# cryptocampagne 2010

Justus und Jan

Fachbereich Informatik
Universität Hamburg

Kunterbuntes Seminar - 9. Dezember 2010

# Motivation

## Wozu das Ganze?

- Kommunikation unter sich
- Freiheit
- Sicherstellen von Informationsquellen
- Terroristen unterstützen?
- Kryptografie als interessantes Gebiet in der Informatik
- geleakte Diplomaten-Depeschen sicher verwahren
- Weils geht

# Motivation

## Wozu das Ganze?

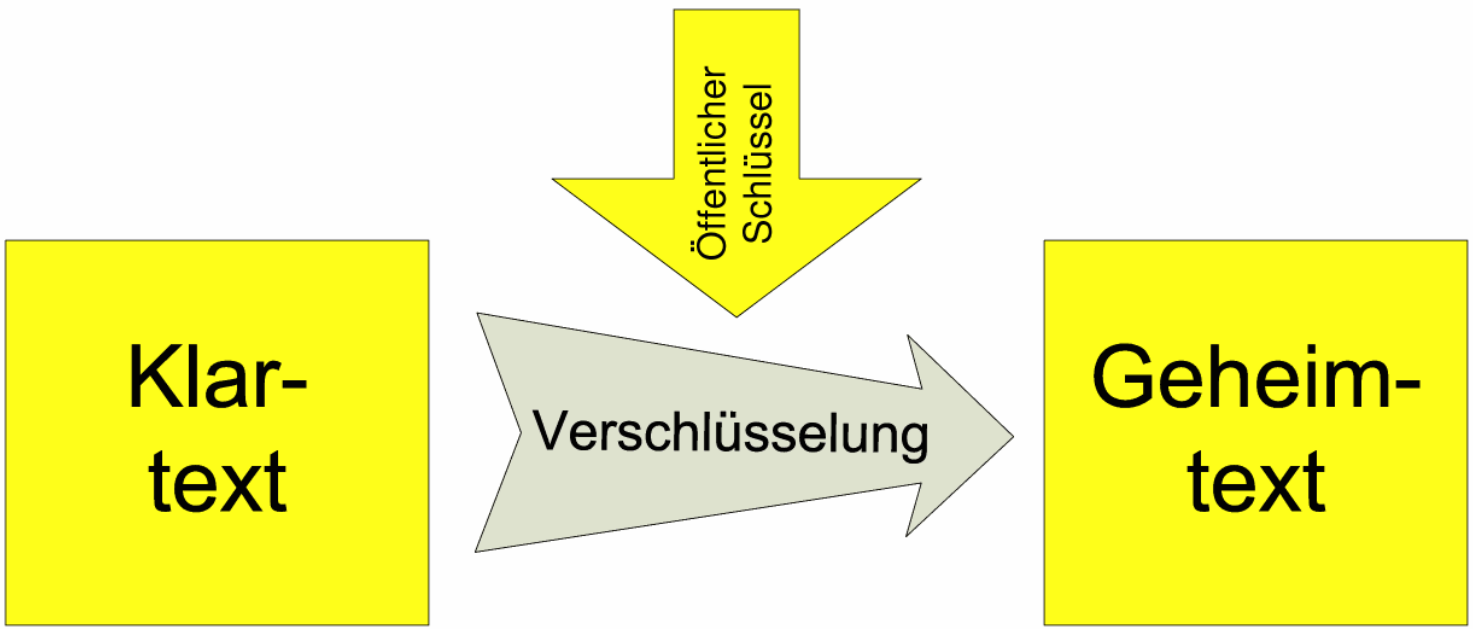
- Kommunikation unter sich
- Freiheit
- Sicherstellen von Informationsquellen
- Terroristen unterstützen?
- Kryptografie als interessantes Gebiet in der Informatik
- geleakte Diplomaten-Depeschen sicher verwahren
- Weils geht

# Einführung

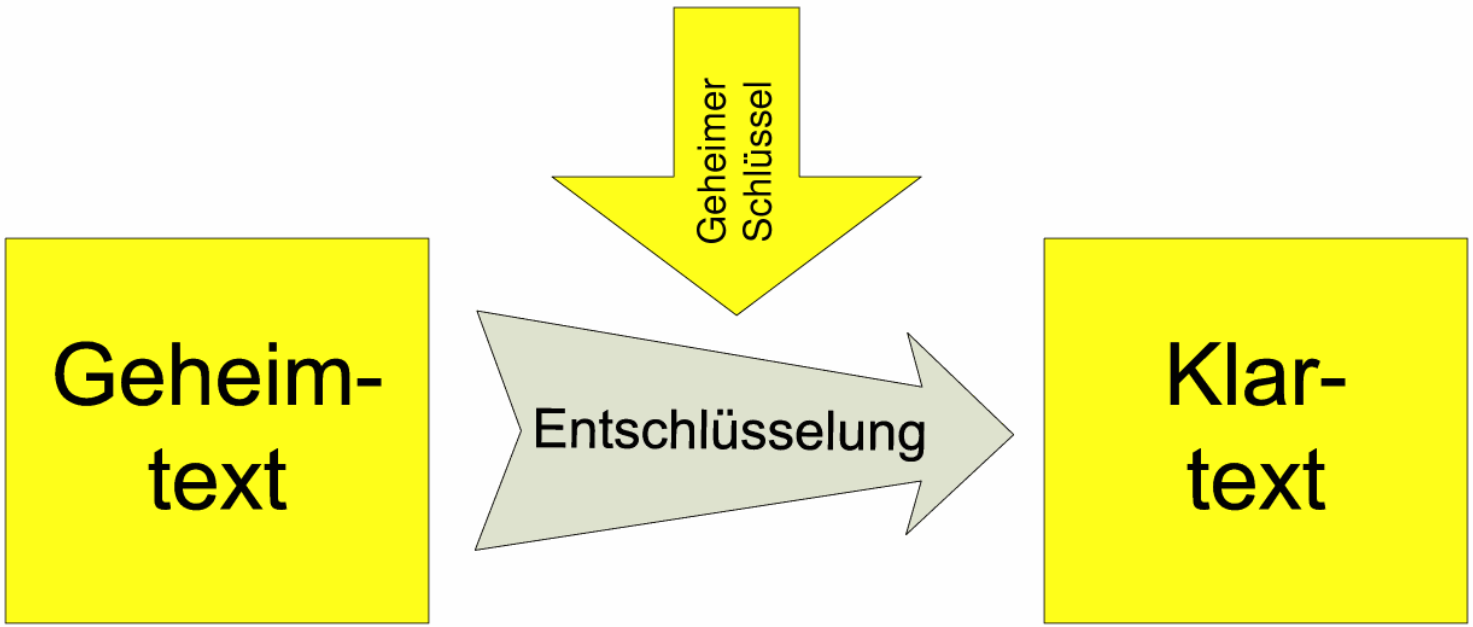
Public Key Cryptography

- sind Cryptoverfahren
- **verschlüsseln** und **signieren** Texte
- arbeiten mit **zwei** Schlüsseln, $k_{private}$ und $k_{public}$
- andere Eigenschaften als symmetrische Verfahren

# Verschlüsselung



# Entschlüsselung



# Die Schlüssel

- Public Key
  - wird veröffentlicht
  - verschlüsseln von Daten und verifizieren von Signaturen
  - ist beispielsweise über Keyserver zu beziehen
- Private Key
  - wird geheimgehalten
  - entschlüsseln von Daten und signieren
- stehen in Beziehung, jedoch soll es nicht möglich sein, von dem einen auf den anderen zu schließen

# Die Schlüssel

- Public Key
  - wird veröffentlicht
  - verschlüsseln von Daten und verifizieren von Signaturen
  - ist beispielsweise über Keyserver zu beziehen
- Private Key
  - wird geheimgehalten
  - entschlüsseln von Daten und signieren
- stehen in Beziehung, jedoch soll es nicht möglich sein, von dem einen auf den anderen zu schließen

# Die Schlüssel

- Public Key
  - wird veröffentlicht
  - verschlüsseln von Daten und verifizieren von Signaturen
  - ist beispielsweise über Keyserver zu beziehen
- Private Key
  - wird geheimgehalten
  - entschlüsseln von Daten und signieren
- stehen in Beziehung, jedoch soll es nicht möglich sein, von dem einen auf den anderen zu schließen

# Vergleich zu symmetrischen Verfahren

## Vorteile

- im Vergleich ist das Geheimnis klein
- Verteilungsproblem: die Public Keys können einfach ausgetauscht werden
- Public Keys von dritten müssen nicht geschützt werden

## Nachteile

- langsam → hybride Verfahren
- man-in-the-middle, Schlüssel können vorgetäuscht werden → **Web of Trust**
- kurze Schlüssel sind nicht sicher → Schlüssellänge bewusst wählen

# Vergleich zu symmetrischen Verfahren

## Vorteile

- im Vergleich ist das Geheimnis klein
- Verteilungsproblem: die Public Keys können einfach ausgetauscht werden
- Public Keys von dritten müssen nicht geschützt werden

## Nachteile

- langsam $\rightarrow$ hybride Verfahren
- man-in-the-middle, Schlüssel können vorgetäuscht werden $\rightarrow$ **Web of Trust**
- kurze Schlüssel sind nicht sicher $\rightarrow$ Schlüssellänge bewusst wählen

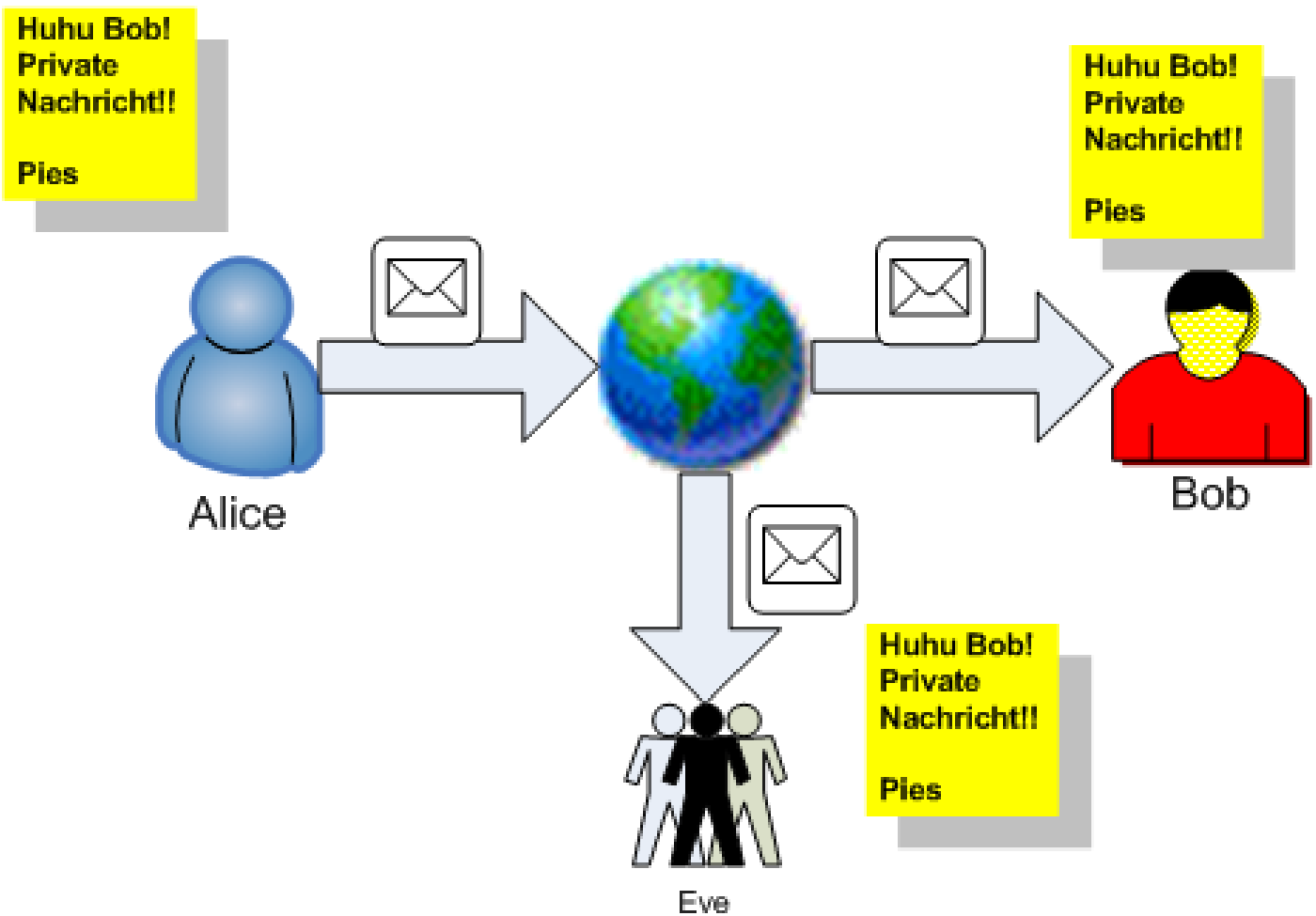
# Alice and Bob

## Vorstellung einiger Figuren in der Kryptografie

- *Alice* möchte Bob eine Mail senden
- *Bob* möchte wissen, wer die Mail gesendet hat
- *Eve* würde gerne die Mail mitlesen

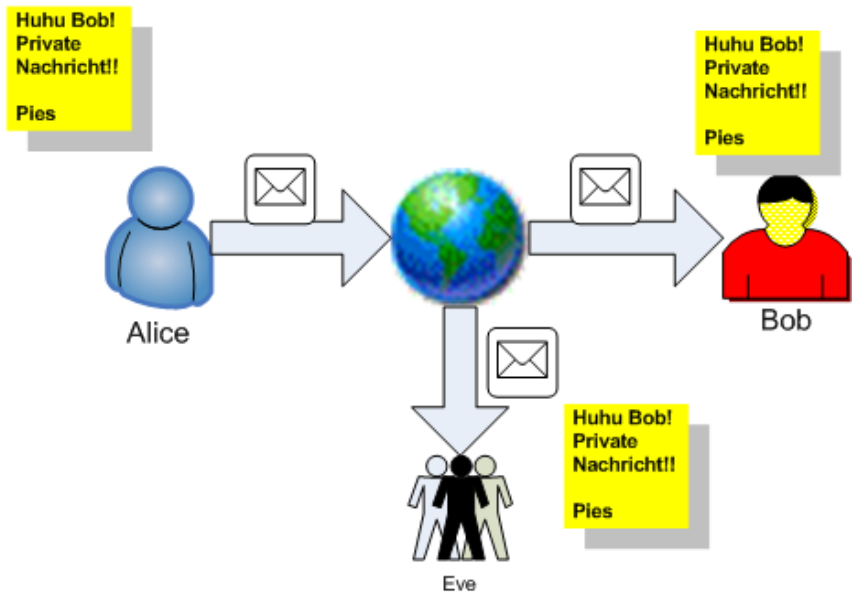
# Eavesdropping

Alice schickt Bob eine Nachricht, Eve kann sie lesen



# Eavesdropping

## Alice schickt Bob eine Nachricht, Eve kann sie lesen




Konventionelle Art eMails und andere Daten zu verschicken!

# Eavesdropping

## Eine Demonstration von `traceroute`

### Code

```
4tmuelle@rzdspc10:~$ /usr/sbin/traceroute mail.gmx.net
traceroute to mail.gmx.net (213.165.64.20), 30 hops max, 40 byte packets
 1  cbswitch02.informatik.uni-hamburg.de (134.100.9.250)  0.805 ms
 2  atm-informatik.informatik.uni-hamburg.de (134.100.5.33)  0.659 ms
 3  nixgate-ge-crrz.rrz.uni-hamburg.de (134.100.254.178)  0.627 ms
 4  188.1.47.37 (188.1.47.37)  0.540 ms
 5  cr-berlin1-po0-0.x-win.dfn.de (188.1.18.109)  13.425 ms
 6  cr-frankfurt1-po13-0.x-win.dfn.de (188.1.18.54)  13.548 ms
 7  ir-frankfurt2-po6-0.x-win.dfn.de (188.1.80.66)  13.401 ms
 8  212.227.112.37 (212.227.112.37)  13.522 ms
 9  so-4000.gw-backbone-a.bs.ka.schlund.net (212.227.120.8)  15.881 ms
10  a0kac1.gw-distg-a.bs.ka.schlund.net (212.227.116.214)  16.029 ms
11  mail.gmx.net (213.165.64.20)  15.905 ms
```

# Keine Sicherheit möglich?

Alice und Bob wollen nun sicher kommunizieren. Keiner soll die Nachricht

- lesen
- verändern
- nochmal abschicken

können.

Symmetrische Verschlüsselung fällt weg, da sicherer Kanal für ein Passwort benötigt wird → **Public Key Verfahren**

# Keine Sicherheit möglich?

Alice und Bob wollen nun sicher kommunizieren. Keiner soll die Nachricht

- lesen
- verändern
- nochmal abschicken

können.
Symmetrische Verschlüsselung fällt weg, da sicherer Kanal für ein Passwort benötigt wird → Public Key Verfahren

# Keine Sicherheit möglich?

Alice und Bob wollen nun sicher kommunizieren. Keiner soll die Nachricht

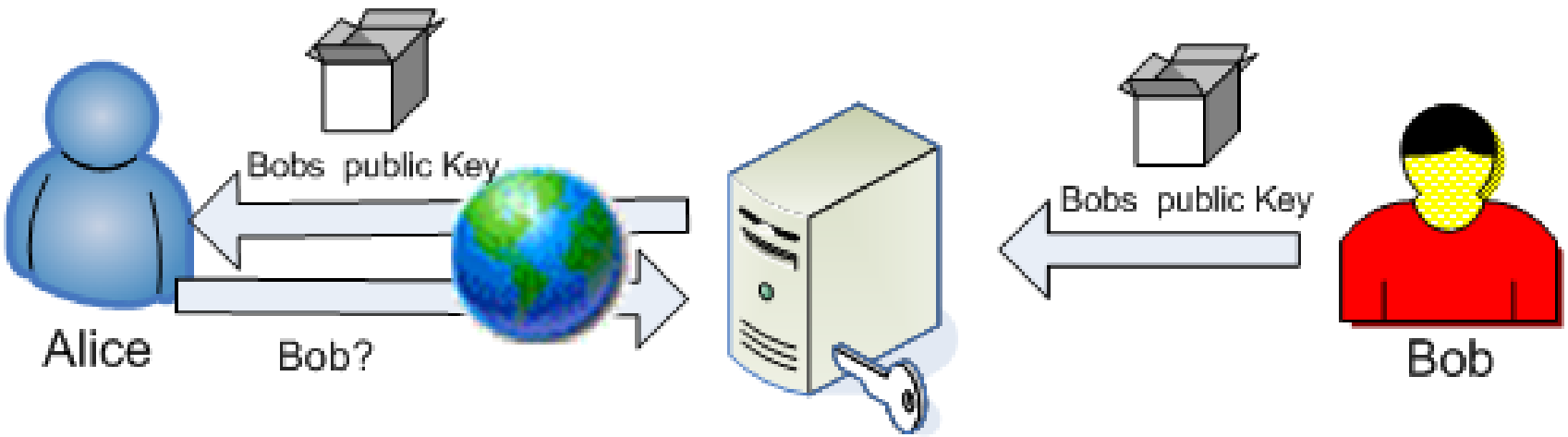
- lesen
- verändern
- nochmal abschicken

können.
Symmetrische Verschlüsselung fällt weg, da sicherer Kanal für ein Passwort benötigt wird $\rightarrow$ **Public Key Verfahren**

# Was sind Keyserver

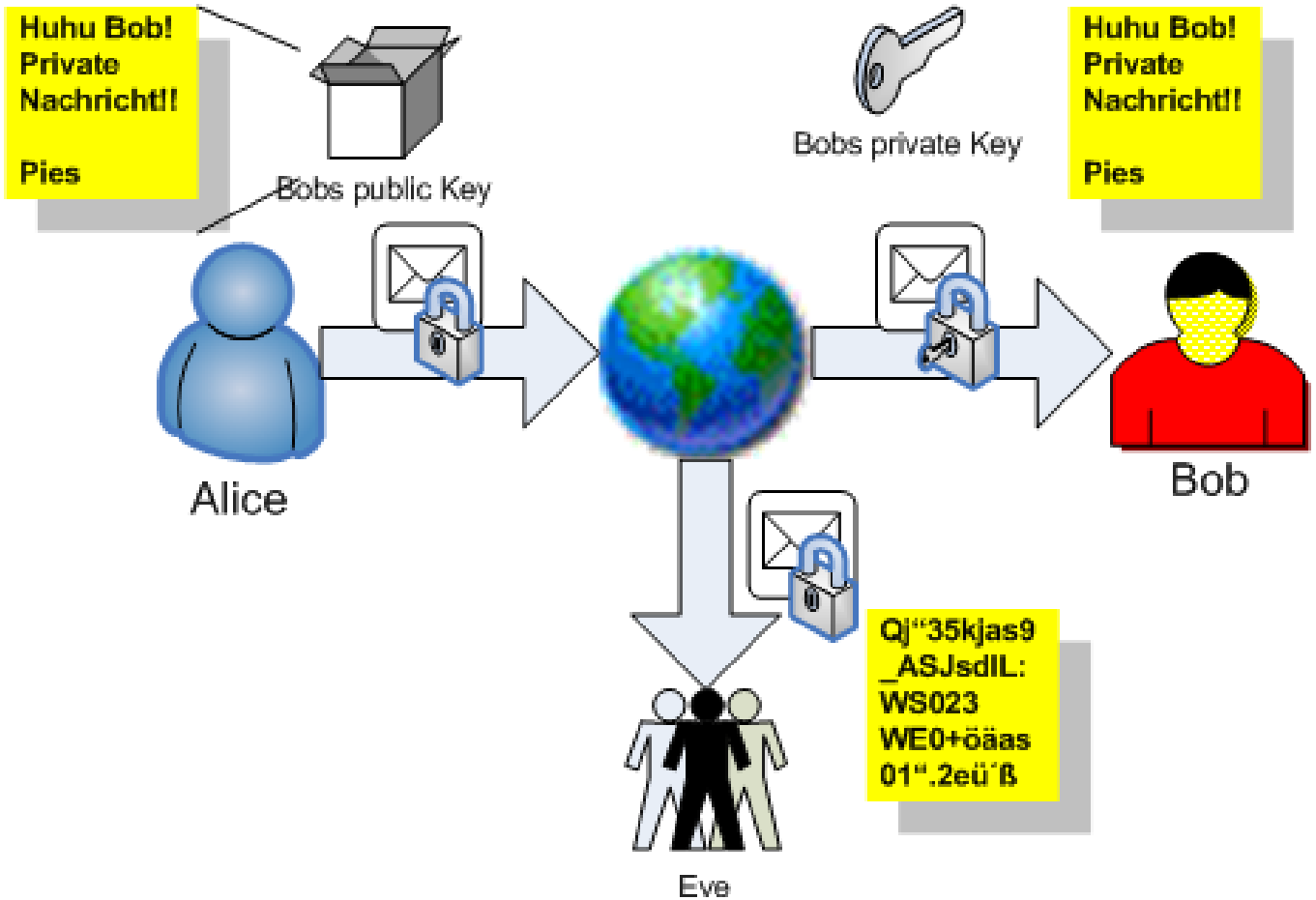
Keyserver

- speichern (beliebige) Öffentliche Schlüssel
- erlauben eine Suche nach Schlüsseln
- geben jedem diese Schlüssel



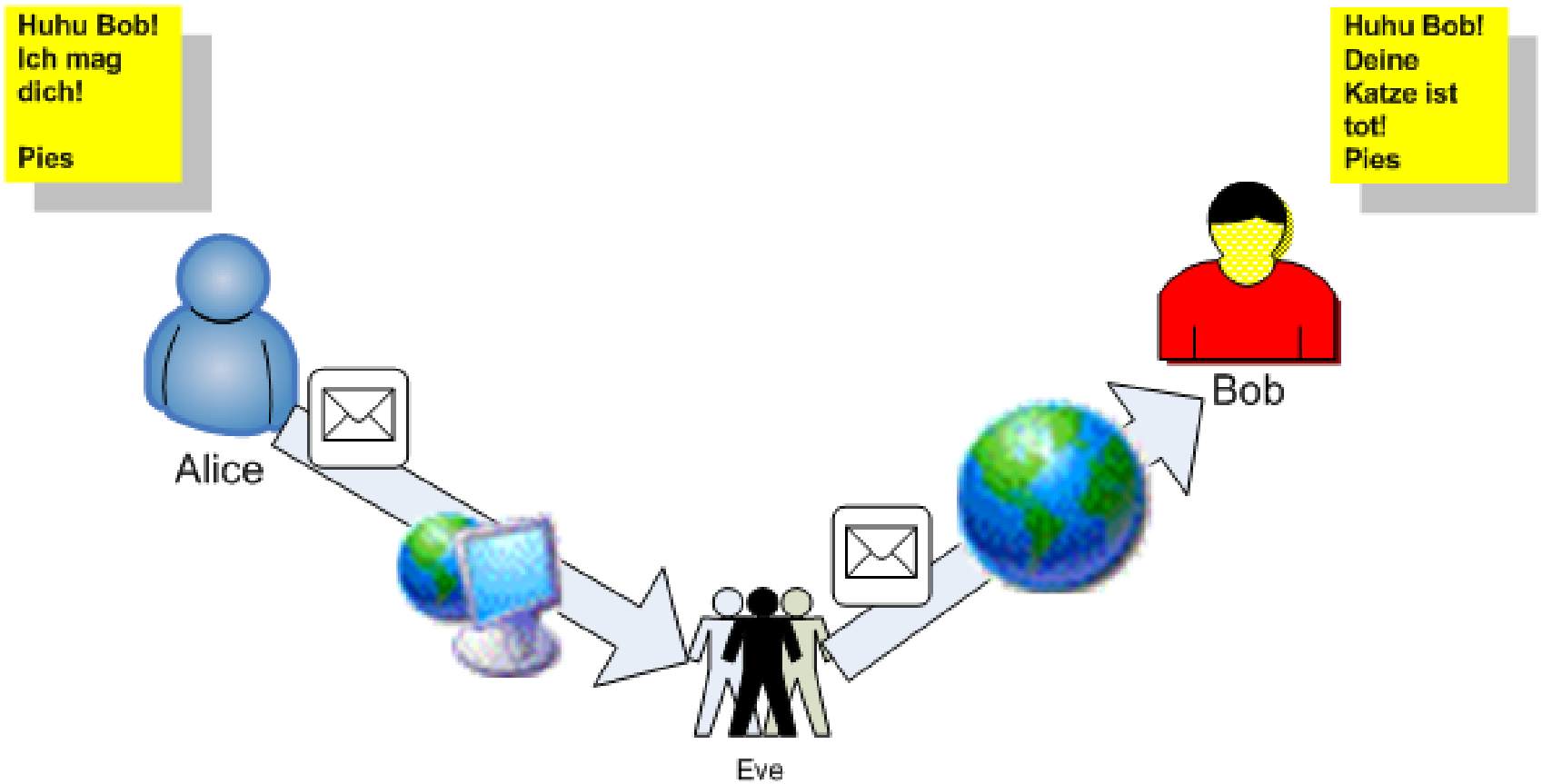
# Verschlüsseln

Alice schickt Bob eine verschlüsselte Nachricht, Eve kann sie **nicht** lesen



# Spoofing

Alice schickt Bob eine Mail. Eve verändert unbemerkt diese Mail



# Spoofing

## Eine Demonstration mit `telnet`

### Code

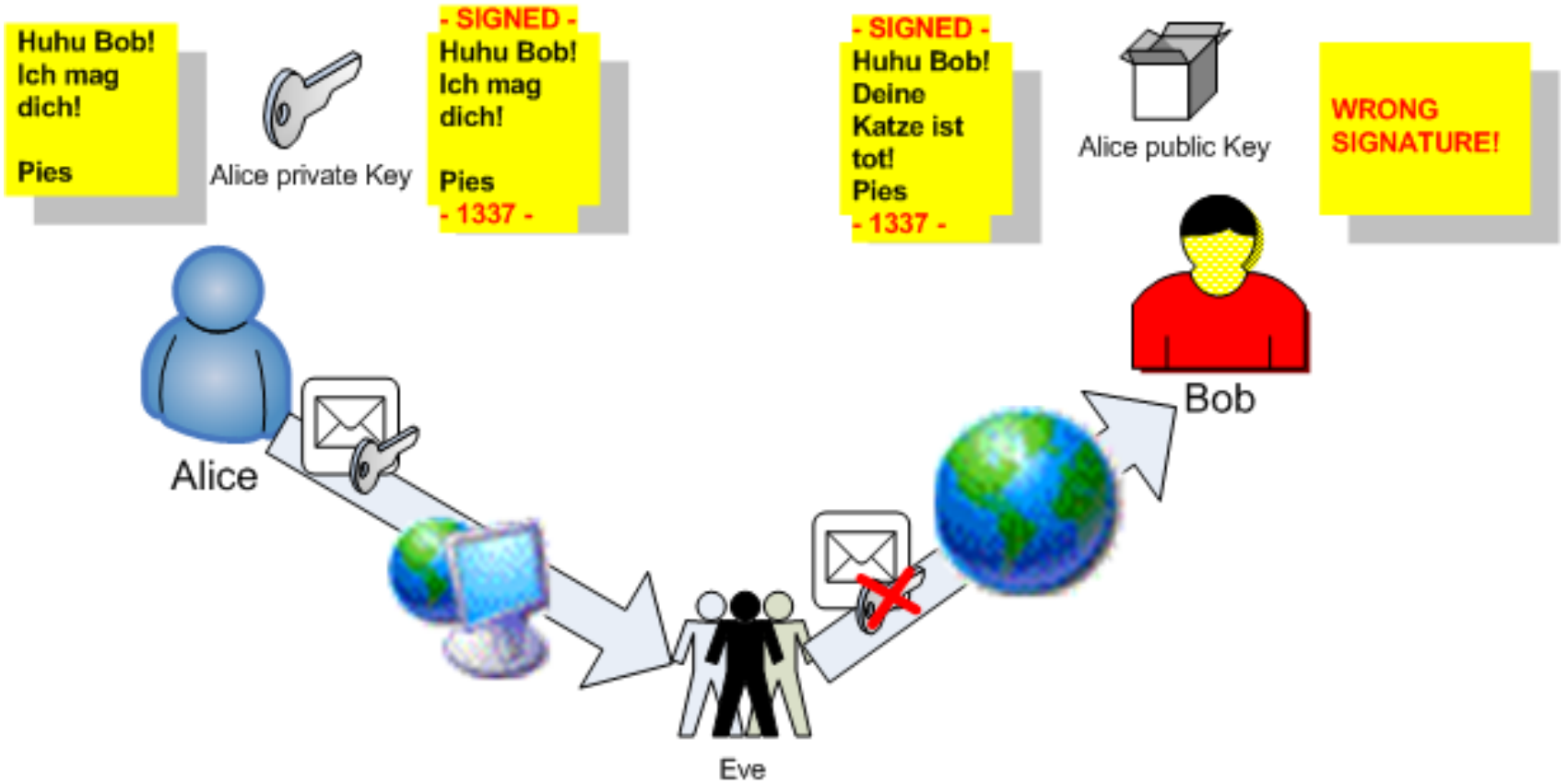
```
$ telnet mail3.netbeat.de 25
Trying 193.254.185.27...
Connected to mail3.netbeat.de.
Escape character is '^]'.
220 mail3.netbeat.de ESMTP
HELO mail3.netbeat.de
250 mail3.netbeat.de
MAIL FROM:<dekan@uni-hamburg.de>
250 ok
RCPT TO:<4tmuelle@informatik.de>
250 ok
```

```
DATA
354 go ahead
From: <dekan@uni-hamburg.de>
To: <4tmuelle@informatik.de>
Subject: Ihre Exmatrikulation
Hallo Muelli,
du bist exmatrikuliert

Pies
.
250 ok 1134425889 qp 21627
QUIT
221 mail3.netbeat.de
Connection closed by foreign host.
```

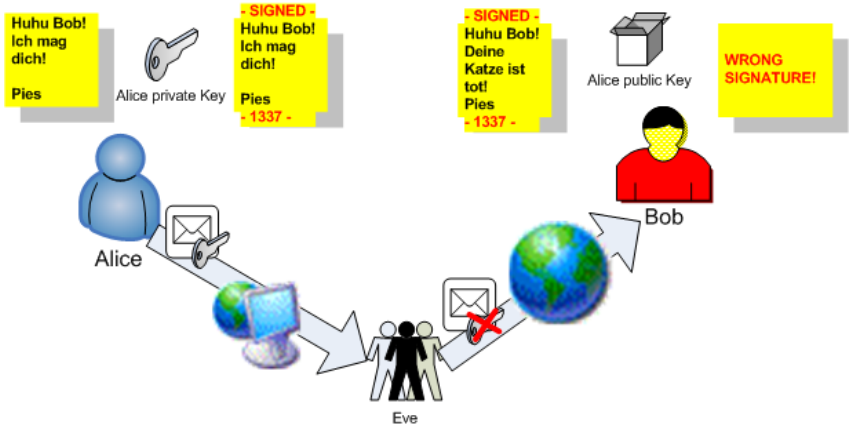
# Signieren

Alice signiert ihre Mail, der Hash stimmt nicht mehr → die Signatur ist ungültig



# Signieren

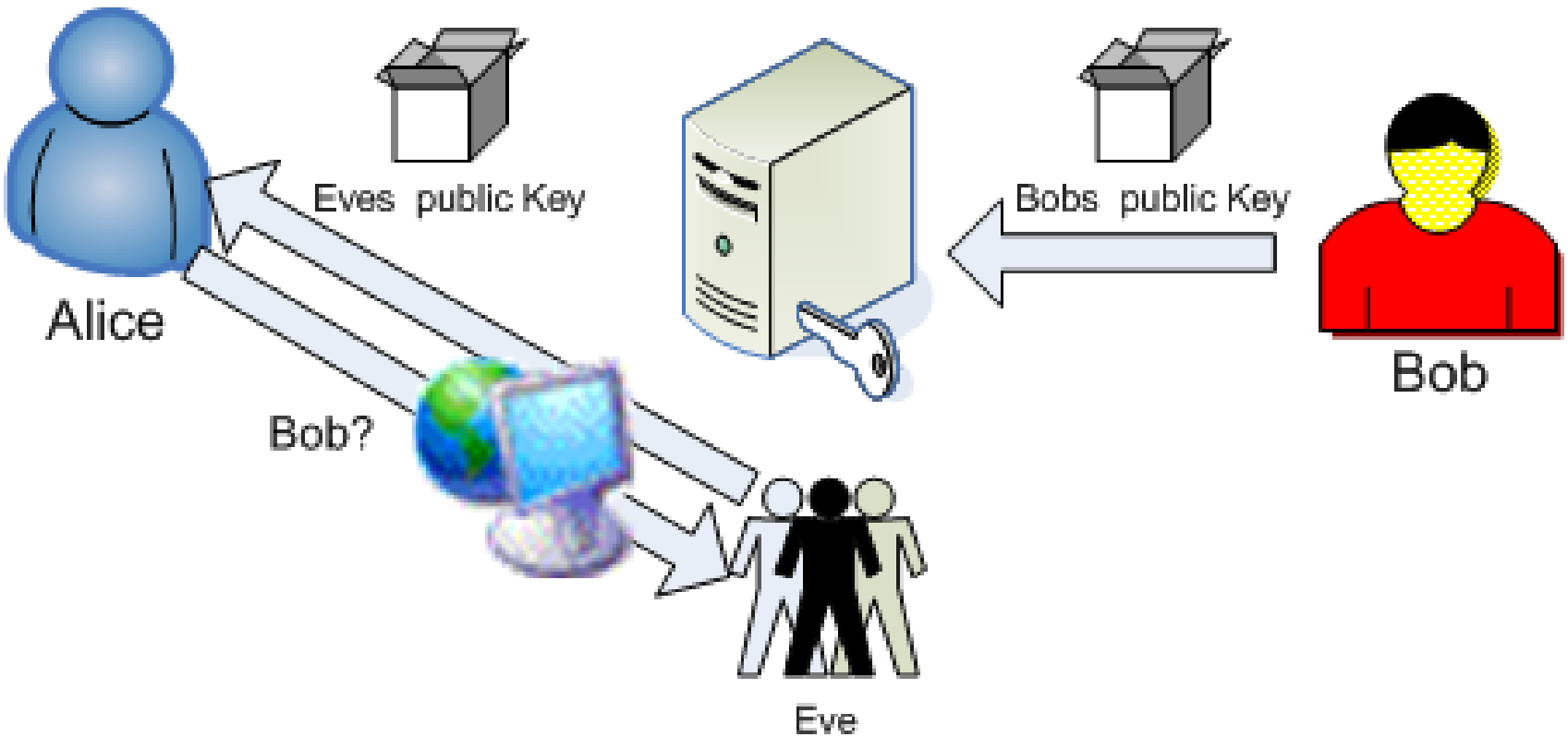
Alice signiert ihre Mail, der Hash stimmt nicht mehr → die Signatur ist ungültig




Nicht nur Texte können signiert werden, sondern auch andere Public Keys! → Web of Trust

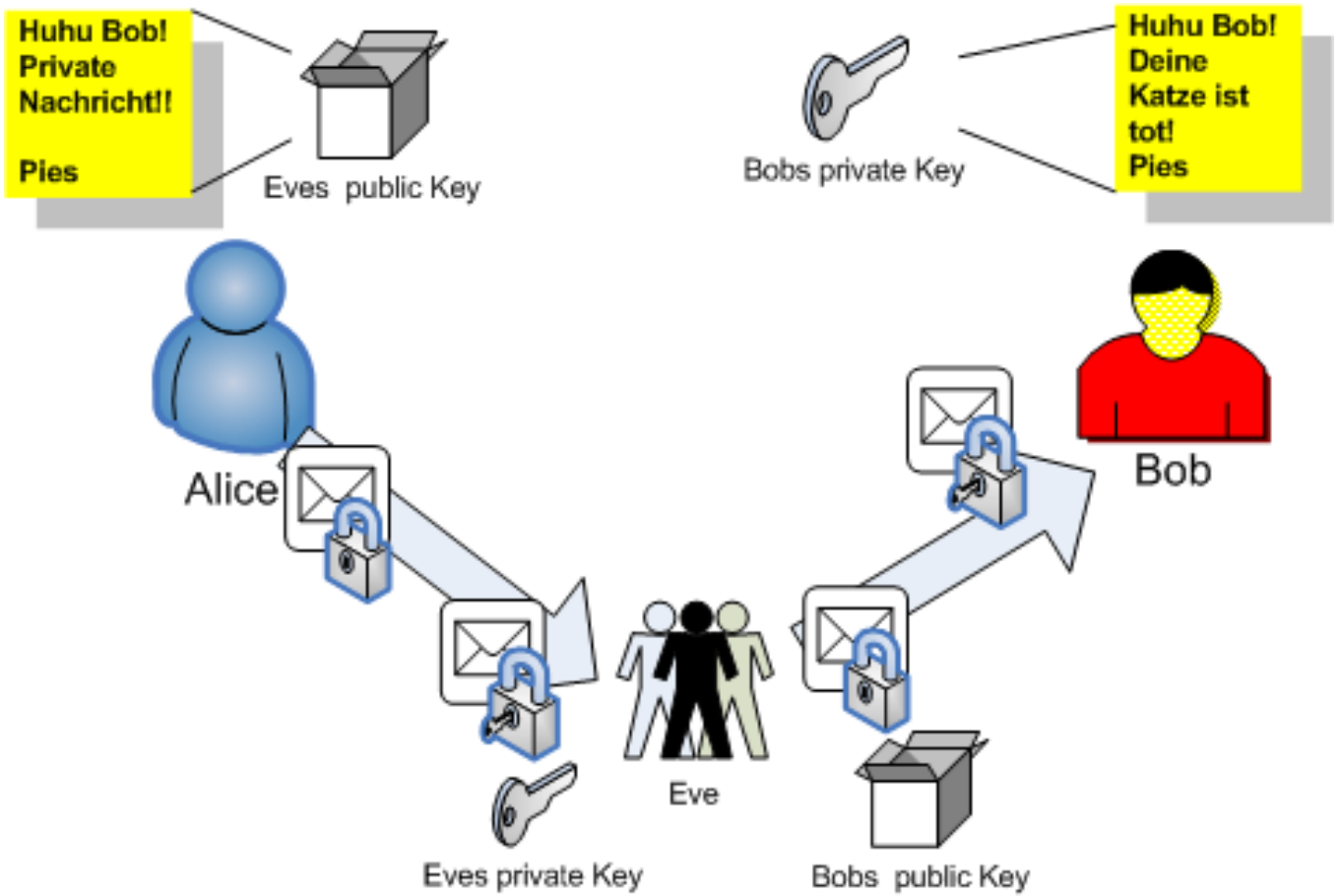
# Man In The Middle

## Ein Schlüssel wird vorgetäuscht



# Man In The Middle

## Eine Nachricht wird vorgetäuscht



# Web of Trust

- Signieren von Public Keys von Zertifizierungsstellen (Certificate Authority)
  - unflexible Struktur
  - single point of failure
- Signieren von Public Keys mit dem eigenen Private Key
  - flexibel
  - selbstheilend → Revocation Certificate
  - keinen single point of failure

# Web of Trust

- Signieren von Public Keys von Zertifizierungsstellen (Certificate Authority)
  - unflexible Struktur
  - single point of failure
- Signieren von Public Keys mit dem eigenen Private Key
  - flexibel
  - selbstheilend $\rightarrow$ Revocation Certificate
  - keinen single point of failure

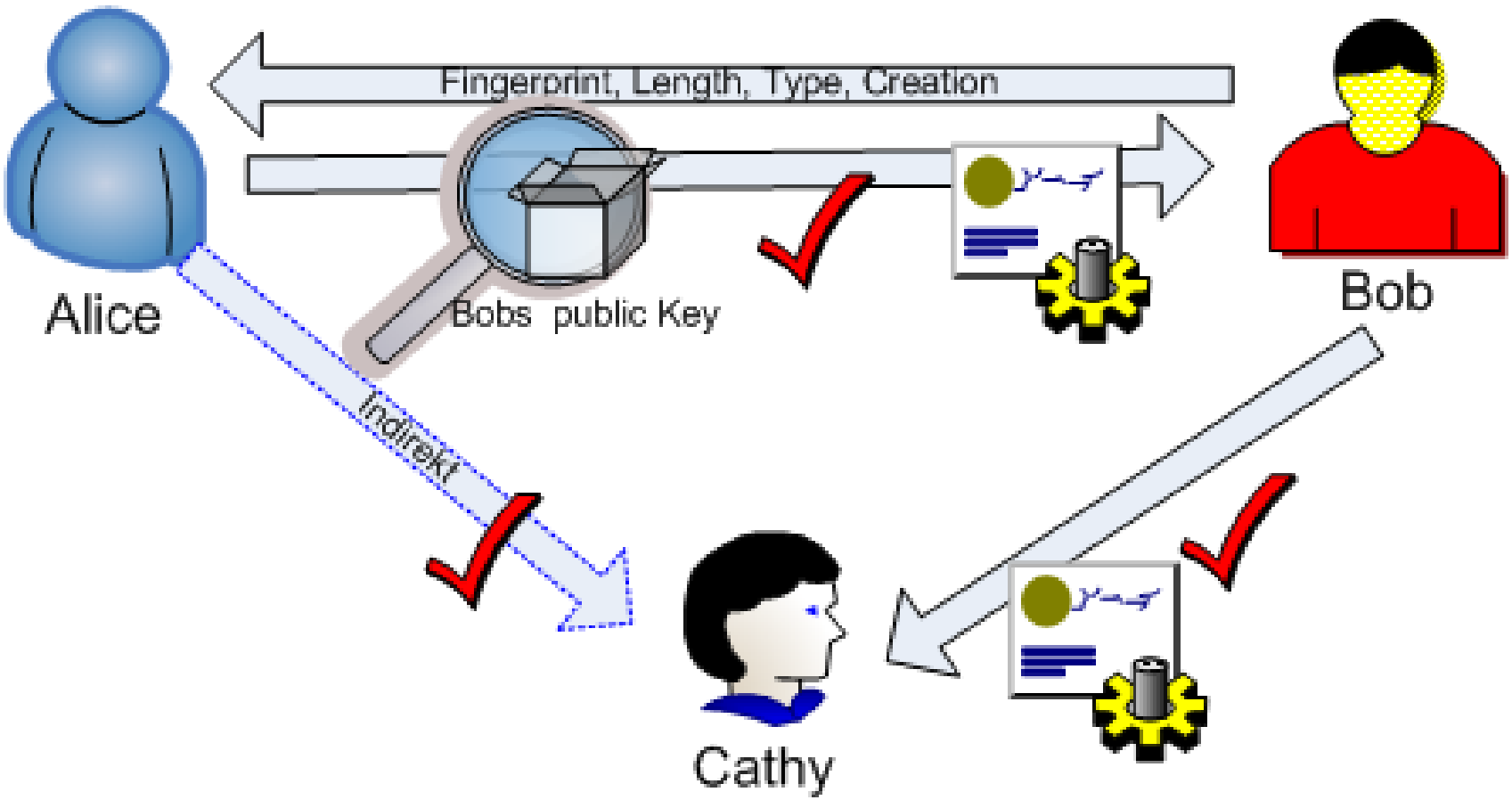
# Schlüsseltausch im Web of Trust I

- Alice möchte Bob eine Mail schicken
  - sie holt seinen Public Key vom Keyserver
  - sie überprüft ihren Public Key durch Vergleichen des Fingerprints über einen sicheren Kanal mit Bob
  - sie signiert Bobs Public Key mit ihrem Private Key und lädt diese Signatur zu einem Keyserver hoch

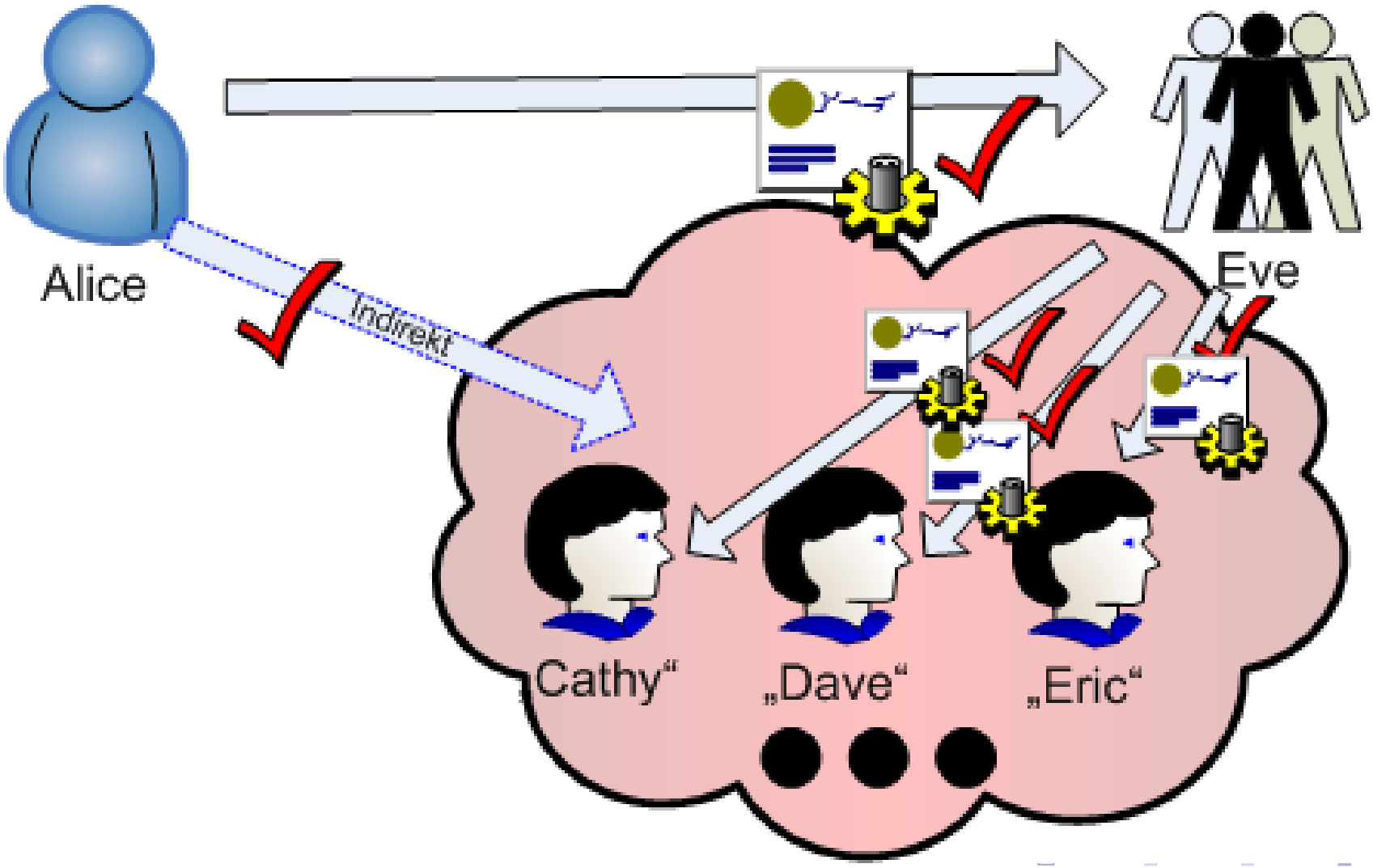
# Schlüsseltausch im Web of Trust II

- Alice möchte Cathy eine Mail schicken
  - sie holt Cathys Public Key vom Keyserver
  - auf dem Keyserver liegt auch die Signatur von Bob
  - da Alice die Identität von Bob kennt und ihm zutraut, Schlüssel nur nach gründlicher Überprüfung zu signieren, muss sie die Identitätsprüfung von Cathy nicht wiederholen

# Simples Web of Trust



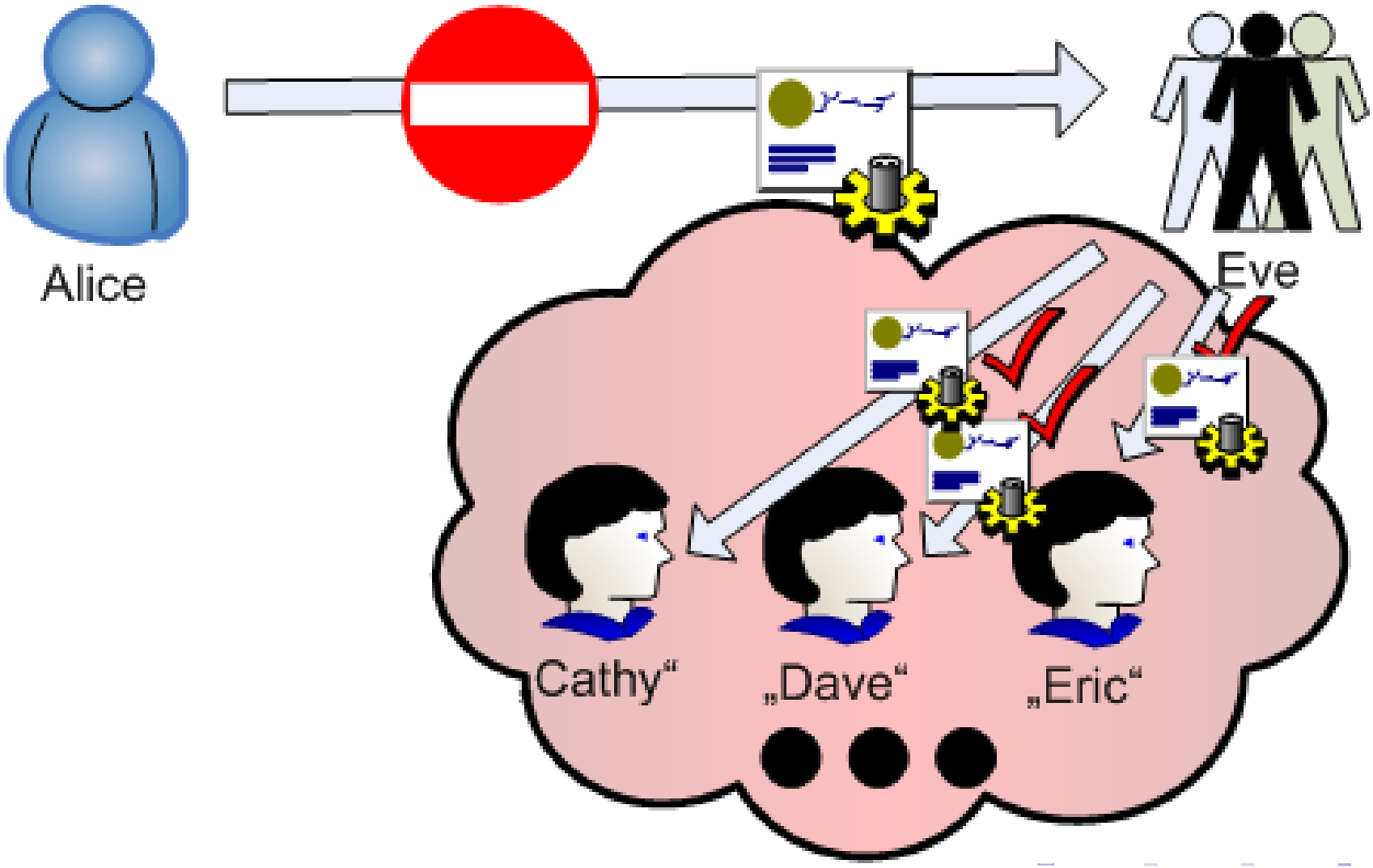
# Angriff auf das WoT



# Was ist Owner Trust?

- Benutzer bekommen einen “Vertrauens-Wert” zugewiesen
- bilden ab, wie sinnvoll jemand mit dem WoT umgeht
- somit werden Signaturen dieses Benutzers unterschiedlich bewertet
- wird **lokal** gespeichert und **nicht** veröffentlicht

# Beispiel Owner Trust



# Probleme

- signaturVZ - wer verkehrt wann mit wem?
- Metadaten: Wer schreibt wem, wann eine Mail über was?
- Keyserver: Abschneiden von Signaturen (insb. Revocation)
- Pgp2Pgp: Binärdaten in Keys
- Keine plausible deniability

# Zusammenfassung

- Mit PGP kann man Botschaften signieren oder verschlüsseln
- Public Key zum verschlüsseln und verifzieren von Signaturen
- Private Key zum entschlüsseln und erstellen von Signaturen
- man kann andere Public Keys signieren $\rightarrow$ Web of Trust

# Quellen

- http://wikipedia.org/
- http://alfie.ist.org/projects/gpg-party/gpg-party.de.html
- http://hp.kairaven.de/pgp/index.html
- man gpg