Linux User Group Tübingen

Public-Key Kryptographie

theoretische Grundlagen und praktische Anwendung mit GNU Privacy Guard und KDE

Jan Petránek <jan@petranek.de>

Linux User Group Tübingen

Jan Petránek

Übersicht

- Motivation
- Symmetrische Verschlüsselung
- Asymmetrische Verschlüsselung
  - Schlüsselaustausch
  - Kommunikation
  - Digitale Signatur
  - Man-in-the-middle-Angriffe
  - Authentizität öffentlicher Schlüssel
- GNU Privacy Guard unter KDE
  - graphische Userinterfaces: kgpg, kmail
- Die Mathematik dahinter
  - inverse Elemente in komischen Zahlenräumen
  - RSA-Verfahren

Linux User Group Tübingen

Jan Petránek

Diplomatengepäck in E-Mail

- EU schützt interne elektronische Kommunikation mit der EU-Vertretung in Ankara durch...?

Linux User Group Tübingen

Jan Petránek

Diplomatengepäck in E-Mail

- EU schützt interne elektronische Kommunikation mit der EU-Vertretung in Ankara durch...?

  Internationales Recht.

Linux User Group Tübingen

Jan Petránek

Diplomatengepäck in E-Mail

- EU schützt interne elektronische Kommunikation mit der EU-Vertretung in Ankara durch...?

  Internationales Recht.
- Februar 2002: Extreme türkische Gruppierungen spionieren diesen Verkehr aus und veröffentlichen das Material.

Linux User Group Tübingen

Jan Petránek

Diplomatengepäck in E-Mail

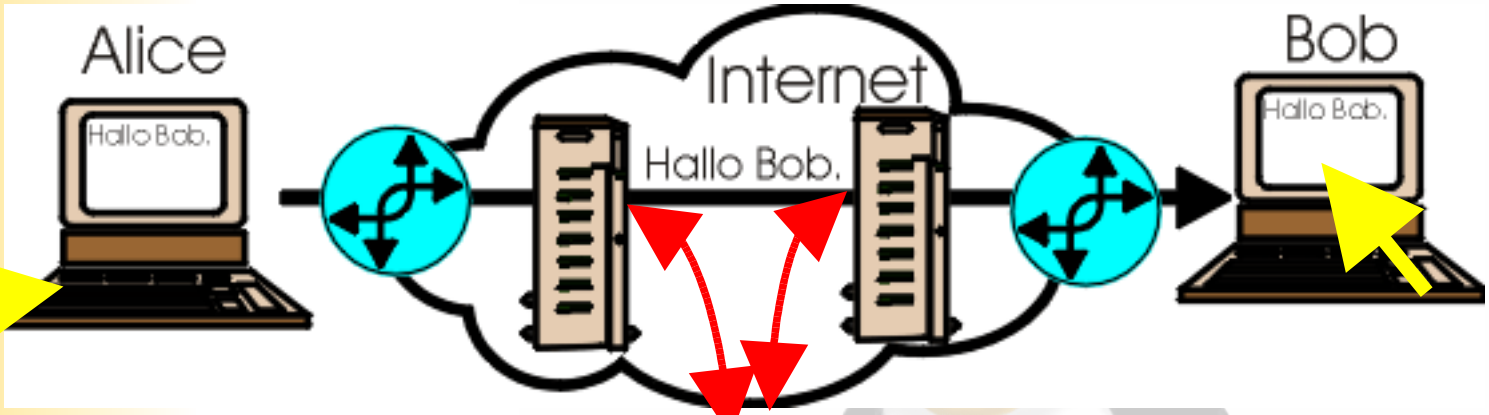
- EU schützt interne elektronische Kommunikation mit der EU-Vertretung in Ankara durch...?

  Internationales Recht.
- Februar 2002: Extreme türkische Gruppierungen spionieren diesen Verkehr aus und veröffentlichen das Material.
- 19.2.2002 EU-Kommissar Günter Verheugen bestellt den türkischen Botschafter bei der EU, Nihat Akyol zu sich.
- Am 20.2.2002 entschuldigt sich der türkische Premierminister Bülent Ecevit bei EU-Kommissionschef Romano Prodi.

Linux User Group Tübingen

Jan Petránek

E-Mail und Sicherheit

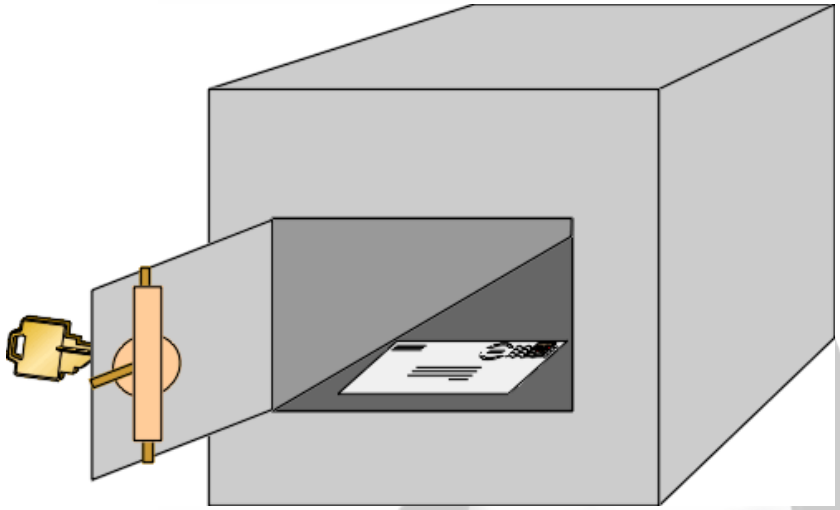


- Sicherheit bei Kommunikation:
  - Vertraulichkeit: Nur Alice und Bob kennen den Inhalt
  - Authentizität: Alice ist sicher die Absenderin
  - Integrität: Die Nachricht kommt unverfälscht an
- E-Mail ≠ Briefpost!
- Angriffspunkte bei unverschlüsselter E-Mail:
  - am Transport beteiligte Stationen
  - Client-Rechner

Linux User Group Tübingen

Jan Petránek

Symmetrische Verschlüsselung: ein digitaler Tresor

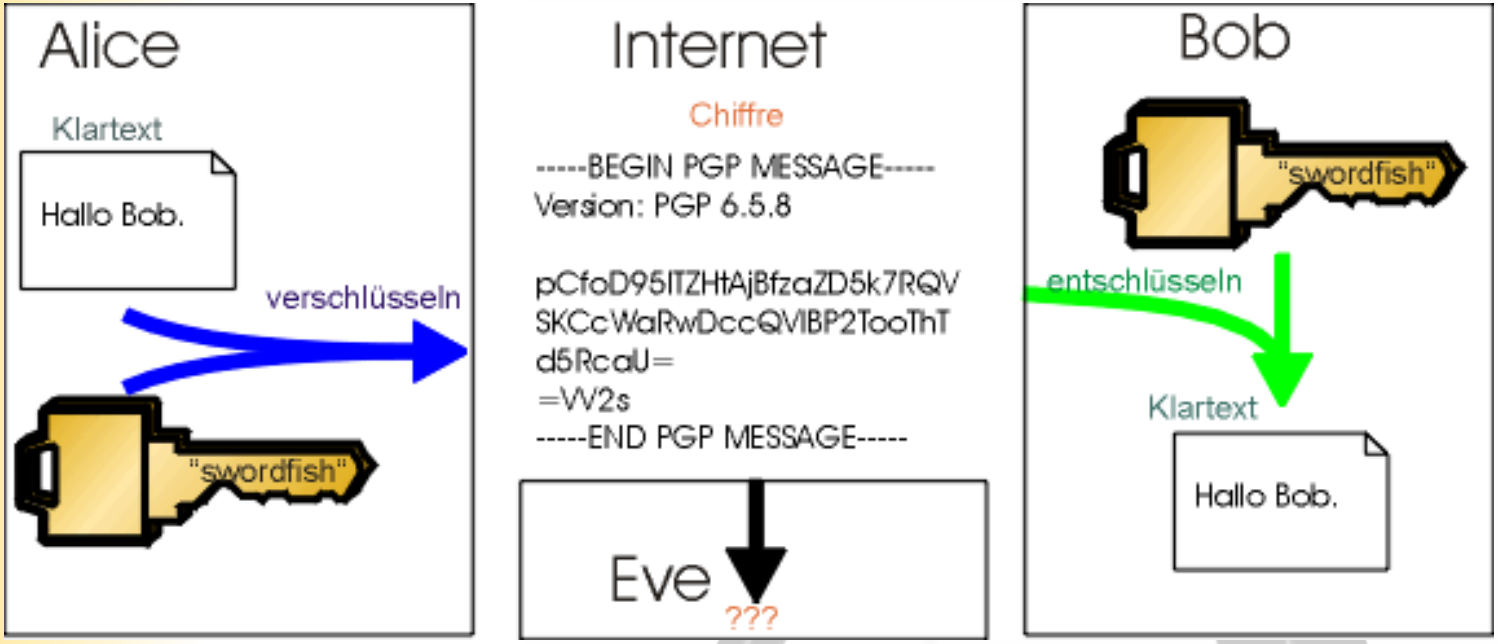


- Datei wird mit einem Schlüssel verschlüsselt
- mit demselben Schlüssel wird die Datei wieder entschlüsselt
- weil beidesmal derselbe Schlüssel verwendet wird, heißen solche Verfahren symmetrisch

Linux User Group Tübingen

Jan Petránek

Symmetrische Verschlüsselung: Kommunikation

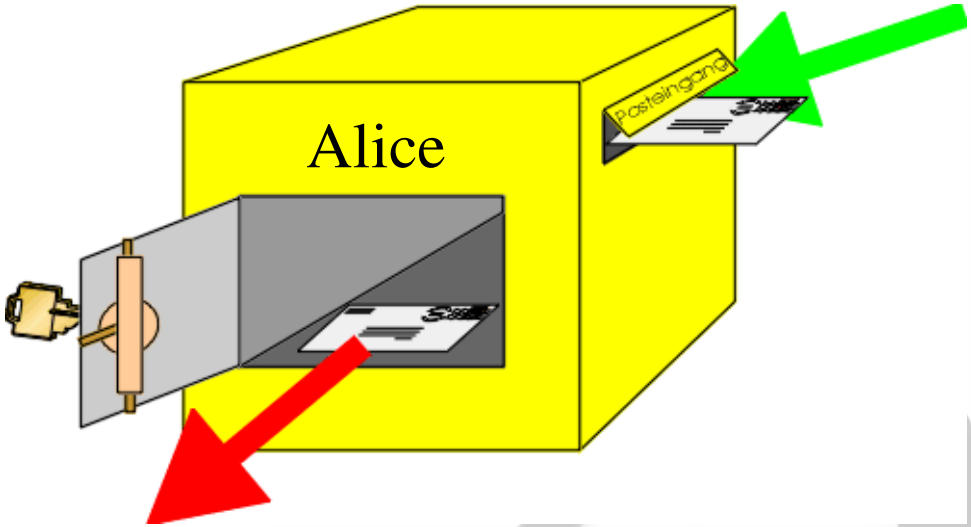


- Lauscher kann mit Chiffre nichts anfangen
- Wie gibt Alice den Schlüssel an Bob?
  - Schlüsselaustausch benötigt sicheren Kanal
  - nicht per E-Mail

Linux User Group Tübingen

Jan Petránek

Asymmetrische Verschlüsselung: digitaler Briefkasten statt Tresor

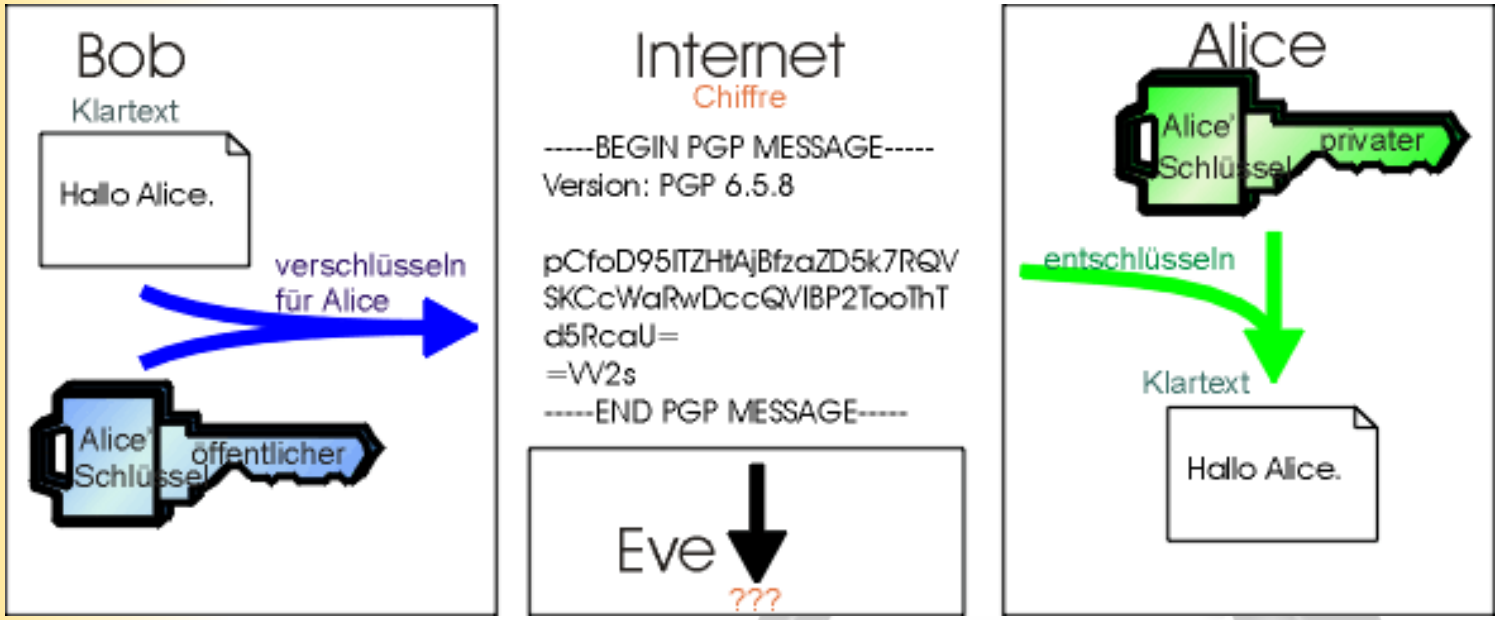


- Jeder kann eine Nachricht für Alice einwerfen (= öffentlicher Schlüssel)
- nur Alice kann mit ihrem Briefkastenschlüssel (= privater Schlüssel) ihre Post holen
- unterschiedliche Schlüssel für Ver-und Entschlüsselung, daher “asymmetrisches Verfahren“

Linux User Group Tübingen

Jan Petránek

Asymmetrische Verschlüsselung: Kommunikation



- Bob verschlüsselt mit Alice' öffentlichem Schlüssel
- nur Alice kann die verschlüsselte Nachricht mit ihrem privaten Schlüssel entschlüsseln
- selbst, wenn Eve Alice' öffentlichen Schlüssel kennt, kann sie die Nachricht nicht entschlüsseln

Linux User Group Tübingen

Jan Petránek

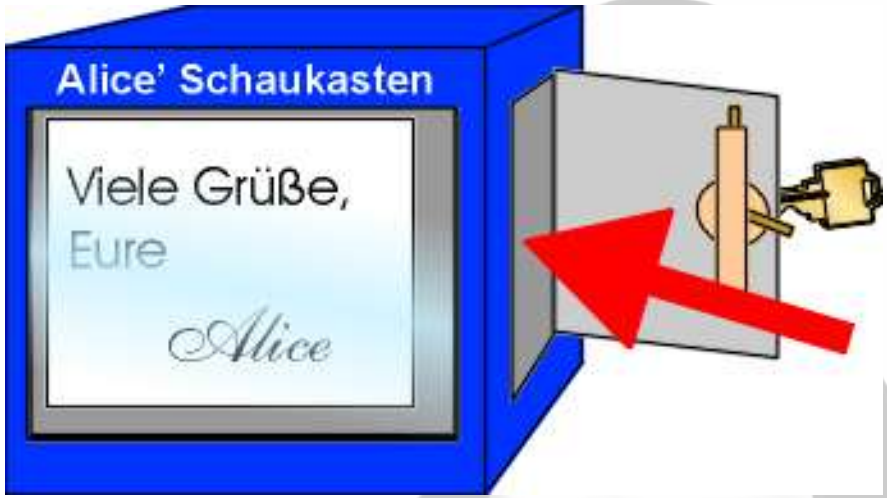
Asymmetrische Verschlüsselung: Schlüsselaustausch

- Alice erzeugt ihr Schlüsselpaar (= Briefkasten)
- Alice' öffentlicher Schlüssel (= Briefkastenschlitz):
  - dient der Verschlüsselung an Alice
- Alice' privater Schlüssel (= Briefkastenschlüssel):
  - entschlüsselt Nachrichten an Alice
  - Alice hält ihn geheim! (wird auf Festplatte verschlüsselt)
- Schlüsselaustausch
  - Alice schickt ihren öffentlichen Schlüssel an Bob
  - kein sicherer Kanal nötig

Linux User Group Tübingen

Jan Petránek

Digitale Signatur: noch ein Schlüsselpaar

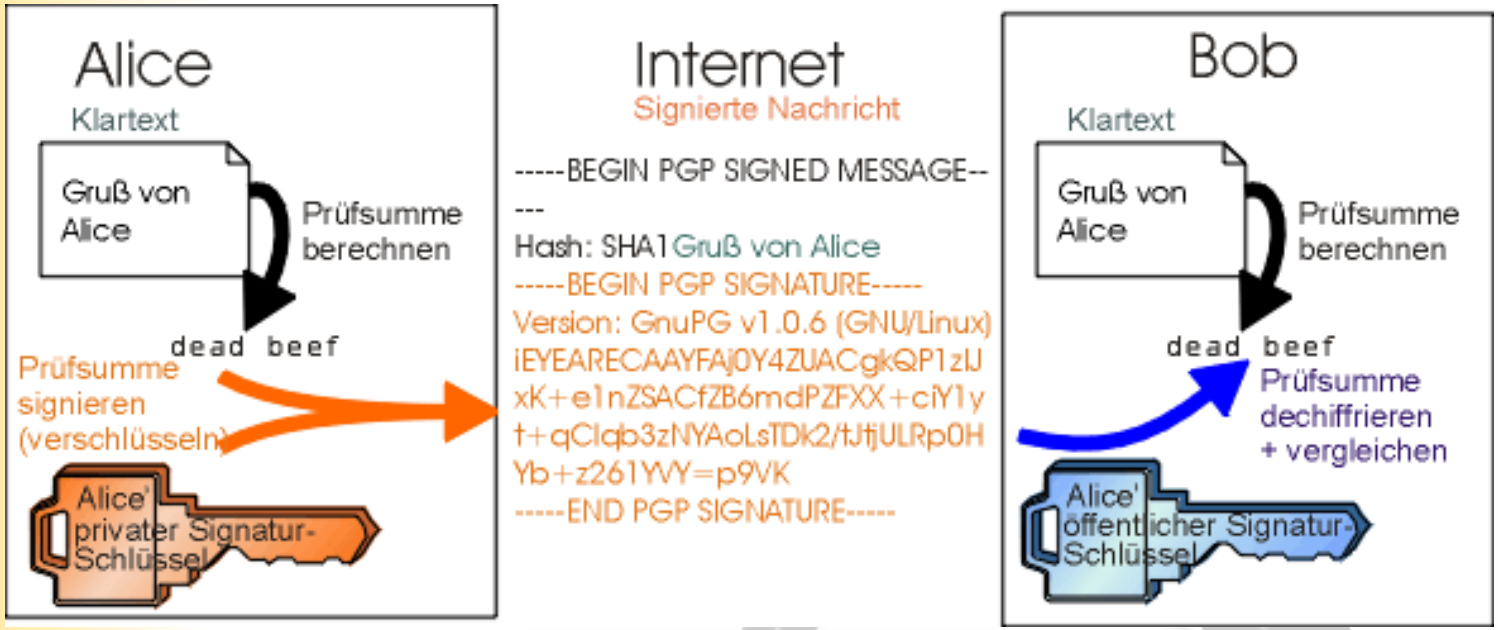


- Alice erzeugt ein Schlüsselpaar zur Signatur
- Alice behält den Verschlüsselungsschlüssel (privater Signaturschlüssel)
- Alice veröffentlicht den Entschlüsselungsschlüssel (öffentlicher Signaturschlüssel)

Linux User Group Tübingen

Jan Petránek

Digitale Signatur: Überprüfung



- Alice errechnet Prüfsumme der Nachricht
  - berechnet und verschlüsselt Prüfsumme mit ihrem privatem Signaturschlüssel
  - nur privater Signaturschlüssel erzeugt gültige Signatur
- Bob entschlüsselt Prüfsumme und überpüft, ob sie zur Nachricht paßt

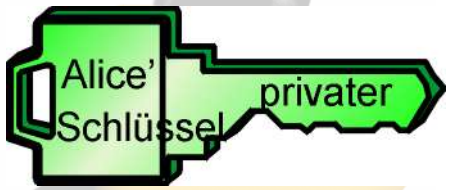
Linux User Group Tübingen

Jan Petránek

Private und öffentliche Schlüssel
Alice' Schlüsselbund

Zweck \ Nutzer	öffentlich	privat
Verschlüsselung	verschlüsselt an Alice	entschlüsselt an Alice verschlüsselte Nachrichten
Signatur	überprüft Alice' Signatur	erzeugt passende Signatur

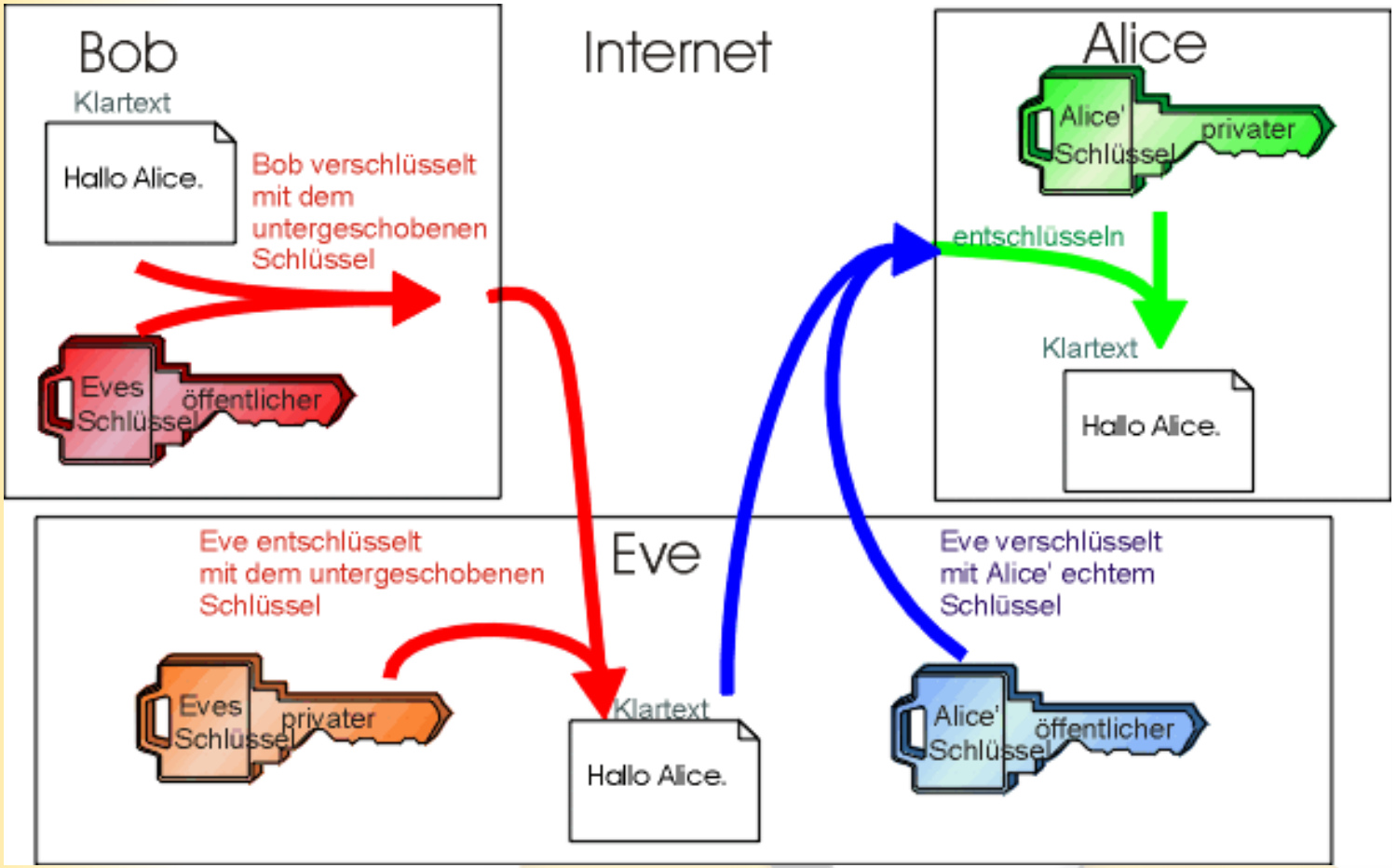
als „öffentlicher Schlüssel“ werden der Einfachheit halber alle öffentlichen Schlüssel bezeichnet



Linux User Group Tübingen

Jan Petránek

Man-in-the-middle-Angriff



- Untergeschobene Schlüssel erkennen
- Authentizität des Schlüssels sichern

Linux User Group Tübingen

Jan Petránek

Schlüssel überprüfen

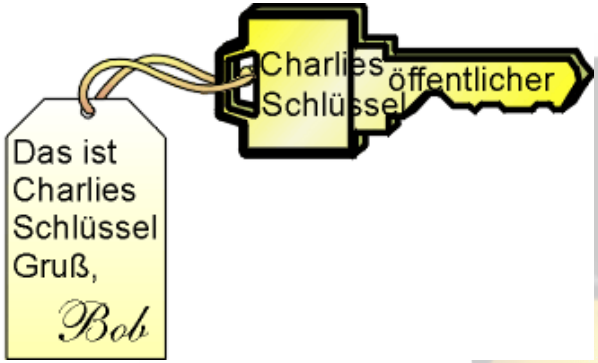
```
pub  1024R/E1AF29F1 2002-02-18 BoB
<Bob@mail.petranek.de>
Key fingerprint = 63 67 55 D0 73 3F B1 DC  2F 56 02
AF 15 FF 3A 30
```

- Echtheit des Schlüssels direkt verifizieren
  - persönlicher Schlüsselaustausch
  - Fingerprint vergleichen, z. B. via Telefon oder Visitenkarte

Linux User Group Tübingen

Jan Petránek

Signierte Schlüssel

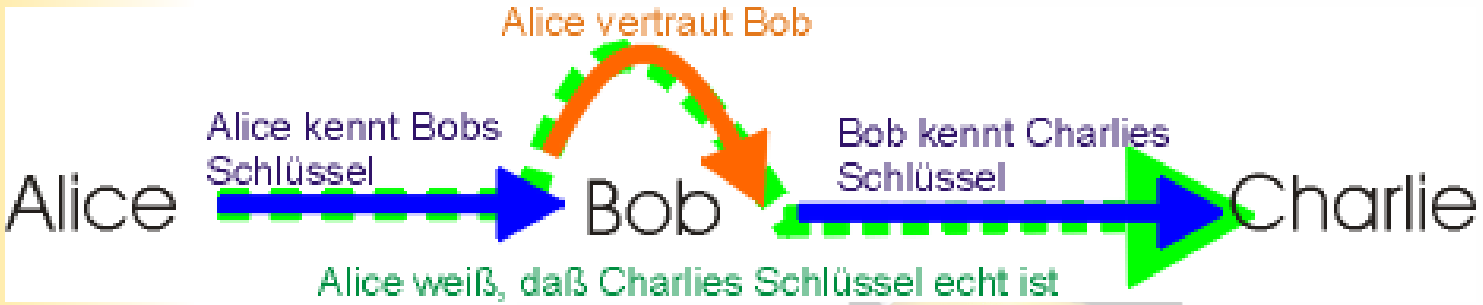


- Bob kennt Charlies Schlüssel
- Bob unterschreibt, daß Charlies Schlüssel echt ist
- Alice kann Charlies Schlüssel indirekt überprüfen:
  - Alice kennt Bobs Schlüssel
  - Alice weiß nun, daß Bob Charlies Schlüssel signiert hat

Linux User Group Tübingen

Jan Petránek

Vertrauensfrage



- Alice weiß, daß Bob Charlies Schlüssel signiert hat
- vertraut Alice Bob?
- traut sie Bob zu, daß er Charlies Identität sorgfältig geprüft hat?
- Vertrauenskette somit lückenlos
- Alice erkennt Charlies Schlüssel als echt an

Linux User Group Tübingen

Jan Petránek

GNU Privacy Guard unter KDE
Schlüssel erzeugen

Linux User Group Tübingen

Jan Petránek

GNU Privacy Guard unter KDE
Schlüsselverwaltung

- Alice hat Bobs Schlüssel signiert
- Alice vertraut Bob
- Bob hat Charlies Schlüssel signiert

GNU Privacy Guard unter KDE
E-Mail verschlüsseln und signieren

Linux User Group Tübingen

Jan Petránek

- **Motivation**
- **Symmetrische Verschlüsselung**
- **Asymmetrische Verschlüsselung**
- Schlüsselaus-tausch
- Kommunikation
- Digitale Signatur
- Man-in-the-middle Angriffe
- authentische Schlüssel
- **GPG unter KDE**
- kgpg
- kmail
- **Mathematik**
- Inverse Elemente
- RSA-Verfahren

Linux User Group Tübingen

Jan Petránek

- **Motivation**
- **Symmetrische Verschlüsselung**
- **Asymmetrische Verschlüsselung**
- Schlüsselaus-tausch
- Kommunikation
- Digitale Signatur
- Man-in-the-middle Angriffe
- authentische Schlüssel
- **GPG unter KDE**
- kgpg
- kmail
- **Mathematik**
- Inverse Elemente
- RSA-Verfahren

GNU Privacy Guard unter KDE
E-Mail lesen

Verschluesselte und signierte Nachricht

Von: Jan Petranek <jan.petranek@student.uni-tuebingen.de> (Uni Tübingen)

An: jan.petranek@student.uni-tuebingen.de

Datum: Heute 15:07:09

Verschlüsselte Nachricht

Nachricht enthält Signatur von Jan Petranek (Uni-Mail) (Schlüssel-ID: 0x7466AAE9).
Die Signatur ist gültig, und der Schlüssel ist vollständig vertrauenswürdig.

Hallo ich,

ich sende Dir eine geheime Botschaft.

Pssst. Geheim!

JanP

Ende der signierten Nachricht

Ende der verschlüsselten Nachricht

Linux User Group Tübingen

Jan Petránek

Die Mathematik dahinter inverse Elemente

- Wie teilt man einen Schlüssel auf?
- nach 2 Operationen soll wieder dasselbe rauskommen
- Inverse Elemente (Umkehroperationen)
- Addition $4 + 3 = 9$ $9 + (-3) = 4$
- Mutliplikation $5 * 7 = 35$ $35 * \left(\frac{1}{7}\right) = 5$
- Ungeeignet für Kryptographie, da das inverse Element leicht zu errechnen ist

Linux User Group Tübingen

Jan Petránek

Die Mathematik dahinter komische Zahlenräume

- Restring Z_n (Kilometerzähler)

$$a \bmod n = \begin{cases} a \text{ für } a < n \\ a - n \bmod n \text{ sonst} \end{cases}$$

- Beispiel:

$$23 \bmod 8 = 15 \bmod 8 = 7 \bmod 8$$

- Wähle zwei große Primzahlen p und q
- Rechne in Restring Z_n mit n=p*q
- Wenn p und q bekannt sind, ist es einfach, ein Paar e und d zu finden, sodaß d das inverse Element zu e ist:

$$x^{(e*d)} \bmod n = x^1 \bmod n = x \bmod n$$

Linux User Group Tübingen

Jan Petránek

Die Mathematik dahinter
Das RSA-Verfahren

- Alice erzeugt Schlüsselpaar e,d passend zu n
- privater Schlüssel d,n
- Bob bekommt den öffentlichen Schlüssel e,n
- Bob berechnet

$$c = m^e \bmod n$$

- Bob schickt c an Alice
- Alice berechnet

$$c^d \bmod n = \left(m^e \bmod n\right)^d \bmod n = m^{(e*d)} \bmod n = m \bmod n$$

Linux User Group Tübingen

Jan Petránek

Die Mathematik dahinter ein Beispiel

- http://www-fs.informatik.uni-tuebingen.de/~reinhard/krypto/index.html

Linux User Group Tübingen

Jan Petránek

Zusammenfassung

- Motivation
- Symmetrische Verschlüsselung
- Asymmetrische Verschlüsselung
  - Schlüsselaustausch
  - Kommunikation
  - Digitale Signatur
  - Man-in-the-middle-Angriffe
  - Authentizität öffentlicher Schlüssel
- GNU Privacy Guard unter KDE
  - graphische Userinterfaces: kgpg, kmail
- Die Mathematik dahinter
  - inverse Element in komischen Zahlenräumen
  - RSA-Verfahren