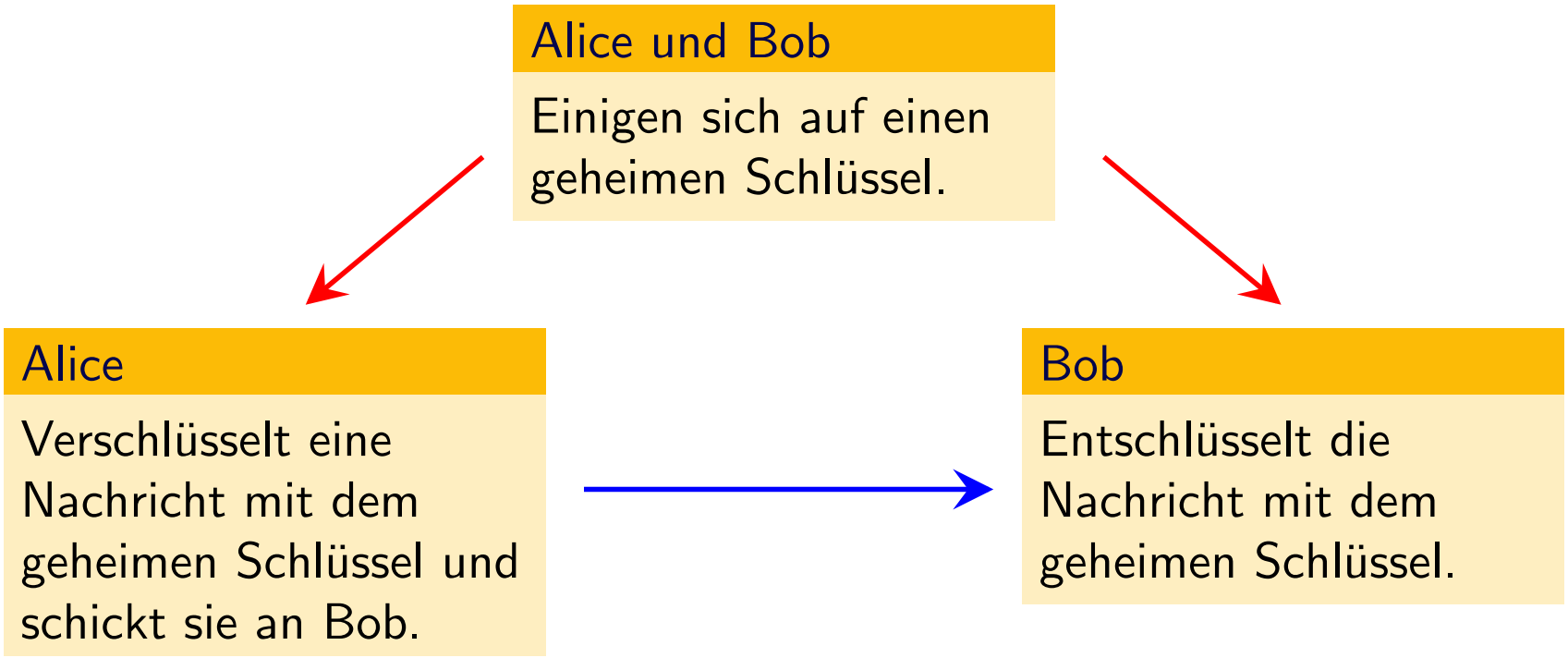
Das RSA Kryptosystem

www.math.tu-berlin.de/~kant

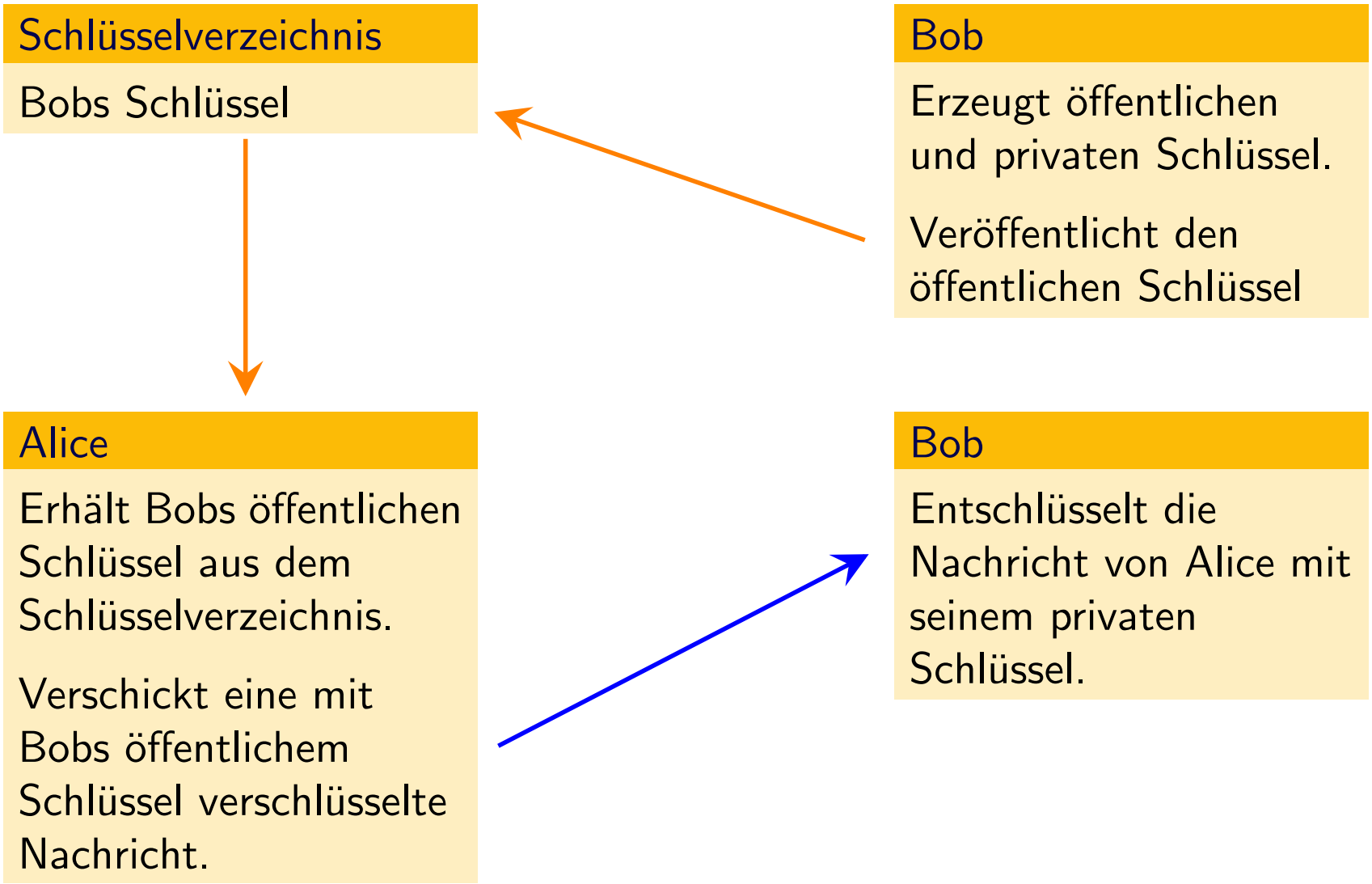
Institut für Mathematik
Technische Universität Berlin

Kryptografie mit geheimem Schlüssel



Nachteil: Alice und Bob müssen miteinander kommunizieren um sich auf einen geheimen Schlüssel zu einigen, den sie für sichere Kommunikation verwenden.

Kryptografie mit öffentlichem Schlüssel



Realisierung von Kryptografie mit öffentlichen Schlüsseln

Einweg Funktionen mit Trapdoor

Können leicht berechnet werden, aber nicht in vertretbarer Zeit invertiert werden. Kennt man aber eine geheime Zusatzinformation, so kann man die Funktion auch effizient invertieren.

Beispiele

- Multiplizieren ist sehr viel einfacher als Faktorisieren.
- In endlichen Strukturen ist Exponenzieren sehr viel einfacher als Wurzeln zu ziehen.
- In endlichen Strukturen ist Exponenzieren sehr viel einfacher als Logarithmen zu berechnen.

Division mit Rest

Definition

Seien $a, b \in \mathbb{N}$.
Dann gibt es $q, r \in \mathbb{N}$ mit $a = qb + r$ und $0 \leq r < b$.
Die Zahl r heißt Rest der Division von a durch b und wird mit $r = a \bmod b$ bezeichnet.

Bemerkung

Wenn $r = a \bmod b$, dann gibt es $q \in \mathbb{Z}$ mit $r - a = qb$.

Beispiele

$8 \bmod 5 = 3$
$17 \bmod 2 = 1$
$4765 \bmod 23 = 4$
$347864817 \bmod 23478 = 14769$

Kleiner Fermatscher Satz

Satz

Sei p eine Primzahl, dann gilt $x^{p-1} \bmod p = 1$ für alle $x \in \{1, \ldots, p-1\}$.

Beweis

(i) Sei $1 \leq a \leq p-1$. Wenn $x, y \in \{1, \ldots, p-1\}$ mit $x \neq y$ dann $ax \bmod p \neq ay \bmod p$.
Angenommen $ax \bmod p = ay \bmod p$ dann $a(x-y) \bmod p = 0$.
Da $-(p-1) < x - y < p-1$ folgt $x - y = 0$.

(ii) Sei $1 \leq a \leq p-1$. Wegen (i) gilt

$$\{1, \ldots, p-1\} = \{a \cdot 1 \bmod p, \ldots, a \cdot (p-1) \bmod p\}.$$

Also $\prod_{b=1}^{p-1} b \bmod p = \prod_{b=1}^{p-1} ab \bmod p = a^{p-1} \prod_{b=1}^{p-1} b \bmod p$ und damit $a^{p-1} \bmod p = 1$.

Euklidischer Algorithmus

Algorithmus

Seien $r_0, r_1 \in \mathbb{N}$ mit $r_0 > r_1$.
Bestimme q_2 und $0 < r_2 < r_1$ so dass $r_0 = q_2 r_1 + r_2$.

$\vdots$

Bestimme q_{i+2} und $0 < r_{i+2} < r_{i+1}$ so dass $r_i = q_{i+2} r_{i+1} + r_{i+2}$.

$\vdots$

Bis zu q_m mit $r_{m-1} = q_m r_m$.

r_m ist größter gemeinsamer Teiler von r_0 und r_1.

r_m teilt r_{m-1}, r_m teilt $r_{m-2}, \ldots, r_m$ teilt r_1 und r_m teilt r_0.
Also ist r_m Teiler von r_0 und r_1.

Ein gemeinsamer Teiler t von r_0 und r_1 teilt auch r_i für $i \leq m$.
Also teilt jeder gemeinsame Teiler von r_0 und r_1 sie Zahl r_m.

Daher ist r_m der grösste gemeinsame Teiler von r_0 und r_1.

Euklidischer Algorithmus

Beispiel

$r_0 = 74188,\ r_1 = 391$

$r_0 = 74188 = 189 \cdot 391 + 289$
$r_1 = 391 = 1 \cdot 289 + 102$
$r_2 = 289 = 2 \cdot 102 + 85$
$r_3 = 102 = 1 \cdot 85 + 17$
$r_4 = 85 = 5 \cdot 17 + 0$

Es gibt $b, c \in \mathbb{Z}$ mit $br_0 + cr_1 = r_m = \mathrm{ggT}(r_0, r_1)$.

$r_2 = r_0 - q_1 r_1,\ r_3 = r_1 - q_2 r_2 = (1 + q_1 q_2) r_1 - q_2 r_0, \ldots$
Also läßt sich r_m als Linearkombination vom r_0 und r_1 darstellen.

Zu $a, n \in \mathbb{N}$ mit $\mathrm{ggT}(a, n) = 1$ bestimme $b \in \mathbb{Z}$ mit $ba \bmod n = 1$.

Es gibt $b, c \in \mathbb{Z}$ mit $ba + cn = 1$, also $ba = 1 - cn$ und $ba \bmod n = 1$.

Chinesischer Restsatz

Chinesischer Restsatz

Seien $a, b, p, q \in \mathbb{N}$ mit $\text{ggT}(p, q) = 1$, $0 \leq a < p$ und $0 \leq b < q$.
Es gibt ein modulo pq eindeutig bestimmtes $x \in \mathbb{N}$ mit
$x \bmod p = a$ und $x \bmod q = b$.

Existenz

Bestimme $c, d \in \mathbb{N}$ mit $qc \bmod p = 1$ und $pd \bmod q = 1$.
Setze $x = acq + bdp \bmod pq$. Dann $x \bmod p = a$ und $x \bmod q = b$.

Eindeutigkeit

Zu jeder der pq Kombinationen von $0 \leq a < p$ und $0 \leq b < q$ gibt es ein x. Es gibt pq Zahlen mit $0 \leq x < pq$ also ist x eindeutig.

RSA

1977 von Ron **Rivest**, Adi **Shamir** und Len **Adleman** erfunden.

RSA war das erste Kryptosystem mit öffentlichem Schlüssel und ist noch heute das am weitesten verbreitete.

Einsatzgebiete

- Homebanking und e-commerce,
- SSL/TLS und IPSec,
- automatische Softwareupdates,
- Chipkarten und
- Reisepässe.

RSA Grundlagen

Satz

Seien p, q Primzahlen mit $\mathrm{ggT}(p, q) = 1$.
Für alle $a \in \{0, \ldots, pq - 1\}$ gilt $a^{1+c(p-1)(q-1)} \bmod pq = a$.

Beweis

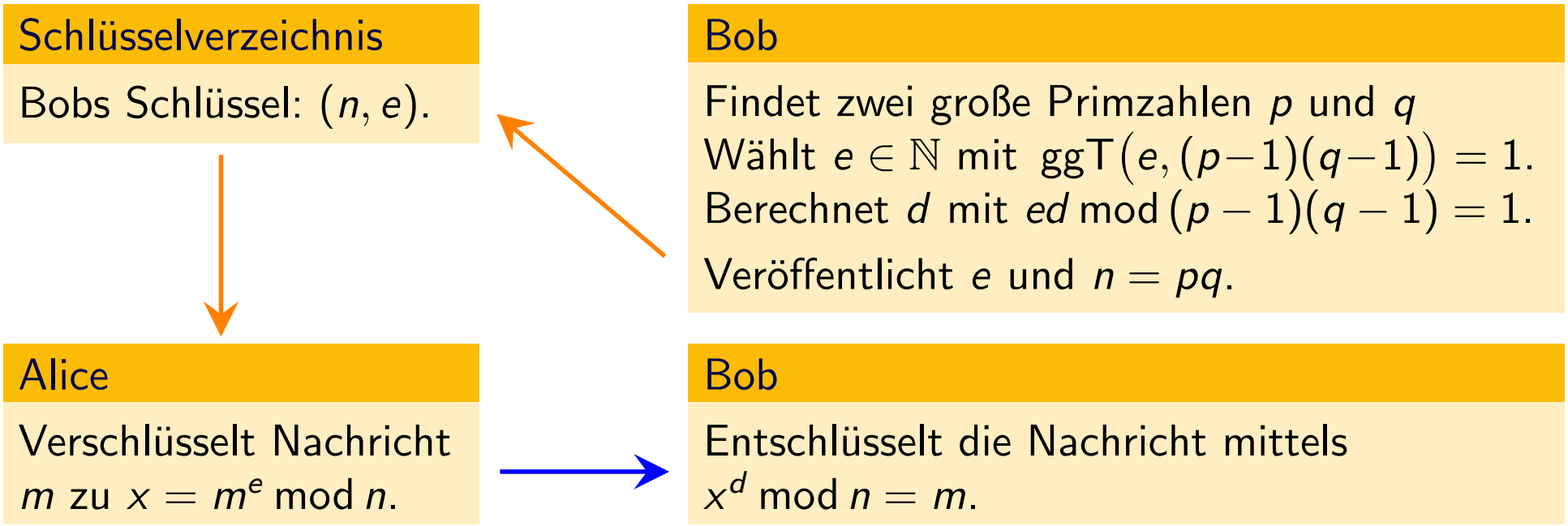
(i) p teilt a, $\mathrm{ggT}(a, q) = 1$.
Nach dem kleinen Fermatschen Satz gilt $a^{(q-1)} \bmod q = 1$.
Also $a^{1+c(p-1)(q-1)} \bmod q = a$.

Weiterhin folgt aus $a \bmod p = 0$, dass $a^{1+c(p-1)(q-1)} \bmod p = 0$.

Nach dem chinesischen Restsatz ist $a^{1+c(p-1)(q-1)} \bmod pq$ eindeutig bestimmt. Daher $a^{1+c(p-1)(q-1)} \bmod pq = a$.

(ii) $\mathrm{ggT}(a, pq) = 1$.
Nach dem kleinen Fermatschen Satz gilt:
$a^{1+c(p-1)(q-1)} \bmod p = a$ und $a^{1+c(p-1)(q-1)} \bmod q = a$.
Die Aussage folgt wiederum mit dem chinesischen Restsatz.

RSA Beschreibung



Bemerkung

Nach Satz gilt für $0 \leq m < n = pq$

$$x^d \bmod n = (m^e)^d \bmod n = m^{ed} \bmod n = m$$

da $ed \bmod (p-1)(q-1) = 1$.

RSA Beispiel

Schlüsselverzeichnis

Bobs Schlüssel:
$(n = 391, e = 5)$

Bob

Wählt Primzahlen 17 und 23
Wählt $e = 5$ mit $\mathrm{ggT}(e, 16 \cdot 22) = 1$.
Berechnet $d = 141$ mit $d \cdot 5 \bmod 16 \cdot 22 = 1$.

Veröffentlicht $e = 5$ und $n = pq = 391$.

Alice

Verschlüsselt 8 zu
$x = 8^5 \bmod 391 = 315$

Bob

Entschlüsselt die Nachricht mittels
$x^d \bmod n = 315^{141} \bmod 391 = 8$.

Die Sicherheit von RSA hängt davon ab ob

- schnellere Faktorisierungsalgorithmen entdeckt werden,
- schnellere Computer gebaut werden.

RSA Challenges

Die Firma RSA hat Preise für die Faktorisierung von ganzen Zahlen in für das RSA Kryptosystem relevanter Größenordnung ausgelobt.

RSA-140

Die Zahl RSA-140 (140 Dezimalstellen) ist

21290246318258757547497882016271517497806703963277216278233383215381949984056495911366573853021918316783107387995317230889569230873441936471

Sie wurde 1999 zerlegt in die Primfaktoren

339871742302843855453012362761387583563398649596959742349092 9302771479

und

626420018740128509615165494826444221930203717862350901911166 0653946049

Der Preis betrug 10000 US$.
Die Faktorisierung dauerte 8,9 CPU Jahre.

RSA Challenges

Die Firma RSA hat Preise für die Faktorisierung von ganzen Zahlen in für das RSA Kryptosystem relevanter Größenordnung ausgelobt.

RSA-640

Die Zahl RSA-640 (193 Dezimalstellen) ist

3107418240490043721350750035888567930037346022842727545720161948823206440518081504556346829671723286782437916272838033415471073108501919548529007337724822783525742386454014691736602477652346609

Sie wurde 2005 zerlegt in die Primfaktoren

1634733645809253848443133883865090859841783670033092312181110852389333100104508151212118167511579

und

1900871281664822113126851573935413975471896789968515493666638539088027103802104498957191261465571

Der Preis betrug 20000 US$.
Die Faktorisierung dauerte 30 2.2GHz-Opteron-CPU Jahre.

RSA Challenges

Die Firma RSA hat Preise für die Faktorisierung von ganzen Zahlen in für das RSA Kryptosystem relevanter Größenordnung ausgelobt.

RSA-1024

Die Zahl RSA-1024 (309 Dezimalstellen) ist

135066410865995223349603216278805969938881475605667027524485
143851526510604859533833940287150571909441798207282164471551
373680419703964191743046496589274256239341020864383202110372
958725762358509643110564073501508187510676594629205563685529
475213500852879416377328533906109750544334999811150056977236
890927563

Der ausgelobte Preis beträgt 100000 US$.