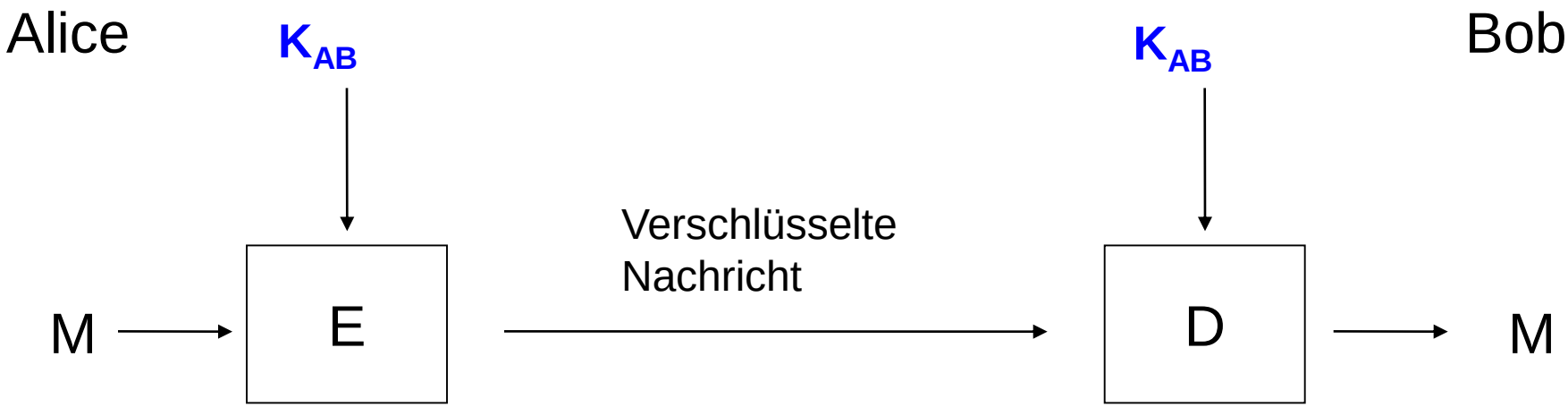
IT-Sicherheit:
Kryptographie

Asymmetrische Kryptographie

Das Problem mit der symmetrischen Verschlüsselung

- Die beiden Kommunikationspartner Alice und Bob brauchen ein **gemeinsames Geheimnis K_{AB}**

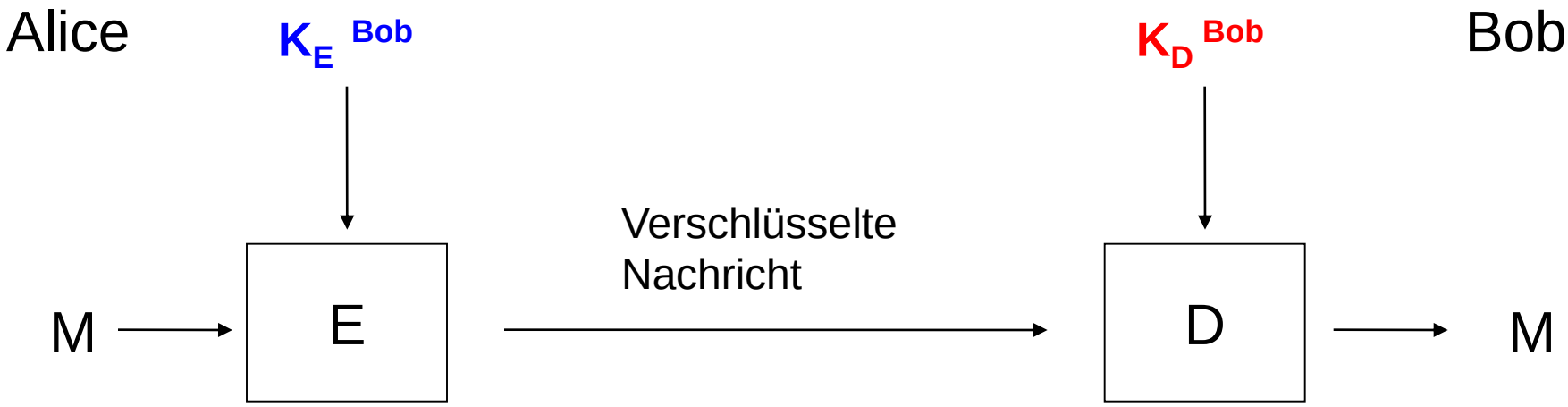


Schlüsselverteilungsproblem

- Wie Schlüssel K_{AB} etablieren, wenn man sich noch nicht kennt?
  - Symmetrische Verschlüsselung setzt einen vertraulichen Informationskanal zur Übertragung des Schlüssels K_{AB} voraus!
    (Problem des Schlüsselaustauschs bzw. Schlüsselvereinbarung)
- Wie viele Schlüssel braucht man für 2, 3, 4, 10, 1000 Teilnehmer, wenn jeder jedem Nachrichten schicken können soll, die kein anderer lesen kann?
  - Für 2, 3, 4, 10, 1000 Teilnehmer werden 1, 3, 6, 45, 499.500 Schlüssel benötigt
- Allgemein: Bei n potentiellen Kommunikationsteilnehmern benötigt man bis zu $n \times (n-1)/2$ Schlüssel.
- n = 10.000 entspricht ungefähr 50.000.000 Schlüsseln!

Asymmetrische (Public Key-) Verschlüsselung

- Whitfield Diffie, Martin E. Hellman (1976): Warum müssen eigentlich Sender und Empfänger denselben Schlüssel benutzen?
- Ein Schlüsselpaar (K_E ,K_D) pro Kommunikationspartner
- Ein öffentlicher und ein privater Schlüssel
- Ein Schlüssel zum Verschlüsseln (K_E), der andere zum Entschlüsseln (K_D)



Allgemeine Eigenschaften asymmetrischer Verfahren

- K_E *sei* der öffentliche, K_D *der* geheime Schlüssel
- Die Schlüsselpaare (K_E, K_D) müssen folgende Eigenschaft erfüllen:

  Für alle Nachrichten *M*: $D\,(E\,(M, K_E), K_D) = M$

  (Eine Nachricht, die mit K_E verschlüsselt wird, wird mit K_D entschlüsselt und umgekehrt)
- Solche Schlüsselpaare müssen leicht zu erzeugen sein.
- Ver- und Entschlüsselungen (E und D) sind effizient durchführbar
- K_D ist aus K_E nicht mit vertretbarem Aufwand berechenbar

Beispiele für asymmetrische Verfahren

- RSA („Quasi-Standard“),
- ElGamal-Verfahren
- Schlüsselvereinbarung nach Diffie und Hellman

Einweg-Funktionen

- Basis für asymmetrische Kryptographie sind Einweg-Funktionen (***one-way***)

  f: X $\rightarrow$ Y

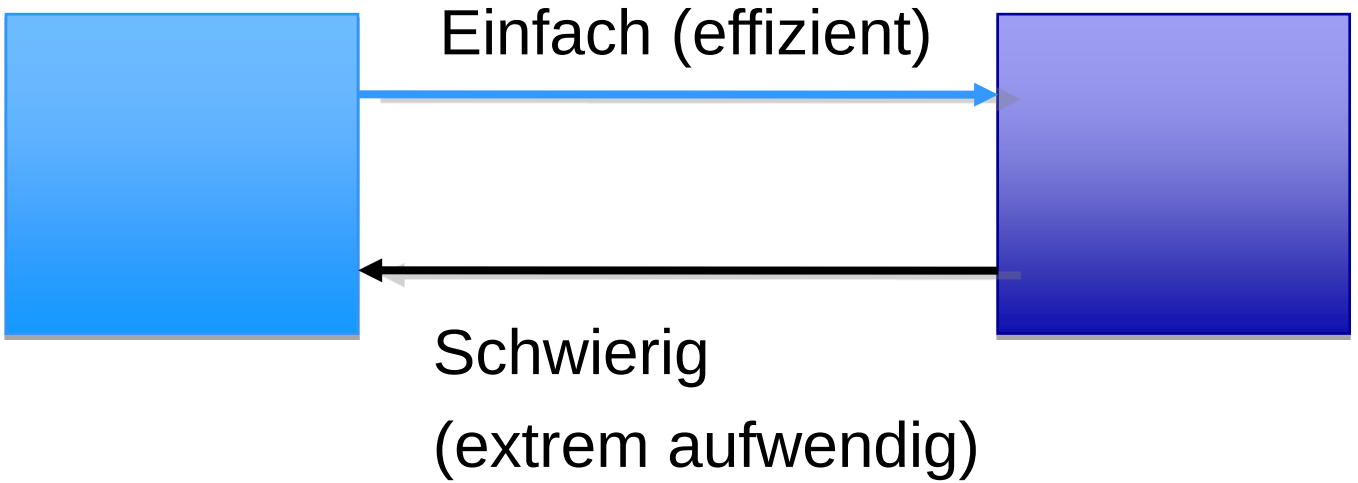
  Eigenschaften von Einweg-Funktionen:

  1) $\forall\, x \in X$ gilt: f(x) ist effizient berechenbar; und

  2) für fast alle $y \in Y$ gilt, dass es nicht effizient möglich ist, das Urbild zu y zu berechnen, also $x \in X$, mit y = f(x)

- Bis heute nicht bewiesen, ob überhaupt Einwegfunktionen existieren.
- Bewiesen ist: Sie existieren gdw. **P != NP**.

Einweg-Funktionen



- „Extrem aufwendig“ = nicht in der Lebenszeit des Universums zu schaffen („unmöglich“)

Einweg-Funktionen mit Falltür

- Einweg-Funktionen mit Falltür (***trapdoor one-way***):
  - mit Zusatzinformation sind Urbilder effizient berechenbar

- Gute Kandidaten für Einweg-Funktionen mit Falltür:
  1) Potenzfunktion modulo n mit n=pq, p und q Primzahlen
     f: $x \rightarrow x^e \bmod n$
  2) Exponentialfunktion modulo p, p Primzahl
     f: $x \rightarrow g^x \bmod p$

- Nachfolgende Folien: Umkehrung der Exponentialfunktion

Primitivwurzel

- Vorausgesetzte Begrifflichkeit:

  Gegeben p Primzahl.
  Eine Primitivwurzel g mod p erzeugt alle von 0 verschiedenen Reste mod p (Generator).

Beispiel: p=7. Dann ist g=5 eine Primitivwurzel mod 7. Denn:

5^1		$= 5 \mod 7$
$5^2 =$	25	$= 4 \mod 7$
$5^3 =$	4x5	$= 6 \mod 7$
$5^4 =$	6x5	$= 2 \mod 7$
$5^5 =$	2x5	$= 3 \mod 7$
$5^6 =$	3x5	$= 1 \mod 7$

Problem des diskreten Logarithmus /1

- Gegeben p Primzahl, g ≤ p−1 primitive Wurzel und ganze Zahl y .

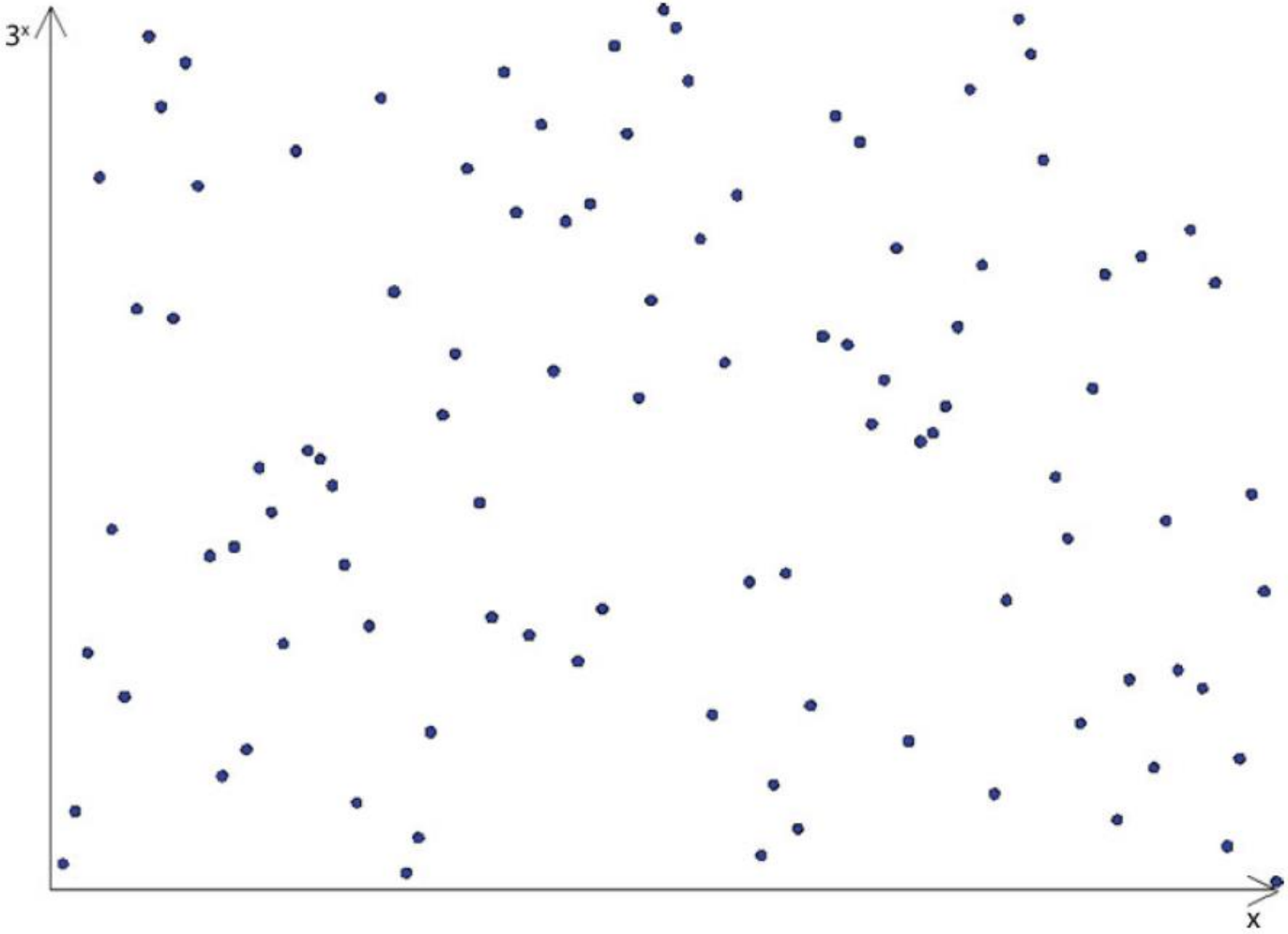
  Bestimme die eindeutige Zahl k (1 ≤ k ≤ p−1) so, dass

  $y = g^k \bmod p$
- Bezeichnung für k: $\mathbf{dlog_g\ y}$, z.B.: $\mathbf{dlog_5\ 4 = 2}$
- Ganzzahliges Problem („diskret“) im Gegensatz zum Begriff aus der Schule
- Diskreter Logarithmus als Umkehrung der diskreten Exponentialfunktion

Problem des diskreten Logarithmus /2

- Berechnung der Exponentialfunktion (auch für große p) effizient durchführbar
- Bestimmung des diskreten Logarithmus aber nicht
  - Nur Inspektion (alles durchprobieren)
  - Problem aus NP
- f: $x \rightarrow g^x \bmod p$ ist eine Einweg-Funktion mit Falltür (x)

Graph der diskreten Exponentialfunktion 3^x mod 101



Aus: Beutelspacher et al., Moderne Verfahren der Kryptographie. 4. Aufl., Vieweg, 2001

Schlüsselvereinbarung nach Diffie und Hellman (1)

- **Ziel:**

  Vereinbarung eines gemeinsamen geheimen Schlüssels, ohne diesen über die Leitung zu schicken (z.B. für eine symmetrische Verschlüsselung)

- **Achtung**:

  Diffie-Hellman allein liefert keine Verschlüsselung, keine Authentisierung der Partner!

  - Einsatz u.a. in TLS, Kerberos, IPsec-Protokollen, PGP, S/MIME
  - Basiert auf dem Problem des diskreten Logarithmus

Schlüsselvereinbarung nach Diffie und Hellman (2)

- **Funktionsweise**
  - Wähle große Primzahl p (allen Teilnehmern bekannt)
  - Wähle einen allen Teilnehmern bekannten Wert g, der primitive Wurzel von p ist

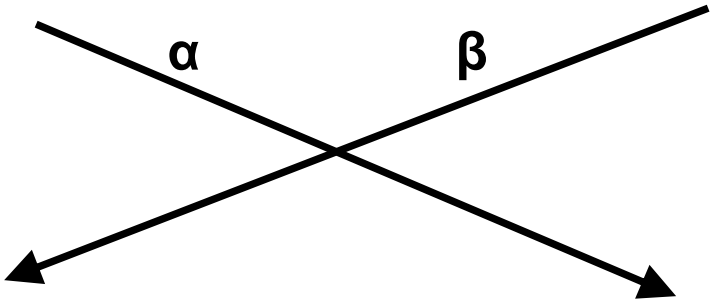
    Es gilt also:

    $\{g^1, ..., g^{p-2}, g^{p-1}\} = \{1, 2, ..., p-2, p-1\}$.
  - g, p öffentlich bekannt

Schlüsselvereinbarung nach Diffie und Hellman (3)

Alice | Bob

Wählt zufällig eine Zahl **a**.
Berechnet $\alpha := g^a \bmod p$.



Wählt zufällig eine Zahl **b**.
Berechnet $\beta := g^b \bmod p$.

Berechnet $\beta^a \bmod p$

Berechnet $\alpha^b \bmod p$

$$\beta^a \bmod p = (g^b)^a \bmod p = g^{ba} \bmod p = g^{ab} \bmod p = (g^a)^b \bmod p = \alpha^b \bmod p =: K_{AB}$$

ElGamal-Verschlüsselungsverfahren

- Taher ElGamal, 1985
- Public-Key-Verfahren, basiert auch auf dem diskreten Logarithmus
- Variante von Diffie-Hellman, ähnliche Idee
- Wir setzen also wieder voraus: p Primzahl, g Primitivwurzel

Funktionsweise: ElGamal-Verschlüsselung (1)

- Schlüsselerzeugung von Bob:
  - Wählt zufällig eine Zahl **b** und berechnet $\beta := g^b \bmod p$.
  - Veröffentlicht **(β, g, p)** als öffentlichen Schlüssel
  - **b** ist der private Schlüssel
- Verschlüsselung einer Nachricht M (mit M < p):
  - Alice schlägt (**β, g, p**) im Schlüsselverzeichnis nach.
  - Alice wählt eine *Nonce* (einmalige Zufallszahl) **a** und berechnet:

    $\alpha := g^a \bmod p$

    $k := \beta^a \bmod p$

    $c := kM \bmod p$ (Maskierung der Nachricht M mit k)

Funktionsweise: ElGamal-Verschlüsselung (2)

- Versenden der Nachricht: Alice verschickt die chiffrierte Nachricht **(α, c)** an Bob
  - **Der Chiffretext ist doppelt so lang wie der Klartext!**

- **Entschlüsselung:**
  1. Bob berechnet zunächst wieder k:

     $\alpha^b \bmod p = (g^a)^b \bmod p = g^{ab} \bmod p = g^{ba} \bmod p = (g^b)^a \bmod p = \beta^a \bmod p = k$
  2. Mittels k kann Bob dann wieder die Nachricht M berechnen:

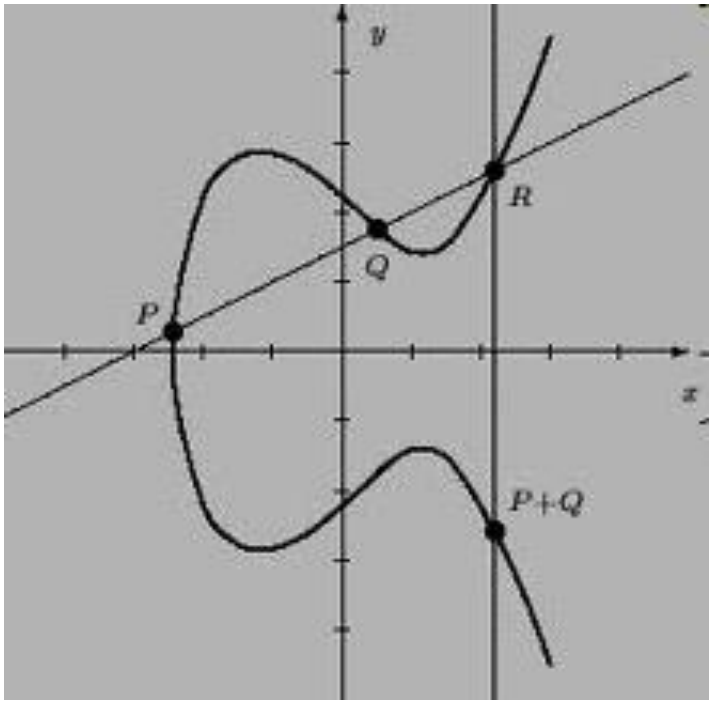
     $k^{-1}\, c \bmod p = k^{-1}\, k\, M \bmod p = 1\, M \bmod p = M$

     (Für Mathematiker: k^{-1} ist das multiplikative Inverse von k mod p)

Asymmetrische Kryptographie auf Basis elliptischer Kurven

- Elliptic Curve Cryptography (ECC), Miller 1985, Koblitz 1987
- Verwendet elliptische Kurven (über Körper K) als algebraische Struktur
  - Für Mathematiker: zyklische Gruppe (erzeugt aus primitivem Element)
  - Spezielle Addition auf dieser Gruppe
  - Beispiel:

    Elemente der Gruppe sind Punkte (x,y) mit der Eigenschaft $y^2 = x^3 + ax + b$

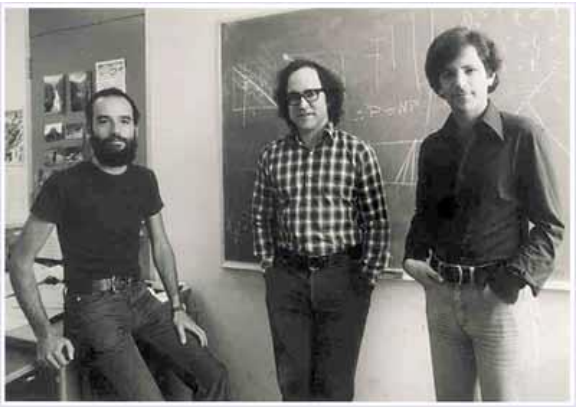


Praktische Bedeutung von ECC

- Problem des diskreten Logarithmus auch auf EC übertragbar
- Implementierung von Diffie-Hellman (ECDH) und ElGamal auf Basis von ECC
- Kleinere Schlüssellängen
  - 224 (256) Bit entspricht 2048 (3072) Bit bei herkömmlicher asymmetrischer Kryptographie
  - Weniger Speicherbedarf und Rechenleistung
  - Deshalb auch Einsatz auf Chipkarten
- Implementierungen in den meisten aktuellen Krypto-Bibliotheken vorhanden wie z.B. OpenSSL, Bouncycastle, Java

RSA-Algorithmus

- **RSA (1977/1978)**:
  Ron Rivest, Adi Shamir, Len Adleman
  - 1983 patentiert, seit 2000 frei
  - Ähnliches System bereits 1973 von Clifford Cocks erfunden, aber geheim gehalten



- **Einsatzbereiche**: Verschlüsseln, Signieren, Schlüsselaustausch
- **Basis**: Primfaktorzerlegung

RSA: Funktionsweise

- Hier: nur skizziert; detaillierte Beschreibung in der Literatur
  1) Wähle Primzahlen *p*, *q*, **Modul *n*** = *pq*
  2) Wähle d so, dass *ggT*((*p* -1)(*q* -1), d) = 1,
  3) Berechne e mit ed = 1 mod ((*p* -1)(*q* -1))
     (typischer Wert: e = 65537 (= 2^{16} + 1))
- (e, *n*) öffentlicher Schlüssel; (d, *n*) geheimer Schlüssel
- Verschlüsselung E: $E(M) = M^{e} \bmod n = C$ (Einwegfunktion: Potenzfunktion)
- Entschlüsselung D: $D(C) = C^{d} \bmod n = M^{ed} \bmod n = M$

  (Falltüren: z.B. Kenntnis von d oder Faktorisierung von p und q)
- Aktuell sichere Größe für n: 2048 Bits, z.T. sicherheitshalber **3072 Bits**

Basis der Sicherheit von RSA: Schwierigkeit der Faktorisierung

- Die Berechnung des Produkts aus zwei großen Primzahlen ist effizient durchführbar

  n = p q
- Die Berechnung von p und q aus n ist dagegen sehr schwierig
- Faktorisierungsproblem ist eines der ältesten Probleme der Zahlentheorie
- Es gibt einige Lösungsverfahren, aber der Berechnungsaufwand ist sehr groß
- Wenn man p und q kennt, dann hat man eine Falltür

Vorgriff: Einsatz von RSA für digitale Signaturen

Verschlüsselung:

- Verschlüsselung E: $E(M) = M^{e} \bmod n = C$
- Entschlüsselung D: $D(C) = C^{d} \bmod n = M^{ed} \bmod n = M$

Digitale Unterschrift:

- Signatur D: $D(M) = M^{d} \bmod n = S$
- Verifikation E: $E(S) = S^{e} \bmod n = M^{de} \bmod n = M$

Hybride Verschlüsselung

- Public-Key-Verfahren wesentlicher ineffizienter (Faktor 1000) als symmetrische Verfahren (z.B. XOR, Shift, Vertauschungsoperationen)
- In der Praxis deshalb häufig in **hybriden Verfahren** eingesetzt:
  - Public-Key-Verfahren: Austausch des geheimen Schlüssels *K* für die symmetrische Verschlüsselung
  - Symmetrisches Verfahren zur effizienten Daten-Verschlüsselung
- Typische Beispiele hierfür:
  - SSL/TLS
  - SSH
  - PGP
  - S/MIME

Ablauf: Hybride Verschlüsselung

- **Beispiel:** vertrauliche Kommunikation zwischen A und B;
  Schlüsselaustausch (mittels Public Key-Verfahren):
  - A: Erzeugt Sitzungsschlüssel K_{AB} für symmetrisches Kryptoverfahren und verschlüsselt diesen mit öffentlichem Schlüssel von B (K_E^B)
  - B: Entschlüsselt mit dem privaten Schlüssel von B (K_D^B)
  - Eigentliche Verschlüsselung zwischen den Partnern A und B mit symmetrischem Verfahren und K_{AB}

Man-in-the-middle-Angriff

- Problem: keine ausreichende Authentisierung bei Public Key-Verfahren
  - Woher weiß Alice, dass der gefundene Public Key wirklich zu Bob gehört?
- Gefahr eines Man-in-the-Middle-Angriffs (besser: Middle-Person-Angriff):
  - Angreifer Mallory gibt sich gegenüber Alice als Bob aus.
  - Angreifer Mallory gibt sich gegenüber Bob als Alice aus.
  - „Problem der Schachgroßmeister“
- Lösungsansätze später in der Vorlesung

Levels of Security

Level (Bits)	Encryption	Hash	RSA, D-H	ECC
80	(3DES)	(SHA-1)	1024	160
112	(3DES)	SHA-224*), SHA-256*)	2048	224
128	AES-128	SHA-256*)	3072	256
192	AES-192	SHA-384*)	7680	384
256	AES-256	SHA-512	15360	521

*) oder SHA-512/nnn

Anwendungen der Kryptographie: E-Mail-Sicherheit

E-Mail-Sicherheit als Anwendung der Kryptographie

- Einsatz kryptographischer Verfahren zur Absicherung von E-Mails zur Erreichung der
  - Vertraulichkeit,
  - Integrität, Authentizität
  - Verbindlichkeit

- Zwei verbreitete Ansätze
  - S/MIME (später)
  - Pretty Good Privacy (PGP)

Sicherheitsprobleme im Zusammenhang mit E-Mails (1)

- **Sicherheitsproblem 1:**
  - E-Mails sind mit Postkarten vergleichbar.
  - E-Mail-Inhalte können von Unbeteiligten gelesen werden.

  Schutzziel: Sicherstellung der Vertraulichkeit

- **Sicherheitsproblem 2:**
  - E-Mails können auf dem Kommunikationsweg verändert werden.

  Schutzziel: Sicherstellung der Integrität der Nachricht

Sicherheitsprobleme im Zusammenhang mit E-Mails (2)

- **Sicherheitsproblem 3:**
  - Der Absender einer E-Mail kann gefälscht werden (s. Postbank-Nachrichten).

  Schutzziel: Authentizität

- **Sicherheitsproblem 4:**
  - Wie kann man nachweisen (auch gegenüber einem neutralen Dritten), dass eine E-Mail wirklich versendet bzw. empfangen worden ist?

  Schutzziel: Nicht-Abstreitbarkeit, Verbindlichkeit

PGP – Pretty Good Privacy

- Entwickelt von Phil Zimmermann, 1991
- Bietet Authentisierung, Verschlüsselung, digitale Signaturen für E-Mails:
  - Schutzziele:
    Vertraulichkeit, Integrität und Authentizität
- Auch Dateiverschlüsselung möglich
- OpenPGP-Standard, RFC 4880

PGP – Implementierungen

- Freie Software und kommerzielle Produkte, z.B.:
  - Lizenzierte Fassung (für 50$):

  http://www.pgp.com (jetzt von Symantec übernommen)
  - Gpg4win:

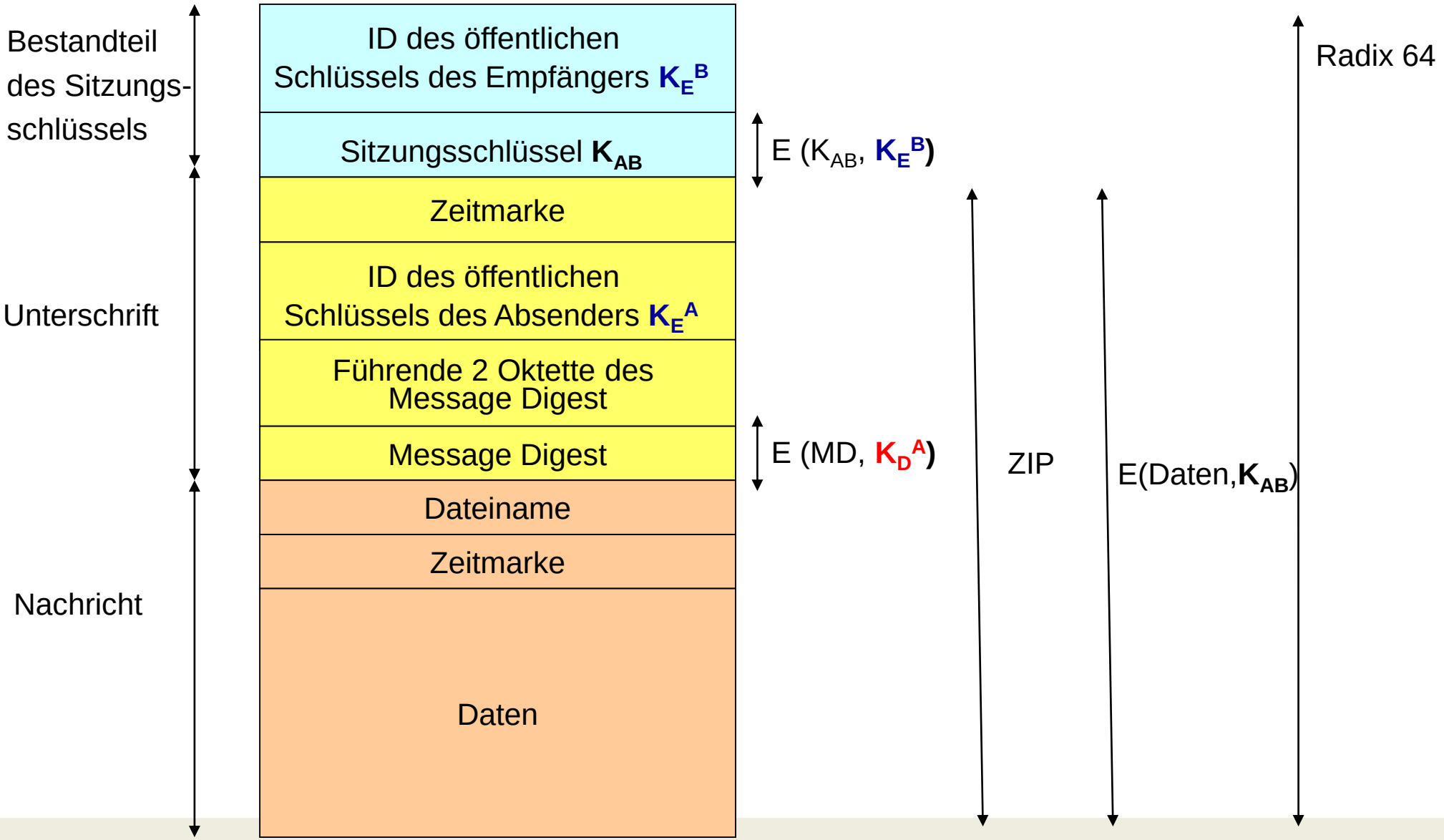
  https://www.bsi.bund.de/DE/Themen/Kryptografie_Kryptotechnologie/Kryptotechnologie/Gpg4win/gpg4win_node.html
  - Enigmail:

  https://enigmail.net/index.php/en/
- Plug-Ins für gängige E-Mail-Clients wie z.B. Outlook, Thunderbird
- PGP ist vor allem für Privatnutzer geeignet

Sicherheitsdienste von PGP

- Einsatz von gängigen Verfahren für Verschlüsselung, digitale Signatur und Authentisierung
- Hybride Verschlüsselung
  - Public Key-Verfahren zum Schutz des Schlüssels für die symmetrische Verschlüsselung
  - Eigentliche Verschlüsselung symmetrisch
- Sicherung des privaten Schlüssels durch Passphrase (→Trojanisches Pferd)
- Kryptographische Verfahren:
  - RSA für asymmetrische Verschlüsselung
  - IDEA, 3DES, CAST, nun auch AES für symmetrische Verschlüsselung
  - Hashverfahren: z.B. SHA 256
  - U.a. RSA für digitale Signatur, DSA

Format einer PGP-Nachricht (von A nach B)

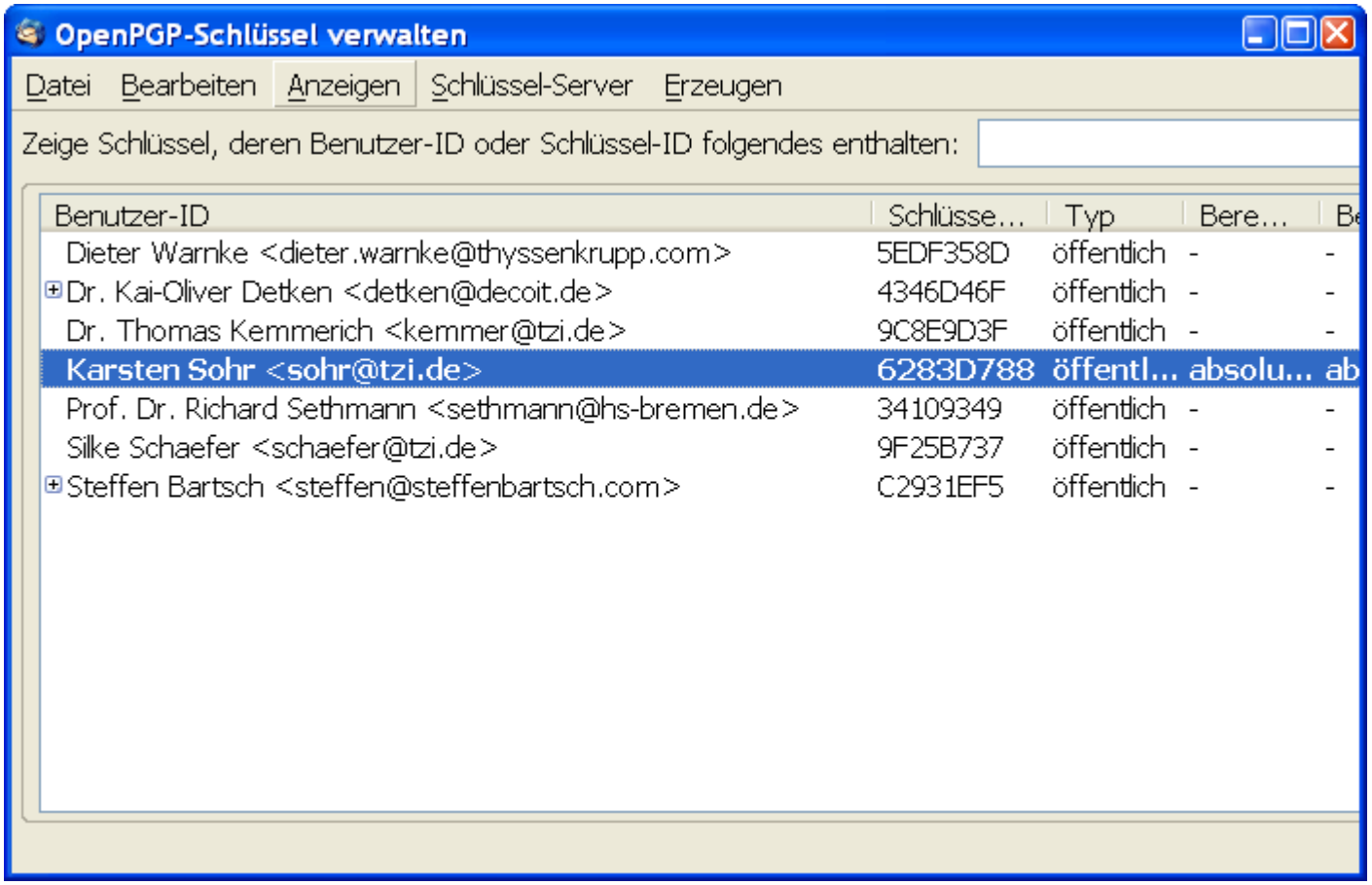


Schlüsselverteilung in PGP

- Woher bekomme ich den öffentlichen Schlüssel des Kommunikationspartners?
  - Import von einem Schlüsselserver wie z.B.: http://wwwkeys.de.pgp.net/
  - Von einem anderen Kommunikationspartner per E-Mail
  - …
- Erzeugung eines Schlüsselbundes mit den öffentlichen Schlüsseln der Kommunikationspartner

- **Problem**: Woher weiß ich, dass es sich wirklich um den öffentlichen Schlüssel des Kommunikationspartners handelt?
  - Auf den Schlüsselservern gibt es öffentliche Schlüssel wie z.B. merkel@bundesregierung.de.

Beispiel für einen Schlüsselbund

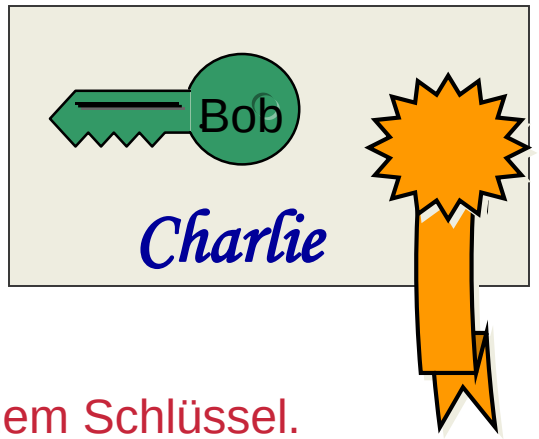


Web of Trust (1)

- Prinzip zur sicheren Verteilung der öffentlichen Schlüssel in PGP:
  - Kommunikationsteilnehmer **zertifizieren** (signieren) sich gegenseitig öffentliche Schlüssel (***Web of Trust, Vertrauensgeflecht***)
  - Einführung eines öffentlichen Schlüssels durch einen Dritten (z.B. gemeinsamer Bekannter, c't-Kampagne, …)
  - Wenn man diesem Dritten vertraut, dann vertraut man auch öffentlichen Schlüsseln, die der Dritte unterzeichnet hat.
  - Signieren eines Schlüssels durch mehrere Kommunikationspartner möglich
- PGP-Implementierungen (wie GnuPP) stellen Software zum Signieren von Schlüsseln zur Verfügung

Web of Trust (2)

- **Beispiel**:
  - Alice will Bob eine verschlüsselte E-Mail senden.
  - Alice hat Bobs öffentlichen Schlüssel nicht im Schlüsselbund.
  - Alice kann den Schlüssel von einem Schlüsselserver importieren.
  - Charlie hat Bobs öffentlichen Schlüssel signiert, und Alice vertraut Charlie vollständig.



  - Also vertraut Alice Bobs öffentlichem Schlüssel.

Vertrauensstufen für öffentliche Schlüssel in PGP

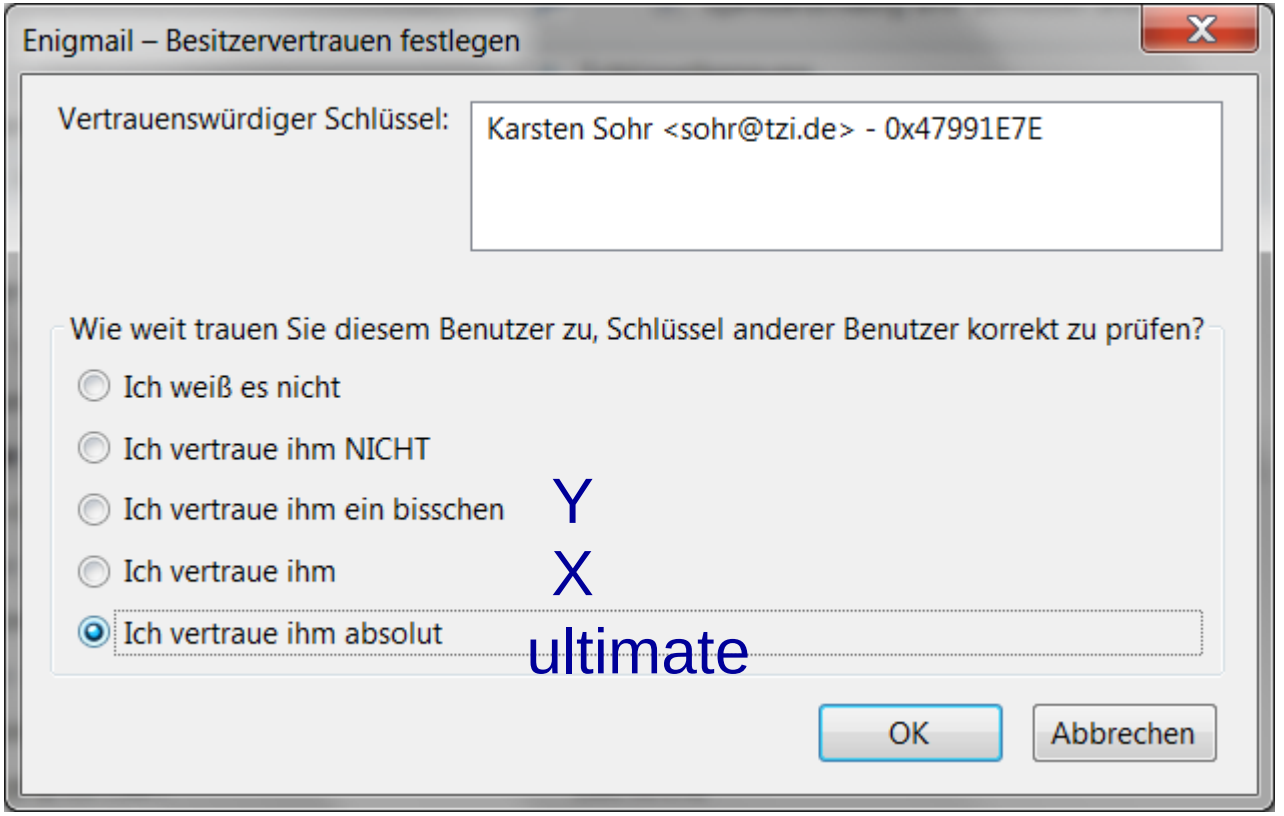
- Vertrauensstufen für öffentliche Schlüssel

1. Undefiniertes Schlüsselvertrauen
2. Schlüsseleigentum nicht gesichert
3. Teilweises Vertrauen in den Schlüsselbesitz
4. Volles Vertrauen in den Schlüsselbesitz

Berechnung von Vertrauen in einen öffentlichen Schlüssel

- PGP berechnet das Vertrauen in den öffentlichen Schlüssel mittels einer gewichteten Summe aus dem Vertrauen in die Signaturen:
  - Forderung für volles Vertrauen in Schlüsselbesitz: gewichtete Summe ≥ 1
  - Absolut („ultimate“) vertrauenswürdige Signaturen haben ein Gewicht von 1
  - Immer vertrauenswürdige Signaturen haben ein Gewicht von 1/X,
  - Gewöhnlich vertrauenswürdige Signaturen haben ein Gewicht von 1/Y
  - X, Y konfigurierbare Parameter: Im Falle von X=2, Y=4 würden z.B. eine immer vertrauenswürdige Signatur und zwei Signaturen mit gewöhnlichem Vertrauen ausreichen

Schlüsselvertrauen: Enigmail-Dialogfenster



Rücknahme von öffentlichen Schlüsseln

- Manchmal wird ein eigener privater Schlüssel gebrochen.
- Oder: Trojanisches Pferd liest den Schlüssel aus dem Speicher!
- Oder: Der Schlüssel war zu lange in Gebrauch.

- Rücknahme des Schlüsselpaares in PGP möglich.

- Vorgehensweise:
  - Veröffentlichen eines signierten Rücknahmezertifikates
  - Rücknahmezertifikat wird durch privaten Schlüssel des zurückzunehmenden öffentlichen Schlüssels unterschrieben.
  - Letzte Aktion dieses Schlüssels