Informatik-Propädeutikum

Dozentin: Dr. Claudia Ermel

Betreuer: Sepp Hartung, André Nichterlein, Clemens Hoffmann

Sekretariat: Christlinde Thielcke (TEL 509b)

TU Berlin
Institut für Softwaretechnik und Theoretische Informatik
Prof. Niedermeier
Fachgruppe Algorithmik und Komplexitätstheorie
http://www.akt.tu-berlin.de

Wintersemester 2013/2014

Gliederung

Kryptologie

Wikipedia: Die Kryptologie (griechisch kryptós „versteckt, verborgen, geheim“) ist eine Wissenschaft, die sich mit Informationssicherheit beschäftigt.

Zentrales Thema: Nachrichtenübertragung in derart kodierter Form ($\rightsquigarrow$ Begriff der **Verschlüsselung** bzw. **Chiffrierung**), dass ein unbefugter Abhörer eines solchen Codes möglichst nicht auf die eigentliche Nachricht rückschließen kann.

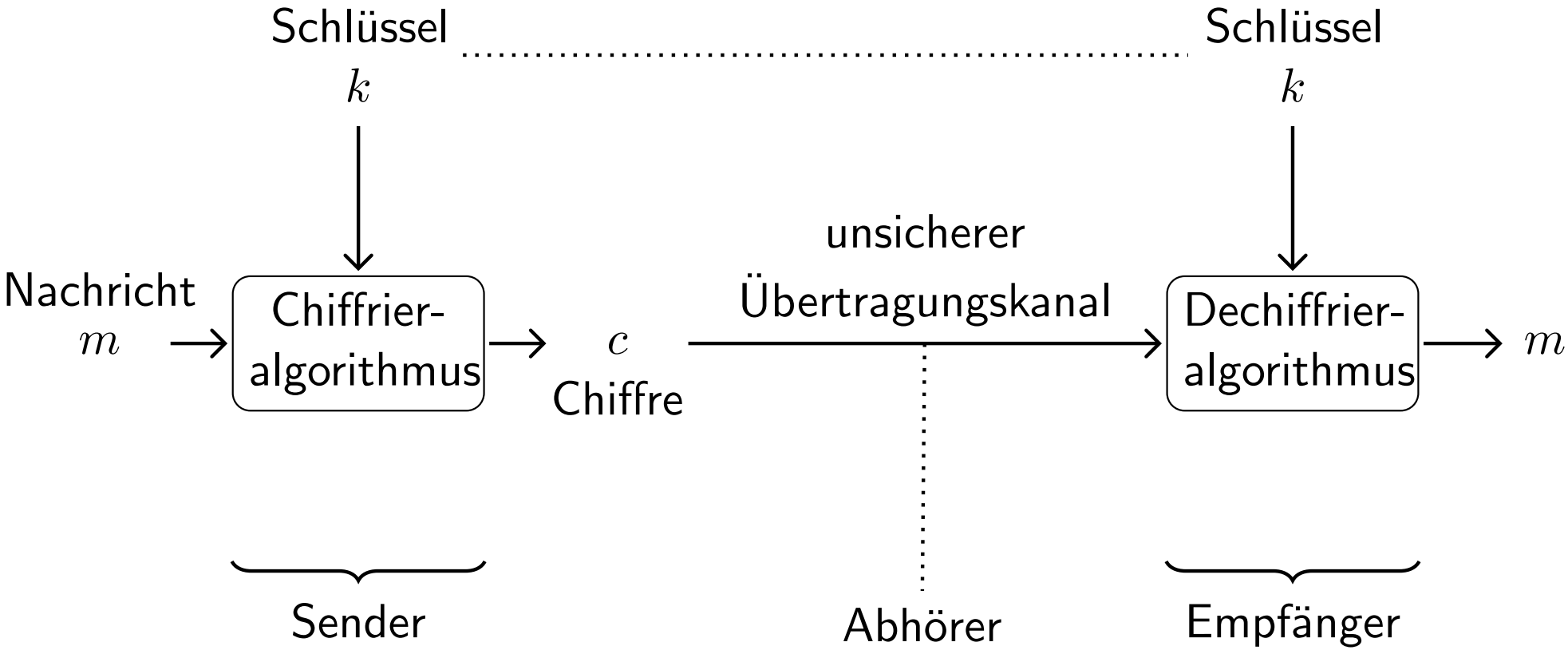
Unterteilung in zwei Teilgebiete:

Kryptographie: Wie man Nachrichten geeignet verschlüsselt.

Kryptoanalyse: Welche Methoden man aus Sicht eines unberechtigten Abhörers einsetzen kann, um evtl. doch an die verschlüsselte Nachricht heranzukommen.

Nebenbemerkung: Die **Steganographie**, sprich das Verstecken von Nachrichten in vermeintlich „harmlosen“ Objekten, wird nachfolgend nicht weiter betrachtet.

Kryptologie schematisch



Kryptologie im Altertum

Spartaner benutzten sog. „Skytalen“ zum Verschlüsseln militärischer Botschaften.

Idee: Lederstreifen mit Buchstaben um Holzstab wickeln...; Entschlüsselung nur mit gleich dickem Holzstab. Prinzip der **Transposition**, d.h. verschlüsselte Nachricht enthält genau die gleichen Buchstaben wie Originalnachricht, nur „umsortiert“.

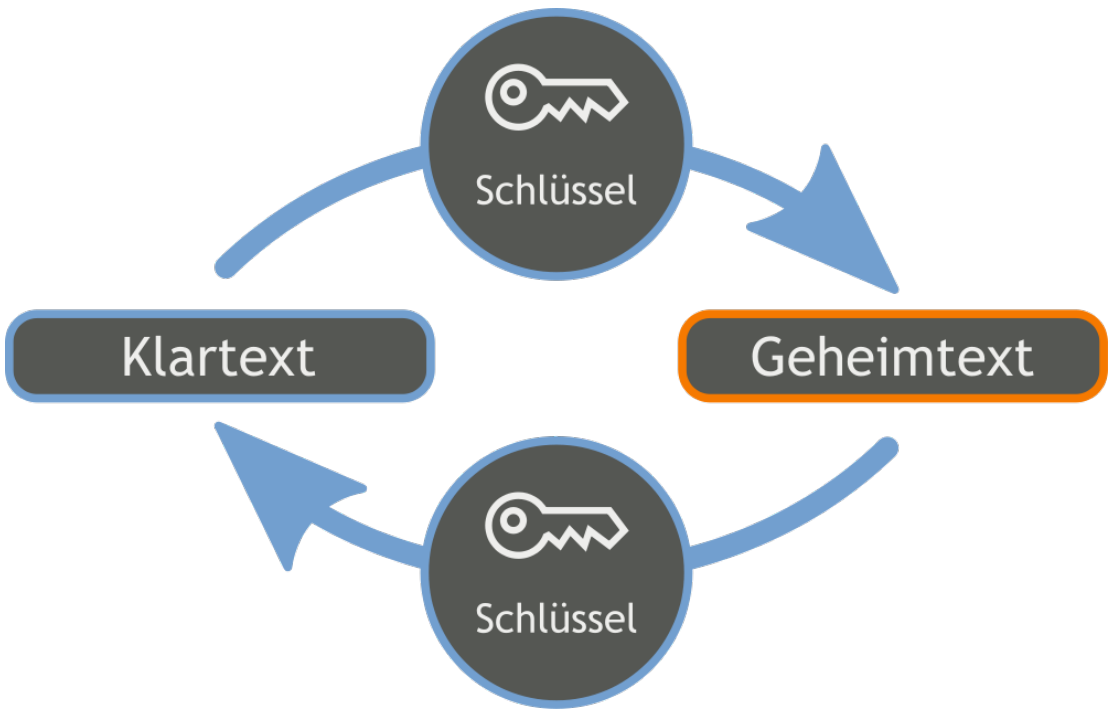
Julius Cäsar beschrieb in seinen „Anmerkungen zum gallischen Krieg“ ein anderes Verfahren: **Substitutions-Chiffre**. Jeder Buchstabe durch einen entsprechend verschobenen Buchstaben ersetzt.

Beispiel:
Originaltext „GALLIA EST OMNIS DIVISA IN PARTES TRES“ wurde verschlüsselt zu Geheimtext „JDOOLD HVW RPQLV GLYLVD LQ SDUWHV WUHV“
(Suetonius: The Life of Julius Caesar, ca. 100 n.C.).

Sichere Verschlüsselung: „One-Time-Pad“ I

Annahmen:

- Nachricht ist binär kodiert.
- Der Schlüssel wird nur einmal benutzt.
- Der Schlüssel ist eine gleichverteilt zufällig gewählte Binärzeichenkette, der genauso lange wie die Nachricht ist.
- Sender und Empfänger verfügen beide über diesen Schlüssel.



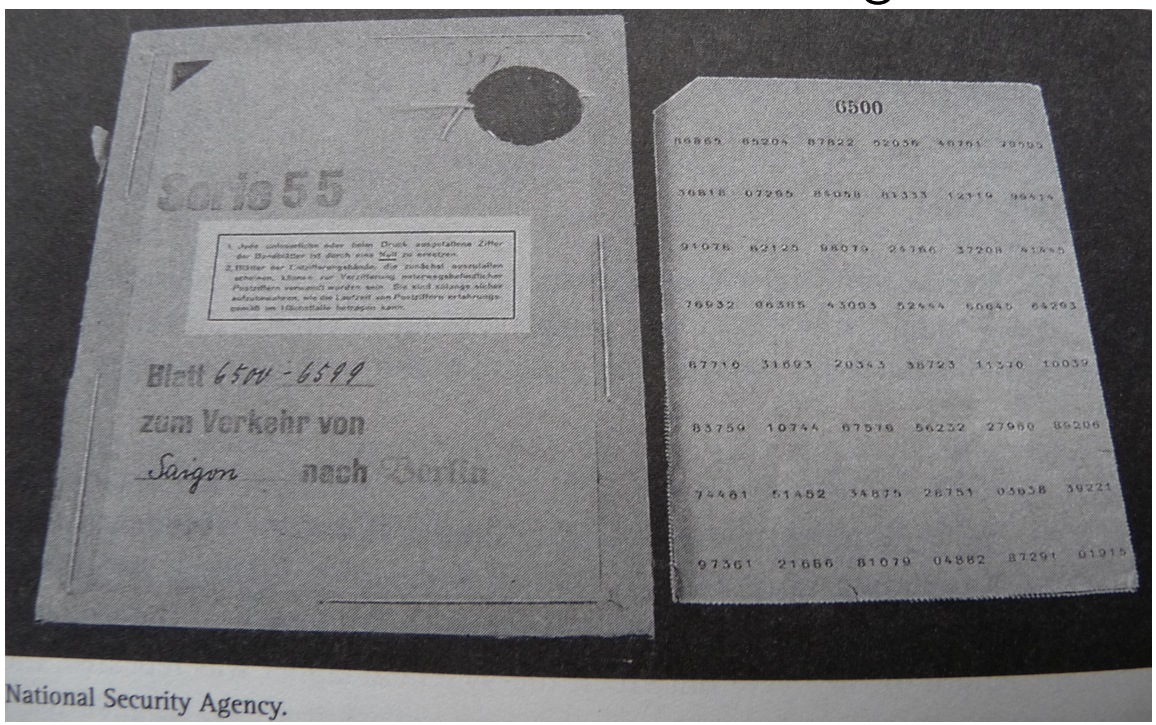
One-Time-Pad ist ein symmetrisches Verschlüsselungsverfahren

Sichere Verschlüsselung: „One-Time-Pad“ II

Vorgehen: Erzeuge mit Hilfe des Schlüssels verschlüsselten Text (Chiffre) aus dem Klartext, indem positionsweise Bits addiert werden. Beispiel: Klartext: 1100; Schlüssel: 0110; Chiffre: 1010.

Shannon: Eine per One-Time-Pad-Verfahren verschlüsselte Nachricht ist absolut sicher! (Grund: Alle Chiffren kommen mit gleicher Wahrscheinlichkeit vor; keine statistischen Rückschlüsse auf Klartext möglich.)

Probleme: Schlüsselaustausch? Schlüssel so lang wie Klartext!

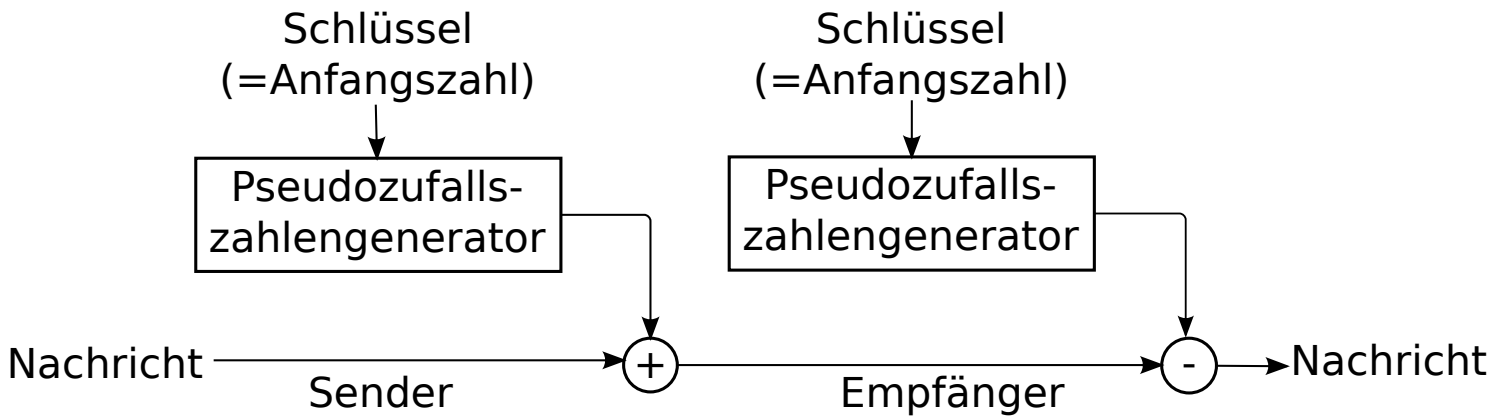


Verschlüsselung mit Pseudozufallszahlengeneratoren

Ansatz für kürzerere Schlüssel als bei One-Time-Pad:

Annahme: Sender und Empfänger besitzen beide den gleichen, „kryptographisch sicheren“ Pseudozufallszahlengenerator, der mit der gleichen Anfangszahl (das ist der Schlüssel!) initialisiert wird.

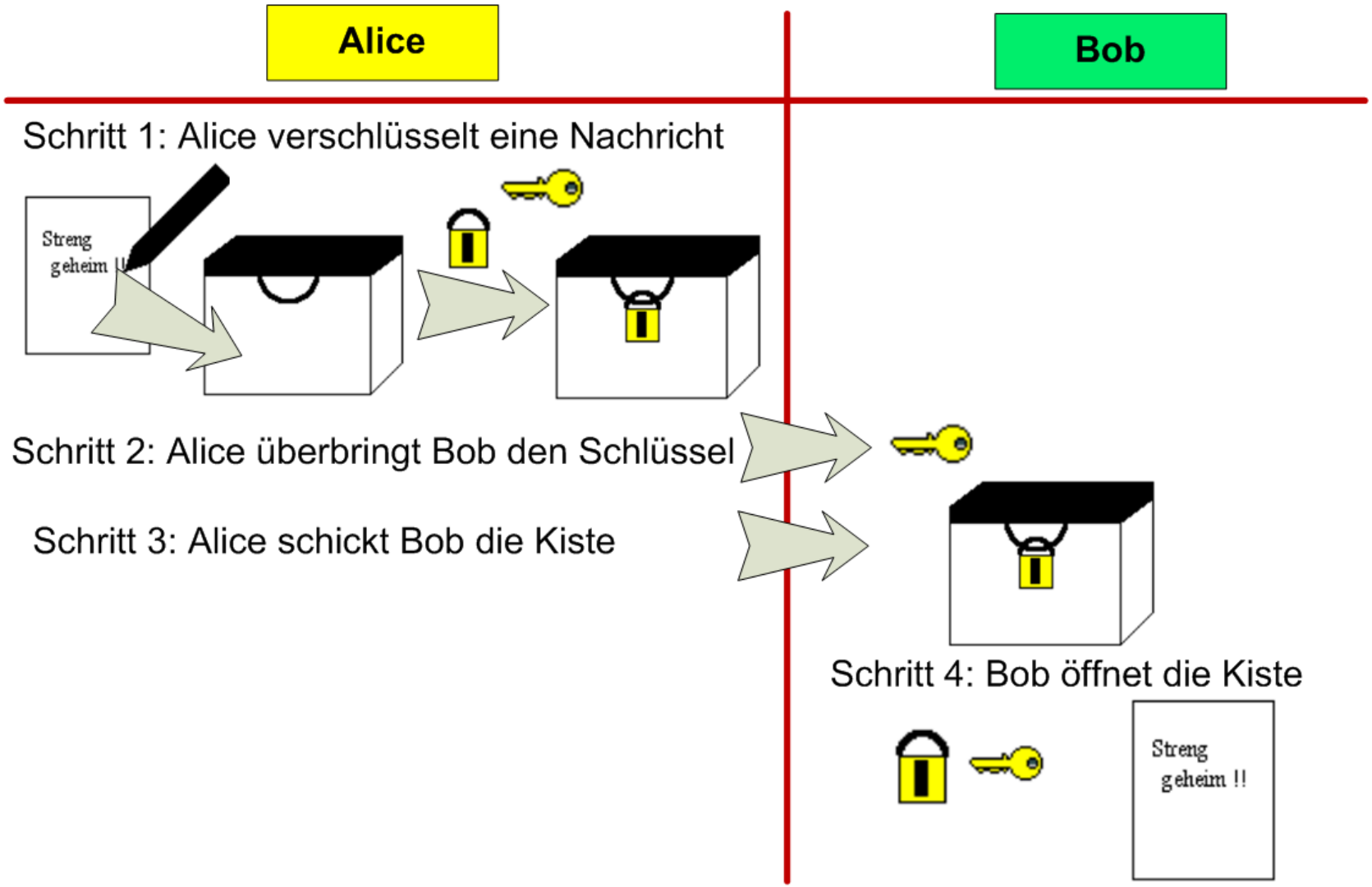
Dann: Erzeuge mit Hilfe des Pseudozufallszahlengenerators eine pseudozufällige Binärzeichenkette und verschlüssele und entschlüssele dann wie bei One-Time-Pad.



Nachteil: Sicherheit stark von Qualität des Pseudozufallszahlengenerators abhängig.

Anwendung z.B. beim Verschlüsseln eines Fernsehsignals beim Pay-TV.

Symmetrische Kryptosysteme



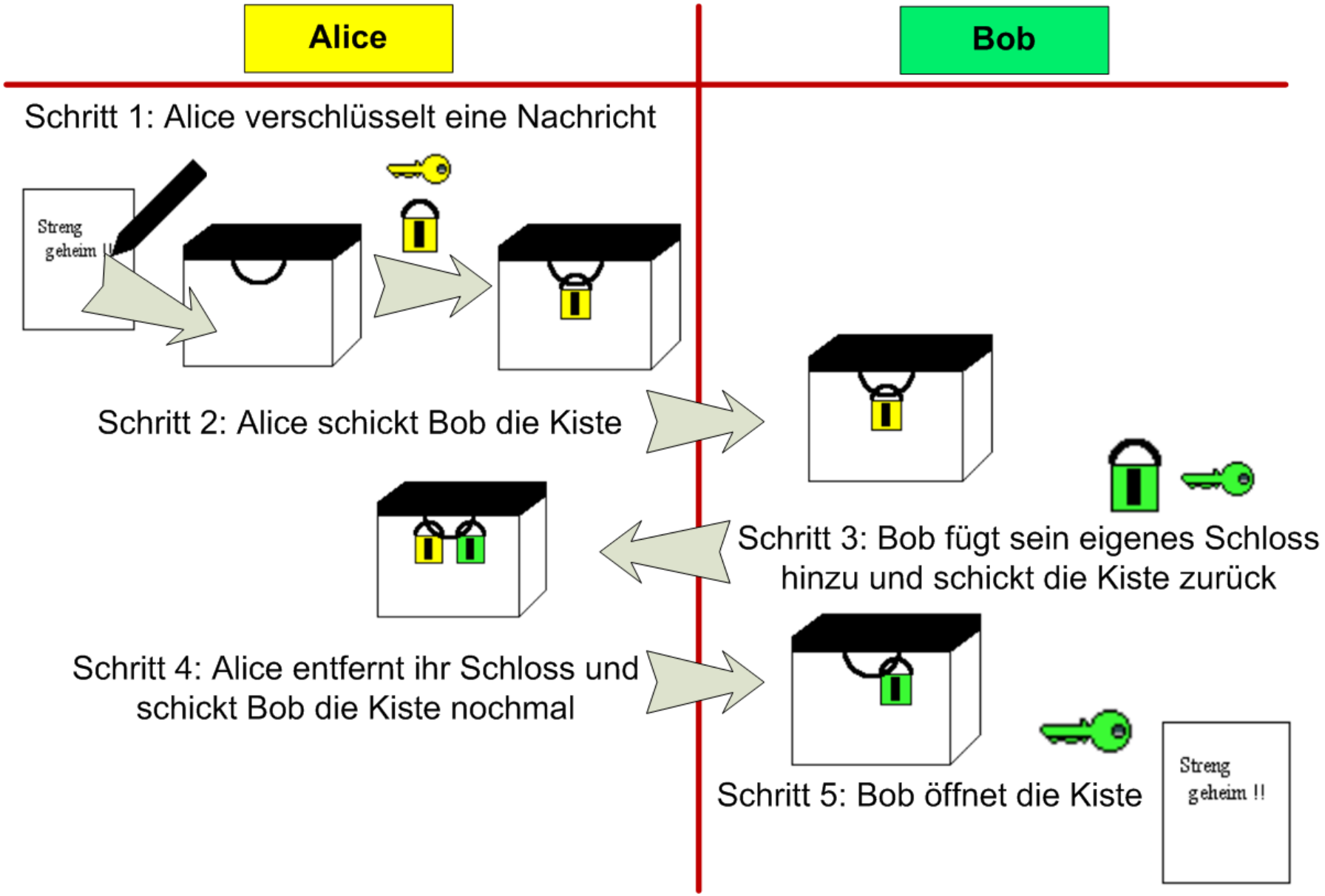
Kerckhoff'sches Prinzip

Das Prinzip besagt, dass man in der Kryptologie grundsätzlich davon ausgehen muss, dass der verwendete Verschlüsselungs- und Entschlüsselungsalgorithmus (in den vorangehenden zwei Beispielen waren diese identisch) allgemein bekannt (also nicht geheim) sind!

Ein oft verwendeter Ansatzpunkt beim Knacken von Kryptosystemen: Häufigkeitsanalyse. Verschiedene Buchstaben kommen im Text verschieden häufig vor...; Analyse des chiffrierten Textes kann diese Häufigkeiten aufdecken...

$\rightsquigarrow$ Eine mögliche Abhilfe: Datenkompression (vgl. Huffman-Codierung). Benutze für häufiger vorkommende Buchstaben mehr als eine Codierung, um so die relativen Häufigkeiten anzugleichen...

Asymmetrische Kryptosysteme



Symmetrische vs. asymmetrische Kryptosysteme

Bislang haben wir symmetrische Kryptosysteme betrachtet: Sender und Empfänger verwenden den gleichen Schlüssel.

Zentrales, hierbei ausgeblendetes Problem: Schlüsselvereinbarung.

In der **modernen Kryptologie** spielen die von Whitfield Diffie und Martin Hellman erstmals 1976 (als Konzept) publizierten asymmetrischen Verfahren eine zentrale Rolle:

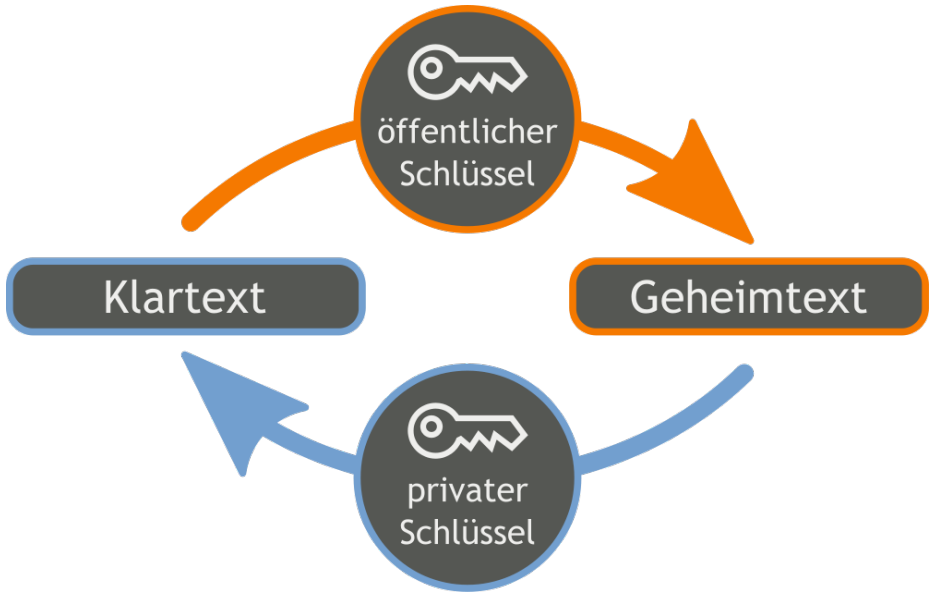
Public Key-Kryptographie: Jeder Teilnehmer am Nachrichtenaustausch besitzt einen *öffentlichen* und einen *privaten* Schlüssel. Ersterer ist jedermann zugänglich und dient zum Verschlüsseln von Nachrichten die an den Besitzer des öffentlichen Schlüssels geschickt werden, die nur dieser mit seinem geheimen Schlüssel dechiffrieren kann.

Zentral wichtig für die Realisierung: Erkenntnisse der *Zahlentheorie*!

Zusätzliche Anwendungen: Digitale Signaturen, Elektronisches Geld, Commitment Schemes, ...

Allgemeine Funktionsweise von Public-Key-Kryptosystemen

Verschlüsseln kann **jeder** mit einem öffentlichen Schlüssel; entschlüsseln **nur** der Inhaber des zugehörigen privaten Schlüssels.
$\rightsquigarrow$ *asymmetrisches* Kryptosystem.



Mögliche Nutzweise: A erzeugt privaten Schlüssel S_p sowie öffentlichen Schlüssel S_o und stellt S_o auf Homepage online. $\rightsquigarrow$ Jeder kann A eine mit S_o verschlüsselte Nachricht schreiben, die nur A mit S_p entschlüsseln kann.

Verwendung: im E-Mail-Verkehr (OpenPGP, S/MIME), in Protokollen wie SSH, SSL/TLS oder https, ...

Public-Key-Kryptosysteme: Prinzipielles Vorgehen

Basis: eine besonders große Zahl g und eine „Einweg"-Operation $\otimes$ mit der Eigenschaft $(x \otimes y) \otimes z = (x \otimes z) \otimes y$

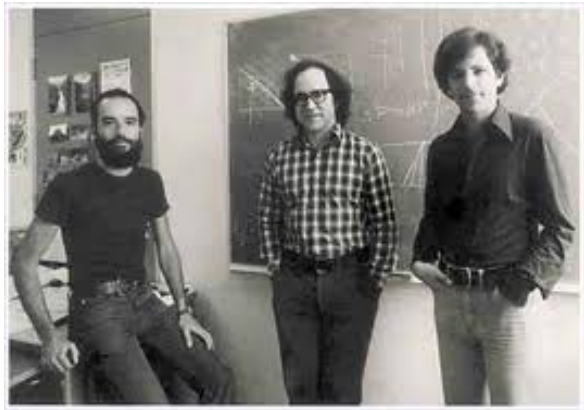
1. Alice und Bob wählen jeder zufällig eine Zahl (a und b, die *privaten Schlüssel*).
2. Alice berechnet $A = g \otimes a$, Bob berechnet $B = g \otimes b$ (die *öffentlichen Schlüssel*). Sie stellen A und B auf ihre Homepages.
3. Alice verschlüsselt Nachricht an Bob mit Schlüssel $B \otimes a$.
4. Bob entschlüsselt Nachricht mit Schlüssel $A \otimes b$.

Wieso funktioniert das?

Die Schlüssel sind gleich: $B \otimes a = (g \otimes b) \otimes a = (g \otimes a) \otimes b = A \otimes b = K$.

Selbst, wenn Abhörer A und B kennen (und g sowieso), können sie nicht K berechnen, weil sie weder a noch b kennen. Da $\otimes$ „Einweg"-Operation, sind sie auf Try- und Error-Suche angewiesen (sehr ineffizent, da g sehr groß) .

RSA-Verschlüsselung



Shamir, Rivest, Adleman

Prominentestes asymmetrisches Kryptosystem: RSA-Verschlüsselung (nach Ronald **R**ivest, Adi **S**hamir, Leonhard **A**dleman, 1978 – damals alle drei am MIT; belohnt mit Turing Award 2002).

Verschlüsseln eines Textes m (als Zahl interpretiert!) um Geheimtext c zu erhalten:

$$c \equiv m^e \mod N$$

Entschlüsseln eines Geheimtextes c um Text m zu erhalten:

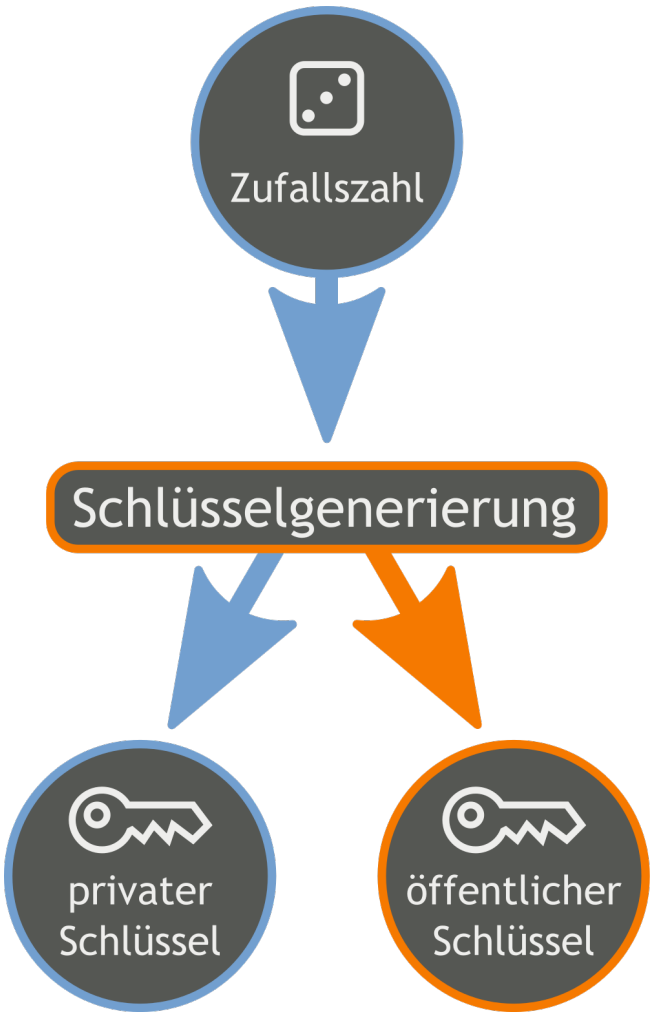
$$m \equiv c^d \mod N$$

$\rightsquigarrow$ Geheimer Schlüssel ist das Paar (d, N), öffentlicher Schlüssel das Paar (e, N).

Frage: Wie werden diese Schlüssel erzeugt?

Schlüsselerzeugung für RSA I

Frage: Wie werden die Schlüssel (d, N) und (e, N) erzeugt?

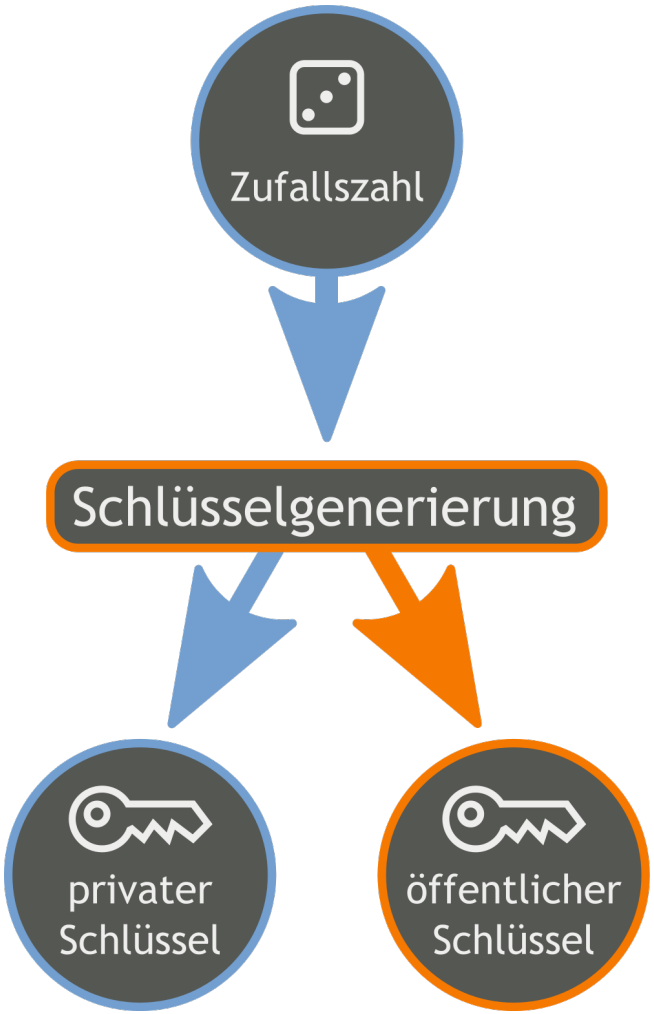


1. Nimm zwei große (mehrere hundert Stellen) *zufällige* Primzahlen p und q.
2. $N := pq$
3. Wähle $1 < e < (p-1)(q-1)$ teilerfremd zu $(p-1)(q-1)$
4. Wähle d sodass $e \cdot d \equiv 1 \mod (p-1)(q-1)$

$\rightsquigarrow$ Sicherheit basiert auf der Annahme, dass die Zerlegung großer ganzer Zahlen in ihre Primfaktoren *nicht effizient* geht. Anforderung: Entschlüsselung soll mit nur e praktisch unmöglich sein.

Schlüsselerzeugung für RSA II

Frage: Passen die Schlüssel?



Verschlüsselung E von M mit N und e:
$E(M) = M^e \bmod N$.
Entschlüsselung D von $E(M)$ mit d:

$$\begin{aligned} D(E(M)) &= (E(M))^d \bmod N \\ &= (M^e \bmod N)^d \\ &= (M^e)^d \bmod N \\ &= M \end{aligned}$$

Anmerkung: Viele benutzte Primzahltests sind randomisiert wie z. B. der „Miller-Rabin-Test“.

„Falsche“ Schlüsselerzeugung bei RSA

Frage: Was passiert wenn Primzahlen p und q nicht zufällig gewählt werden?

Beispiel: *heise Security 13.05.2008:* „Die OpenSSL-Bibliothek der Linux-Distribution Debian erzeugt seit einem fehlerhaften Patch im Jahr 2006 schwache Krypto-Schlüssel. Der Sicherheitsexperte Luciano Bello entdeckte nun in dem OpenSSL-Paket eine kritische Schwachstelle, die die **erzeugten Zufallszahlenfolgen** und somit die erzeugten Schlüssel vorhersagbar macht.“

$\rightsquigarrow$ Zentral wichtiger Aspekt: Wie gut ist mein Zufallszahlengenerator?