrisiko:** Austausch des geheimen Schlüssels.

**Asymmetrische Verfahren:** Sender und Emfpänger benutzen **unterschiedliche Schlüssel**.




**Vorteil:** Privater Schlüssel wird nie weitergegeben.

# Public Key Verfahren mit Alice und Bob




**Frage:** Wie kann Bob eine verschlüsselte Nachricht an Alice senden?

# RSA - Kodierung

Ein wichtiges **Public Key Verfahren** ist **RSA** - entwickelt 1977 von Rivest, Shamir und Adleman.

Um eine Nachricht mit RSA zu verschlüsseln, muss sie zunächst **als natürliche Zahl codiert** werden.

**Beispiel:** Kodiere jeden Buchstaben als Zahl von 01 bis 26, also

| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |

**Aufgaben:**

1. Kodieren Sie die Nachricht `hallo welt` als Zahl.
2. Dekodieren Sie die Nachricht 05200826210518090308.
3. Warum erhält man keine sinnvolle Kodierung, wenn man die Nullen in 01,02, usw. weglässt?

In der Praxis verwendet man eine standardisierte Kodierung wie ASCII oder UTF-8, und stellt Zahlen binär dar.

# RSA - Erzeugung des öffentlichen und privaten Schlüssels

Bevor das RSA Verfahren eingesetzt werden kann, müssen der **öffentliche und der private Schlüssel gewählt** werden.

## RSA Schlüsselerzeugung

1. Wähle zwei verschiedene Primzahlen $p$ und $q$.
2. Berechne den **RSA-Modul** $N = p \cdot q$.
3. Berechne $\varphi(N) = (p-1) \cdot (q-1)$.
4. Wähle eine zu $\varphi(N)$ teilerfremde Zahl $e$ mit $1 < e < \varphi(N)$.
5. Berechne das multiplikative Inverse $d$ von $e$ mod $\varphi(N)$ mit $1 < d < \varphi(N)$.
   Es gilt also $e \cdot d \equiv 1 \pmod{\varphi(N)}$.

Der **öffentliche Schlüssel** ist das Zahlenpaar $(e, N)$.
Der **private Schlüssel** ist das Zahlenpaar $(d, N)$.

Die Zahlen $p$, $q$, $\varphi(N)$ werden nicht mehr benötigt und sollten sicher vernichtet werden.

# RSA Schlüsselerzeugung - Ein Beispiel

## RSA Schlüsselerzeugung

1. Wähle zwei verschiedene Primzahlen $p$ und $q$.
2. Berechne den **RSA-Modul** $N = p \cdot q$.
3. Berechne $\varphi(N) = (p-1) \cdot (q-1)$.
4. Wähle eine zu $\varphi(N)$ teilerfremde Zahl $e$ mit $1 \leq e < \varphi(N)$.
5. Berechne das multiplikative Inverse $d$ von $e$ mod $\varphi(N)$ mit $1 \leq d < \varphi(N)$.
   Es gilt also $e \cdot d \equiv 1 \pmod{\varphi(N)}$.

**Aufgaben:**

1. Wählen sie $p = 11$ und $q = 13$, sowie den öffentlichen Schlüssel $e = 23$.
   Bestimmen Sie den RSA-Modul $N$ sowie den privaten Schlüssel $d$.
2. Machen Sie sich klar: kennt ein Angreifer $p, q$, oder $\varphi(N)$, so kann er den privaten Schlüssel $d$ schnell berechnen.
3. Wenn man nur den öffentlichen Schlüssel $(e, N)$ kennt, wie kann man daraus den privaten Schlüssel $d$ berechnen? Welcher der Rechenschritte geht schnell, wo könnte es ein Problem geben?

# RSA - Verschlüsselung

**Schlüssel:** Öffentlicher Schlüssel $(e, N)$ und privater Schlüssel $(d, N)$.

Wir nehmen an, das alle Nachrichten als Zahlen kodiert sind, **die kleiner als $N$ sind.**
Andernfalls muss die Nachricht in kleinere Blöcke zerlegt werden.

## RSA

**Verschlüsselung** eines Klartexts $m$: berechne Geheimtext

$$c = m^e \pmod{N}$$

**Entschlüsselung** eines Geheimtexts $c$: berechne Klartext

$$m = c^d \pmod{N}$$

**Aufgabe:** Wähle den Modul $N = 143$ und als Schlüssel $e = 23$ und $d = 47$.

1. Verschlüsseln Sie die Nachricht $m = 7$, und entschlüsseln Sie $c = 3$.
2. Zeigen Sie, dass man bei der Entschlüsselung des Geheimtexts $c = m^e \pmod{N}$ tatsächlich den ursprünglichen Klartext $m$ zurück erhält.
3. Warum muss der Klartext $m$ kleiner als der RSA-Modul $N$ sein?

# RSA - Ein vollständiges Beispiel

**Alice** möchte **Bob** die Nachricht

$$m = \texttt{super}$$

schicken. Dazu braucht sie Bob's **öffentlichen Schlüssel**.

## Bob's Schlüsselerzeugung

1. Bob wählt $p = 19$ und $q = 23$.
2. **RSA-Modul:** $N = p \cdot q = 437$.
3. $\varphi(N) = (p-1) \cdot (q-1) = 18 \cdot 22 = 396$.
4. Bob wählt den **öffentlichen Schlüssel** $e = 41$, so dass $\mathrm{ggT}(e, \varphi(N)) = 1$.
5. Er berechnet den **privaten Schlüssel** $d = 29$, so dass $e \cdot d \equiv 1 \pmod{\varphi(N)}$.

Bob **veröffentlicht** den RSA-Modul $N = 437$ und den öffentlichen Schlüssel $e = 41$.

Den privaten Schlüssel $d = 29$ **behält er für sich**.

Die Zahlen $p = 19, q = 23$ und $\varphi(N) = 396$ **vernichtet er** zur Sicherheit.

# RSA - Ein vollständiges Beispiel

## Bob's Schlüssel

RSA-Modul $N = 437$, öffentlicher Schlüssel $e = 41$, privater Schlüssel $d = 29$.

**Alice** will **Bob** die Nachricht

$$m = \texttt{super}$$

schicken.

Mittels der Kodierung

| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |

erhält Alice

$$m = \texttt{super} = 19\,21\,16\,05\,18$$

Die Zahl ist **größer als der Modul** $N = 437$, und muss daher in **Blöcke der Länge** $2$ geteilt werden. Alice verschlüsselt also die Nachrichten

$$m_1 = 19, \quad m_2 = 21, \quad m_3 = 16, \quad m_4 = 05, \quad m_5 = 18.$$

# RSA - Ein vollständiges Beispiel

## Bob's Schlüssel

RSA-Modul $N = 437$, öffentlicher Schlüssel $e = 41$, privater Schlüssel $d = 29$.

Alice verschlüsselt die Klartexte

$$m_1 = 19, \quad m_2 = 21, \quad m_3 = 16, \quad m_4 = 05, \quad m_5 = 18.$$

Dazu berechnet sie mittels des **öffentlichen Schlüssels** $e = 41$ die **Geheimtexte**

$$\begin{aligned} c_1 &\equiv 19^{41} \equiv 152 \pmod{437} \\ c_2 &\equiv 21^{41} \equiv 89 \pmod{437} \\ c_3 &\equiv 16^{41} \equiv 403 \pmod{437} \\ c_4 &\equiv 5^{41} \equiv 237 \pmod{437} \\ c_5 &\equiv 18^{41} \equiv 246 \pmod{437} \end{aligned}$$

Alice schickt nun die Geheimtexte $c_1, \ldots, c_5$ and Bob.

# RSA - Ein vollständiges Beispiel

## Bob's Schlüssel

RSA-Modul $N = 437$, öffentlicher Schlüssel $e = 41$, privater Schlüssel $d = 29$.

Bob erhält von Alice die Geheimtexte

$$c_1 = 152, \quad c_2 = 89, \quad c_3 = 403, \quad c_4 = 237, \quad c_5 = 246.$$

Er berechnet mittels seines **privaten Schlüssels** $d = 29$ die **Klartexte**

$$\begin{aligned} m_1 &\equiv 152^{29} \equiv 19 \pmod{437} \\ m_2 &\equiv 89^{29} \equiv 21 \pmod{437} \\ m_3 &\equiv 403^{29} \equiv 16 \pmod{437} \\ m_4 &\equiv 237^{29} \equiv 5 \pmod{437} \\ m_5 &\equiv 246^{29} \equiv 18 \pmod{437} \end{aligned}$$

Da $N = 437$ drei Stellen hat, weiß Bob, dass Alice die Nachricht in Blöcke der Länge $2$ geteilt hat. Darum ergänzt er $5$ zu $05$.

Bob dekodiert die Nachricht $19\,21\,16\,05\,18$ und erhält den Klartext `super`

# RSA - Sicherheit

Die **Sicherheit von RSA** beruht auf zwei **Annahmen**:

1. Große Zahlen $N$ lassen sich nur schwer in ihre **Primfaktoren zerlegen.**
   Deshalb lässt sich der private Schlüssel $d$ nicht aus dem öffentlichen Schlüssel $(e, N)$ berechnen.
2. **Wurzelziehen modulo** $N$ ist für große $N$ schwierig zu berechnen.
   Deshalb lässt sich der Klartext $m$ nicht ohne Kenntnis von $d$ aus dem Geheimtext $c = m^e \pmod N$ berechnen.

**Aufgaben:**

1. Gegeben sei der öffentliche Schlüssel $N = 25019$ und $e = 37$.
   Bestimmen sie den privaten Schlüssel $d$.
2. Gegeben sei $N = 20413$ und $e = 3$. Wir haben den Geheimtext $c = 343$ abgefangen.
   Bestimmen Sie den zugehörigen Klartext $m$ (ohne Bestimmung des privaten Schlüssels).

# RSA - Aufgaben

**Aufgaben:**

1. Erzeugen Sie jeweils ihre eigenen öffentlichen und privaten Schlüssel.
   Wählen Sie dazu zwei **dreistellige** Primzahlen $p$ und $q$.
2. Bilden Sie Zweierteams und schicken Sie sich gegenseitig eine (kurze) verschlüsselte Nachricht.
3. Knacken Sie den privaten Schlüssel Ihrer Teampartnerin.

*Hinweis:* Sie können für die Berechnungen z.B. `wolframalpha.com` benutzen.

# Fermat-Angriff auf RSA

Liegen die Primzahlen $p, q$ **zu dicht beisammen**, kann man $N$ **schnell faktorisieren**, und **RSA brechen**.

**Idee:** Ist $N = pq$ mit zwei ganzen Zahlen $p \geq q > 1$, so können wir schreiben:

$$N = pq = \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2 = a^2 - b^2, \qquad \text{mit} \quad a = \frac{p+q}{2},\ b = \frac{p-q}{2}.$$

Liegen $p, q$ nah beieinander, so ist $a \sim p \sim \sqrt{N}$ und $b$ ist relativ klein.

## Faktorisierungsverfahren von Fermat

**Eingabe:** Eine zusammengesetzte Zahl $N = pq = a^2 - b^2$.

1. Der kleinstmögliche Wert für $a$ ist $a = \lceil \sqrt{N} \rceil$.
2. Teste, ob $a^2 - N$ ein Quadrat ist. In diesem Fall ist $b = \sqrt{a^2 - N}$, und wegen

   $$N = a^2 - b^2 = (a+b)(a-b) = pq$$

   erhalten wir die zwei Teiler $p, q$ von $N$.
3. Ist $a^2 - N$ kein Quadrat, so mache mit $a + 1$ weiter, dann mit $a + 2$, usw.

# Fermat Faktorisierung

## Faktorisierungsverfahren von Fermat

1. Setze $a = \lceil \sqrt{N} \rceil$.
2. Prüfe ob $a^2 - N$ ein Quadrat ist. Dann ist $b = \sqrt{a^2 - N}$, und
$$p = (a + b), \qquad q = (a - b).$$
3. Falls $a^2 - N$ kein Quadrat ist, so mache mit $a + 1$ weiter, usw.

**Aufgabe:** Faktorisiere $12827$ und $314731$.

# RSA mit kleinem öffentlichem Schlüssel

Ist der öffentliche Schlüssel $e$ **zu klein**, so kann man **kurze Nachrichten ohne den privaten Schlüssel** $d$ **entschlüsseln**.

**Problem:** Wenn $c = m^e$ kleiner als $N$ ist, wird nicht modulo $N$ gerechnet, und man kann $m$ durch Ziehen der $e$-ten Wurzel aus $c$ erhalten.

**Beispiel**

Wir wählen $p = 13, q = 17$ und erhalten $N = 221$.
Als öffentlichen Schlüssel wählen wir $e = 3$.
Wir verschlüsseln die Nachricht $m = 4$:

$$c \equiv m^e \equiv 4^3 \equiv 64 \pmod{221}.$$

Da $64$ kleiner als $221$ ist, wurde nicht modulo $221$ gerechnet.
Ein Angreifer kann nun einfach die dritte Wurzel aus $64$ ziehen, und erhält die Nachricht $m = 4$ zurück.

**Mögliche Lösung:** Stelle sicher (z.B. durch sogenanntes **Padding**) dass $m^e$ stets **größer als** $N$ ist.

# Gleicher Modul mit verschiedenen Schlüsseln

Verwendet man den **gleichen Modul** $N$ mit **verschiedenen öffentlichen Schlüsseln** $e, e'$, um **dieselbe Nachricht** $m$ zu verschlüsseln, so kann ein Angreifer $m$ leicht rekonstruieren.

## Angriff

Bob benutzt immer den gleichen Modul $N$, aber wählt zwei verschiedene öffentliche Schlüssel $e, e'$. Zufällig ist dabei $\mathrm{ggT}(e, e') = 1$.
Alice versendet regelmäßig dieselbe Nachricht $m$ an Bob und verschlüsselt sie dabei abwechselnd mit $e$ und $e'$.
Eve fängt die verschlüsselten Nachrichten

$$c = m^e \pmod N,$$
$$c' = m^{e'} \pmod N,$$

ab.
Eve kann nun $a, b \in \mathbb{Z}$ finden so dass

$$ae + be' = 1.$$

Sie berechnet nun

$$c^a c'^b = (m^e)^a (m^{e'})^b = m^{ae+be'} = m,$$

und erhält damit die Nachricht $m$.