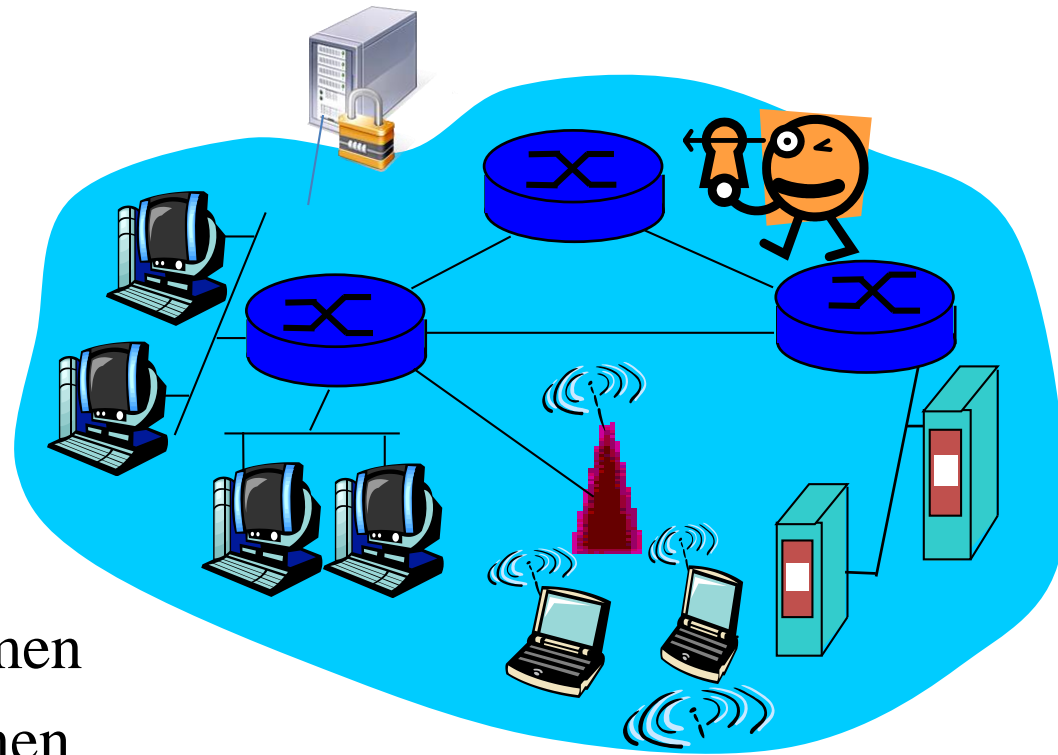


7. Sicherheit

Gliederung

- Sicherheitsziele
- Kryptographie abstrakt
- Authentifikation
- Integrität
- Schlüsselverteilung und Zertifikate
- Firewalls
- Angriffe und Gegenmaßnahmen
- Sicherheit in den verschiedenen Kommunikationsschichten



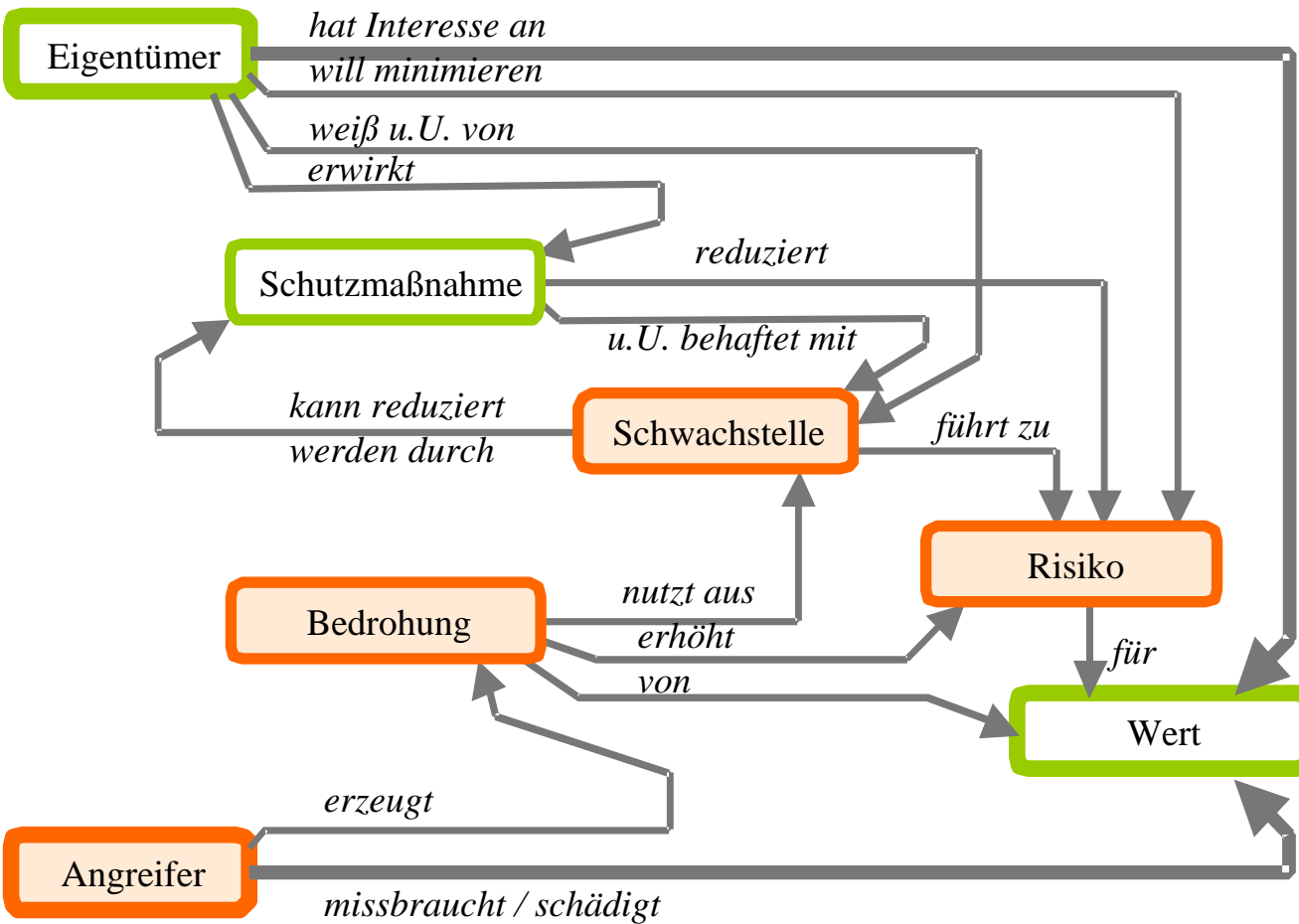
Lernziele:

◆ Prinzipien der Sicherheit im Netz

- Kryptographie und Nutzungen, die über Vertraulichkeitsschutz hinausgehen
- Authentifikation
- Nachrichtenintegrität
- Schlüsselverteilung

◆ Sicherheit in der Praxis

- Firewalls
- Sicherheitsfunktionen in den Kommunikationsschichten



Sicherheitsziele

Vertraulichkeit

Die drei immer genannten Hauptziele

Integrität

Verfügbarkeit

Anonymität

Es gibt weitere Ziele. Ziele können gegensätzlich sein

Nachvollziehbarkeit / Zurechenbarkeit

...

Authentifikation

Die beiden grundlegenden Hilfsdienste

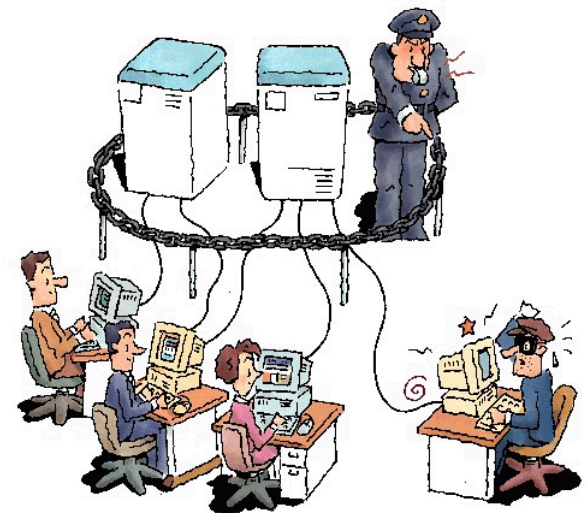
Autorisierung

Im Netz:

Nachrichtenvertraulichkeit / Integrität

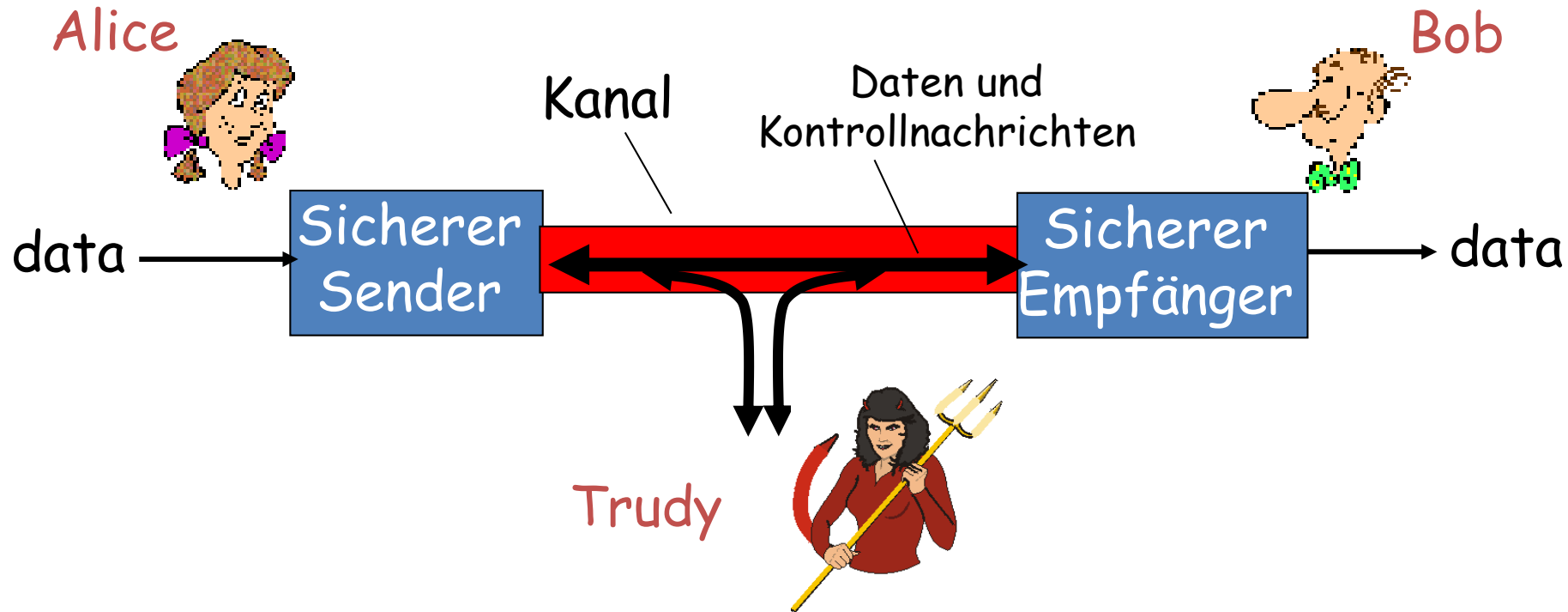
Nachrichten--Absenderauthentifikation,

Empfängerauthentifikation



Freunde und Feinde: Alice, Bob, Trudy

- In der Welt der Netzsicherheit wohlbekannt
- Bob und Alice (befreundet!) wollen sicher kommunizieren
- Trudy (der Eindringling) kann Nachrichten abfangen, löschen, verändern, einschleusen



E: Was kann Trudy (allg. ein “bad Guy”) tun?

A: Jede Menge!

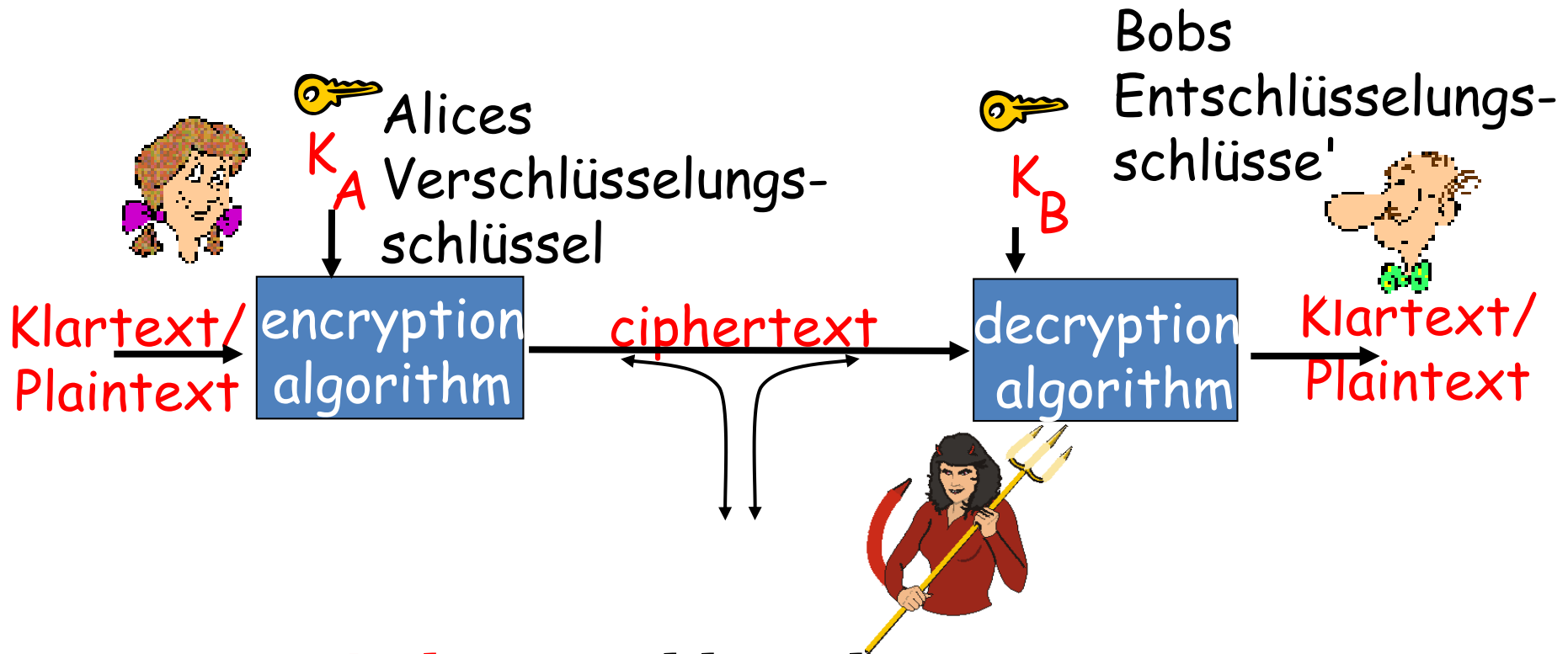
- *Abhören*
- aktiv neue Nachrichten **einfügen** / unterschreiben
- *Maskerade*: fälschen (spoof) der Quelladresse eines Pakets (oder anderer Kontrollfelder)
- *Sitzungsübernahme* (Hijacking) / Verbindungsübernahme
- *Verfügbarkeitsattacke* (Denial of Service / DoS-Attacke)



darüber später mehr.....



Kryptographie abstrakt



Symmetrische Verschlüsselung:

Beide Schlüssel sind identisch – **Shared Secret**

Asymmetrische Verschlüsselung:

Paar aus **öffentlichem** und **privatem** Schlüssel

(Public Key, Private Key), (Privater Schlüssel ist geheim)

Prinzipien der Verschlüsselung

- Algorithmen i.d.R. bekannt, Schlüssel unbekannt
 - Man unterscheidet
 - Monoalphabetische Verschlüsselung
jedes Zeichen wird einzeln verschlüsselt
 - Blockverschlüsselung
ganze Blöcke werden verschlüsselt
- sowie
- Symmetrische Verschlüsselung
identische Schlüssel auf beiden Seiten
 - Asymmetrische Verschlüsselung
unterschiedliche Schlüssel

Blockverschlüsselung

- Nachrichten werden in Blöcken fester Größe verschlüsselt (z.B., 64-bit Blöcke).
- 1-zu-1 Abbildung zwischen Blöcken des Klartextes und des verschlüsselten Textes

Beispiel mit $k=3$:

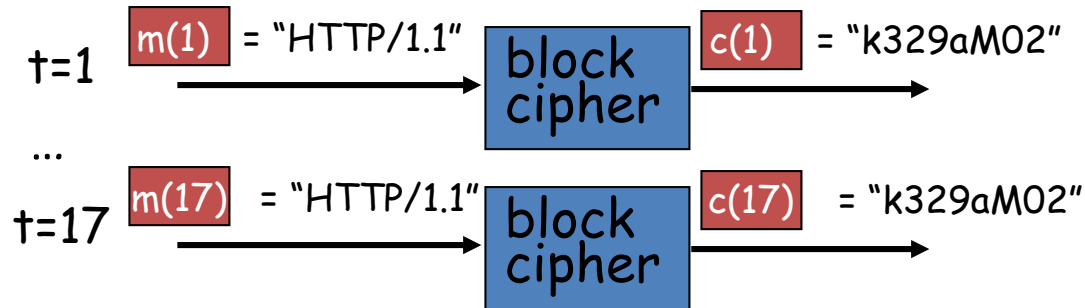
<u>input</u>	<u>output</u>
000	110
001	111
010	101
011	100

<u>input</u>	<u>output</u>
100	011
101	010
110	000
111	001

- Es gibt $2^k!$ Möglichkeiten der Abbildung
(für $k=3$ nur 40320 für $k=64$ sehr viele..)

Blockverschlüsselung

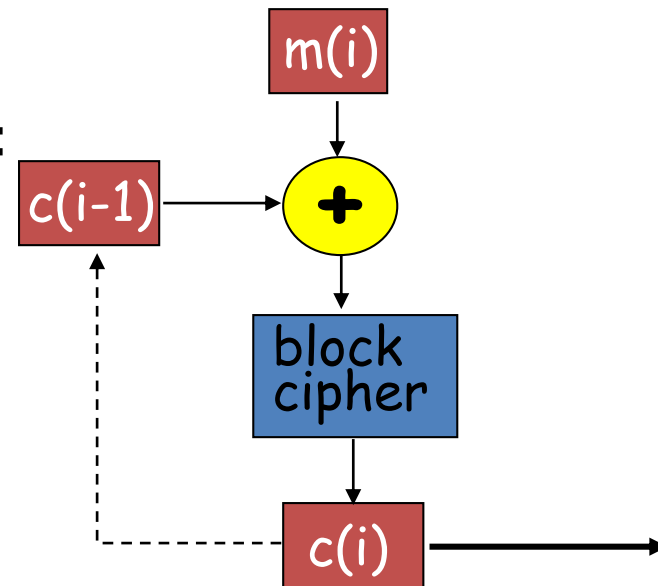
Blockverschlüsselung liefert identische Ergebnisse bei identischen Blöcken



Deshalb i.d.R. Nutzung
positionsabhängiger Schlüssel:

Verfügbare Verfahren:

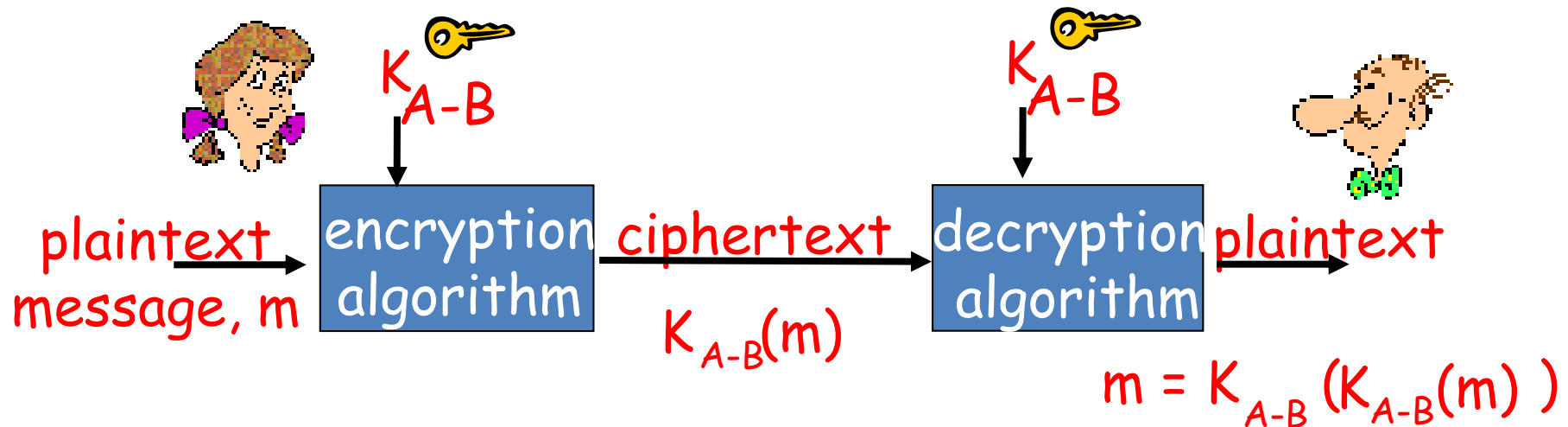
DES, 3DES, AES



Angriffsmöglichkeiten

- Angriff auf Basis des verschlüsselten Textes: Trudy besitzt den verschlüsselten Text und kann ihn analysieren
- Zwei Ansätze:
 - Ausprobieren aller Schlüssel, u.U. sind einzelne Schlüsse wahrscheinlicher als andere
Voraussetzung Klartext kann identifiziert werden
 - Statistische Analyse
- Angriff bei (in Teilen) bekanntem Klartext: Trudy kann Klartext verschlüsseltem Text zuordnen
z.B. bei einem monoalphabetischen Verfahren erkennt Trudy die Verschlüsselung von Alice und
- Angriff bei ausgewähltem Klartext: Trudy kann den verschlüsselten Text zu einem beliebigen Klartext generieren

Symmetrische Verschlüsselung



Symmetrische Verschlüsselung:

Bob und Alice kennen beide gemeinsam denselben Schlüssel: Shared Secret K_{A-B}

- **Problem**

Das Shared Secret muss irgendwann vorher einmal auf sichere Weise kommuniziert worden sein: *Man kann nur dann sicher kommunizieren, wenn man vorher schon einmal sicher kommunizieren konnte!*

- **Vorteil**

Leistungsfähige Algorithmen und Implementierungen verfügbar.

- **Beispiele:** DES, TripleDES, AES

Grundlagen der Public Key Verschlüsselung

Anforderungen:

1. Es gibt zwei Schlüssel, K^+ und K^- , so dass
 $K^-(K^+(m)) = m$
2. K^- kann nicht aus K^+ oder $K^+(m)$ hergeleitet werden

Zugehöriger Algorithmus: **RSA**: Rivest, Shamir, Adelson
Algorithmus

- 1977 publiziert
- 1983 patentiert
- 2000 Patent erloschen

Grundlagen der Public Key Verschlüsselung

Basis: **modulo Arithmetik**

$x \bmod n$ = Rest, wenn x durch n dividiert wird

Eigenschaften:

$$[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n$$

$$[(a \bmod n) - (b \bmod n)] \bmod n = (a-b) \bmod n$$

$$[(a \bmod n) * (b \bmod n)] \bmod n = (a*b) \bmod n$$

Somit gilt

$$(a \bmod n)^d \bmod n = a^d \bmod n$$


Beispiel: $x=14$, $n=10$, $d=2$:

$$(x \bmod n)^d \bmod n = 4^2 \bmod 10 = 6$$

$$x^d = 14^2 = 196 \text{ damit gilt auch } x^d \bmod 10 = 6$$

Grundlagen der Public Key Verschlüsselung

Vorgehen:

1. Wähle zwei große Primzahlen p, q
(z.B. Länge 1024 Bit oder größer)
2. Berechne $n = pq$, $z = (p-1)(q-1)$
3. Wähle ein e ($e < n$), das keine gemeinsamen Primfaktoren mit z hat (e, z sind "relative prim")
4. Wähle d , so dass $ed-1$ durch z teilbar ist
(also: $ed \bmod z = 1$)
5. Öffentliche Schlüssel (n, e) . Private Schlüssel (n, d) .


Grundlagen der Public Key Verschlüsselung

Ver- und Entschlüsselung:

1. Seien (n,d) und (n,e) wie auf der vorherigen Folie berechnet
2. Verschlüsselung der Nachricht m ($< n$)

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

3. Entschlüsselung der Nachricht

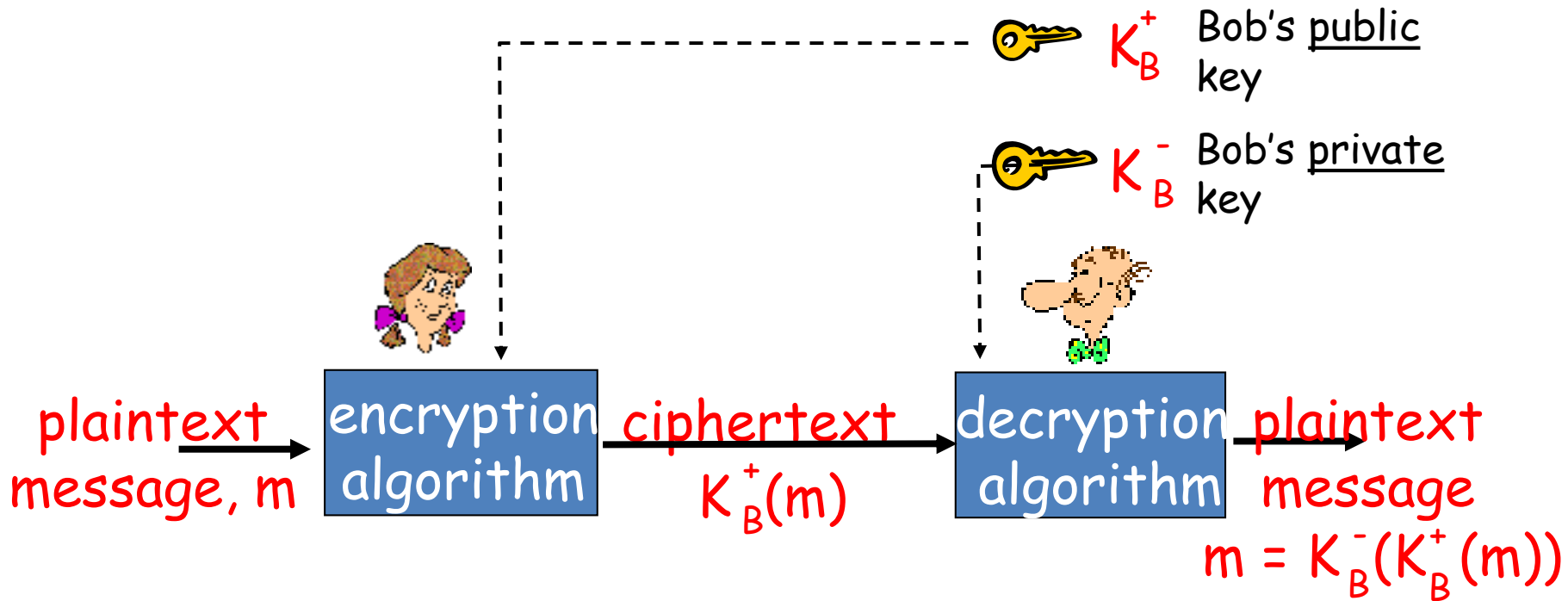
$$m = c^d \bmod n$$

Warum funktioniert das Verfahren??

$$\text{Es gilt hier } m = \underbrace{(m^e \bmod n)}_c^d \bmod n$$

Beweis erfordert Sätze aus der Zahlentheorie!

Public Key Kryptographie – Asymmetrische Verschlüsselung



Public Key Kryptographie [RSA]

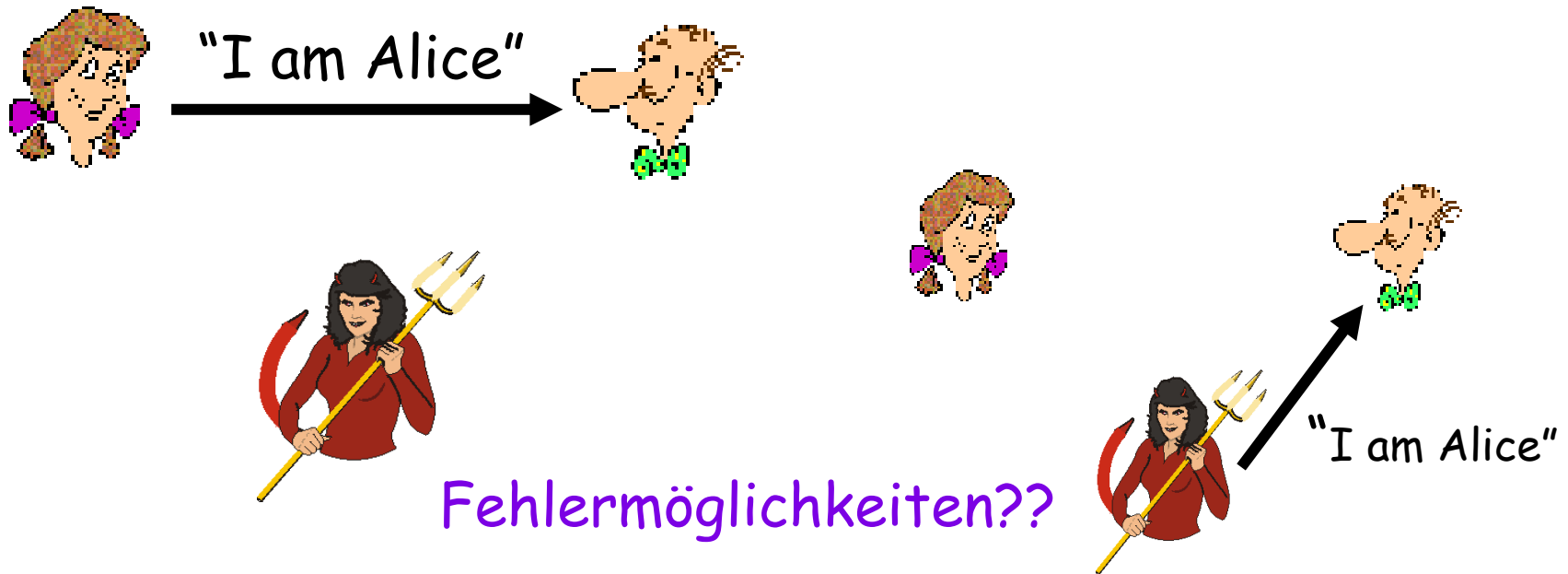
- ❖ Es gibt kein geteiltes Geheimnis
- ❖ *Alle kennen den öffentlichen Schlüssel*
- ❖ *Nur der Empfänger kennt den privaten Entschlüsselungsschlüssel*
- ❖ Es gilt $m = K_B^-(K_B^+(m))$ und $m = K_B^+(K_B^-(m))$

Authentifikation

Bob und Alice kommunizieren per Nachrichtenaustausch.

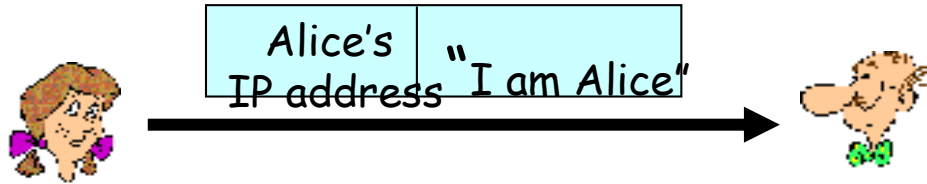
Ziel: Bob möchte, dass Alice ihm beweist, dass sie wirklich Alice ist

Protokoll ap1.0: Alice teilt mit "Ich bin Alice"

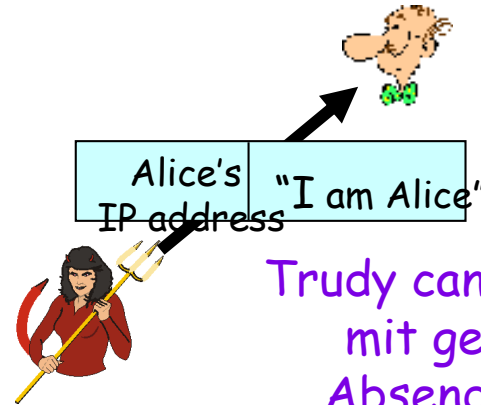


Fehlermöglichkeiten

Protokoll ap2.0: Alice teilt mit "Ich bin Alice" und sende meine IP-Adresse



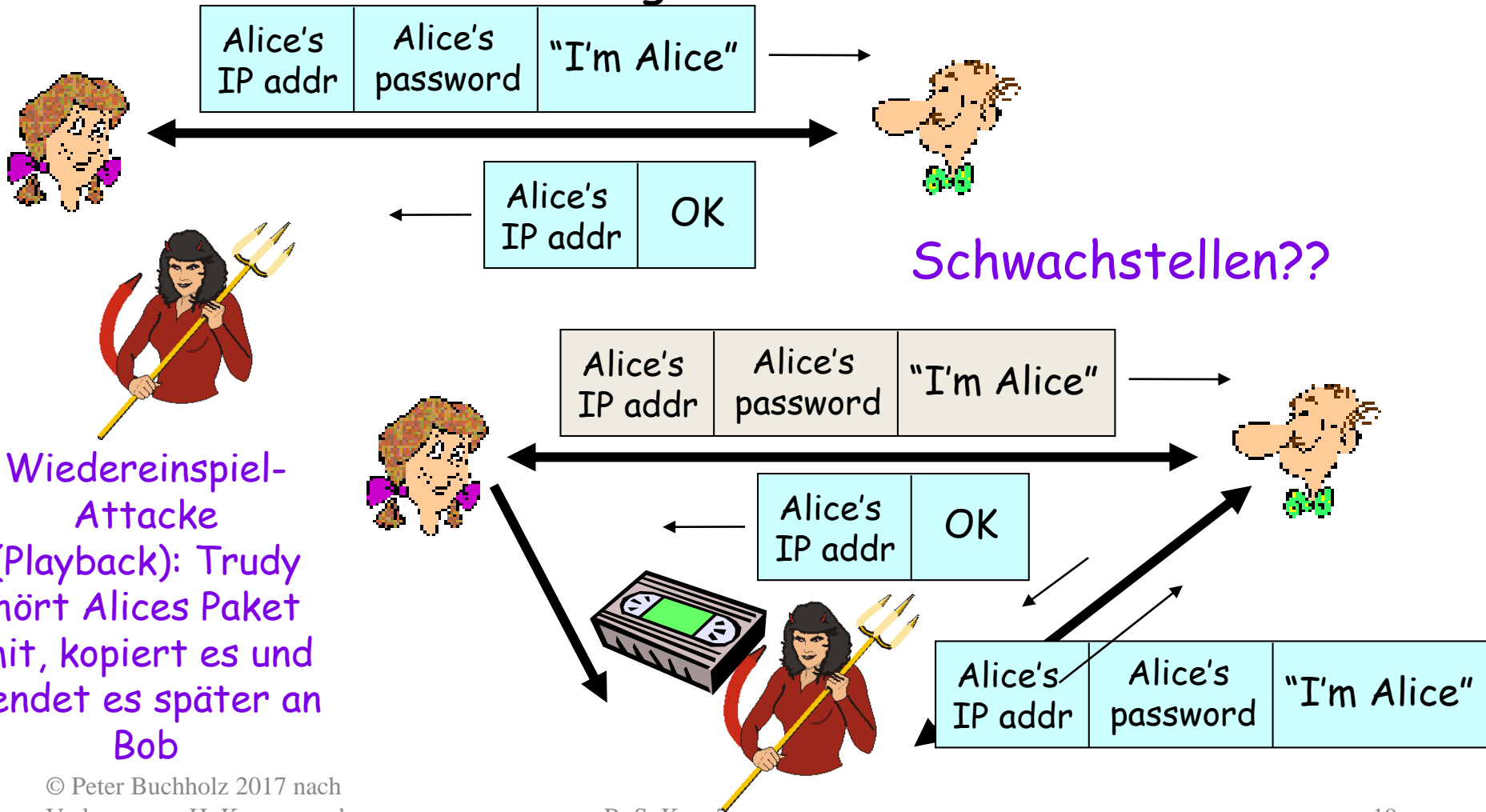
Sende mit
eigener IP-
Adresse



Trudy can ein IP-Paket
mit gefälschter
Absenderadresse
erzeugen (IP-
Spoofing)

Fehlermöglichkeiten

Protokoll ap3.0: Alice teilt mit "Ich bin Alice", sende meine IP-Adresse und ein geheimes Password.

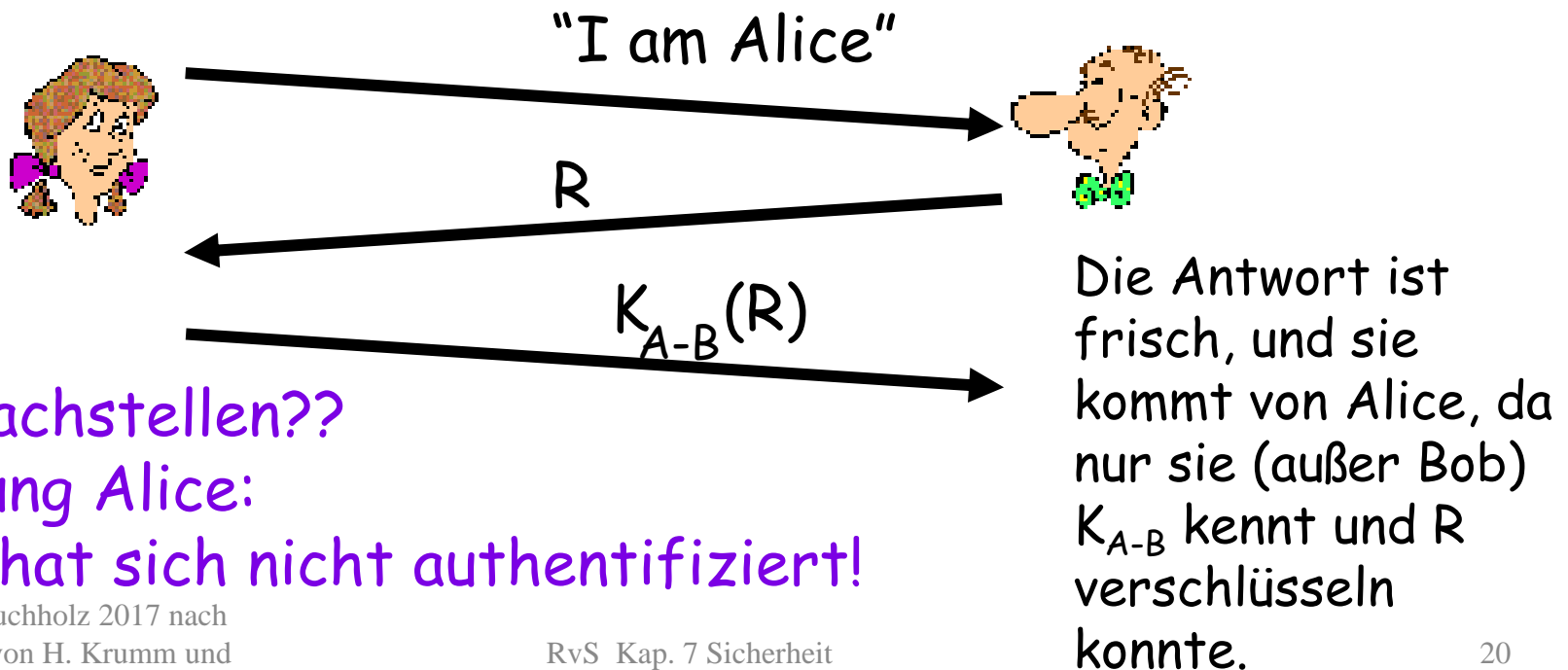


Authentifikation

Ziel: Verhindere erfolgreiche Wiedereinspiel-Attacken

N_{once}: Zahl, die nicht vorhersagbar ist und nur einmal benutzt wird (N_{once})

Protokoll ap.4: Als Beweis dafür, dass Alices Antwort "frisch" ist, sendet Bob eine N_{once} **R** an Alice, Alice muss **R** in verschlüsselter Weise zurücksenden (Challenge-Response-Authentifikation)



Schwachstellen??

Achtung Alice:

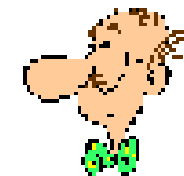
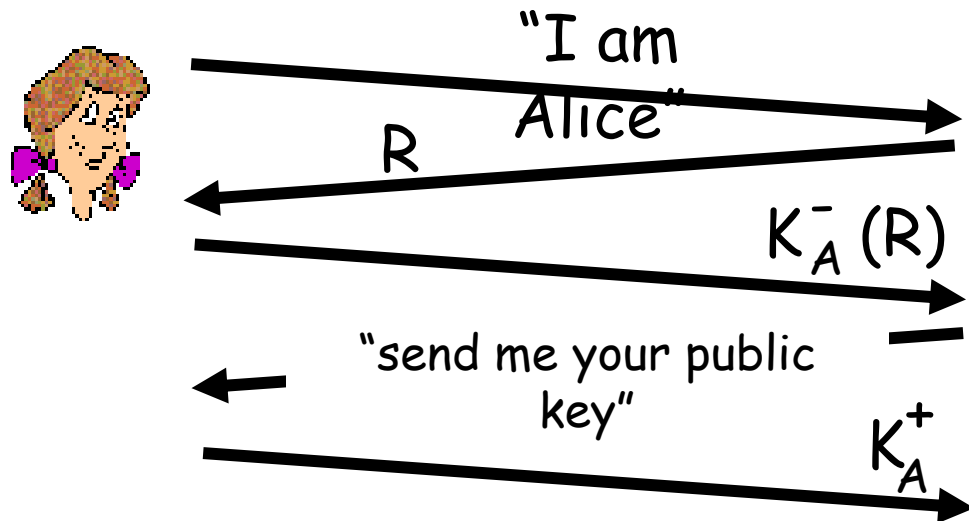
Bob hat sich nicht authentifiziert!

Authentifikation mit Public Key Kryptographie

Bisher wird ein Shared Secret K_{A-B} benötigt, das initial beiden bekannt sein muss

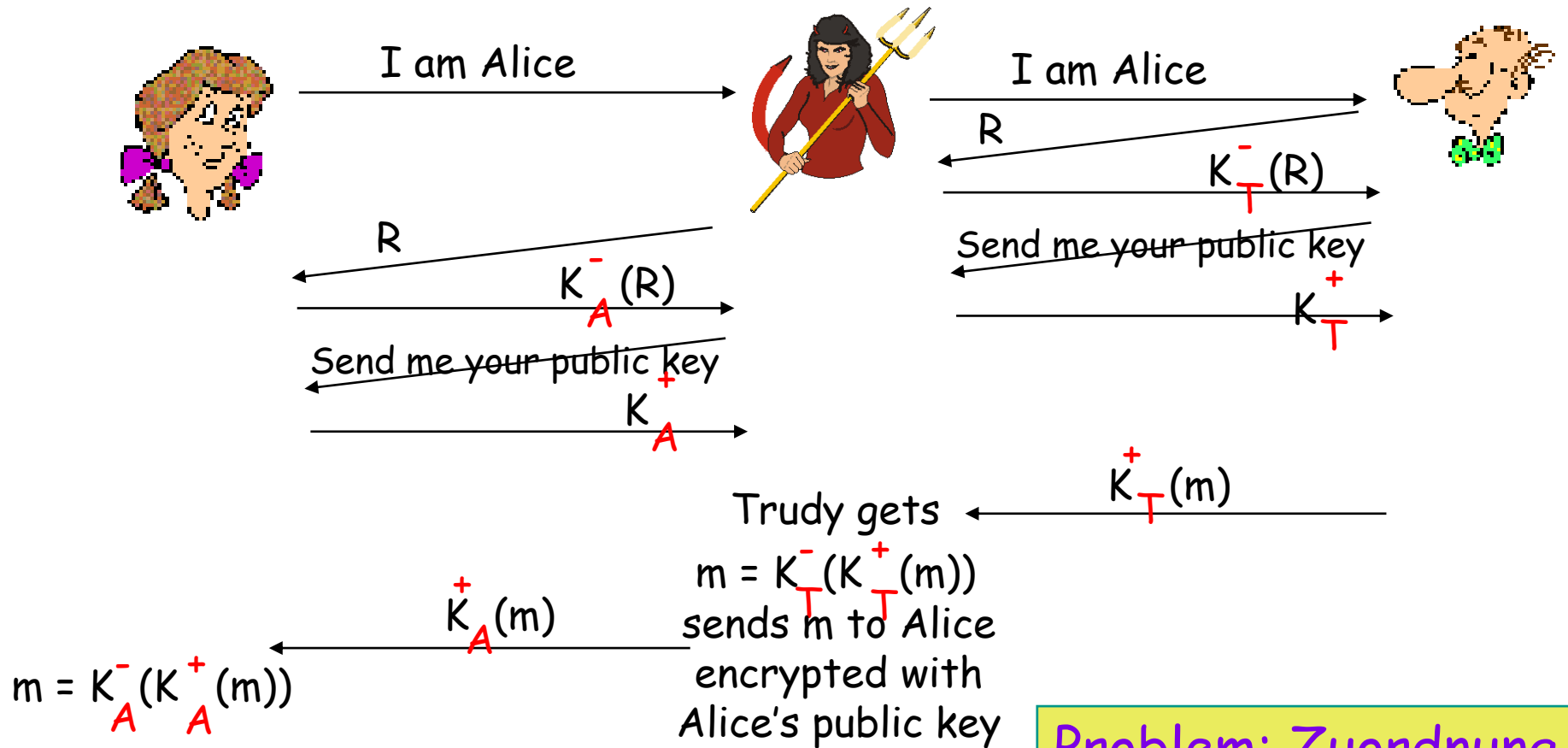
- ◆ Geht es auch mit Public-Key-Verschlüsselung?

Protokoll ap5.0: N_{once} und Signatur



Bob berechnet
 $K_A^+(K_A^-(R)) = R$
und weiß, dass nur
Alice ihren privaten
Schlüssel kennt, so
dass nur sie die
Nachricht erzeugen
konnte

Schwachstelle – “Man in the Middle” Angriff



Problem: Zuordnung
 Alice – K_A^+
 sollte für Bob prüfbar sein

Digitale Signatur

Kryptographische Technik, welche die Funktion handschriftlicher Unterschriften erfüllen soll

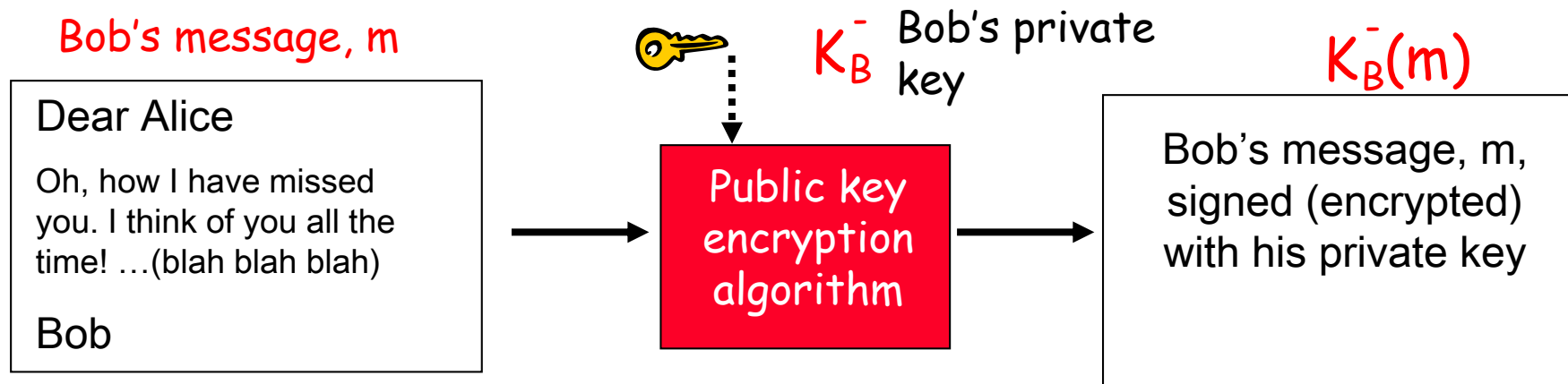
- Sender (Bob) signiert ein Dokument digital und bestätigt damit, dass er das Dokument so erzeugt hat
- **verifizierbar, fälschungssicher:**
Empfänger (Alice) kann Dritten gegenüber beweisen, dass Bob, und niemand anders (auch Alice nicht), das Dokument signiert haben muss
- **ABER:**
 - Kryptoalgorithmen sind nicht ewig sicher:
Digitale Unterschriften müssen alle paar Jahre aufgefrischt werden
 - Private Schlüssel können korumpiert werden: Rückrufe



Digitale Signatur

Einfache digitale Signatur für eine Nachricht m :

- ◆ Bob signiert m dadurch, dass er m mit seinem privaten Schlüssel K_B^- verschlüsselt: $K_B^-(m)$



Wenn Alice diese Nachricht empfängt, den öffentlichen Schlüssel von Bob kennt und davon ausgehen kann, dass Bobs privater Schlüssel nur Bob bekannt ist:

- Bob und kein anderer hat diese Nachricht so signiert
- Bob kann nicht abstreiten, dass er die Nachricht signiert hat

Probleme:

- Asymmetrische Verschlüsselung ist rechenaufwändig
- Wie erfährt Alice den öffentlichen Schlüssel K_B^+ von Bob?

Message Digest – Kryptographische Hashfunktion

Das direkte Signieren langer Nachrichten kostet viel Rechenzeit

Ziel: effizient berechenbarer Fingerabdruck einer Nachricht m : Message Digest $H(m)$

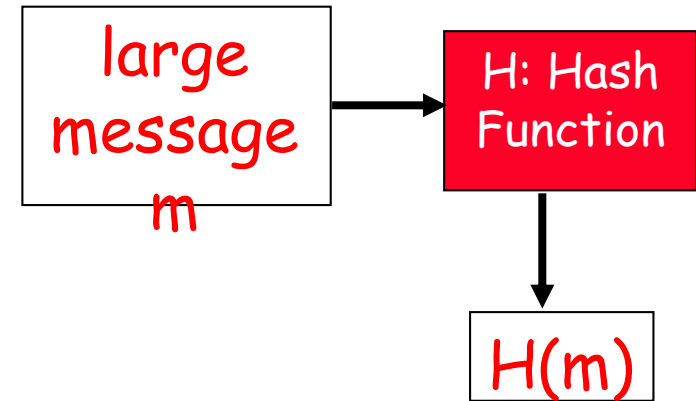
◆ H ist kryptographische Hashfunktion

◆ *Beispiele*

MD5 (RFC 1321)

- Berechnet eine 128 Bit langen Sequenz in 4 Schritten.
- Für eine zufällig gewählte 128 Bit lange Sequenz ist es schwer eine zugehörige Nachricht zu erzeugen, deren MD5 Hash-Sequenz gerade der berechneten Sequenz entspricht.

SHA-1 (NIST Standard)



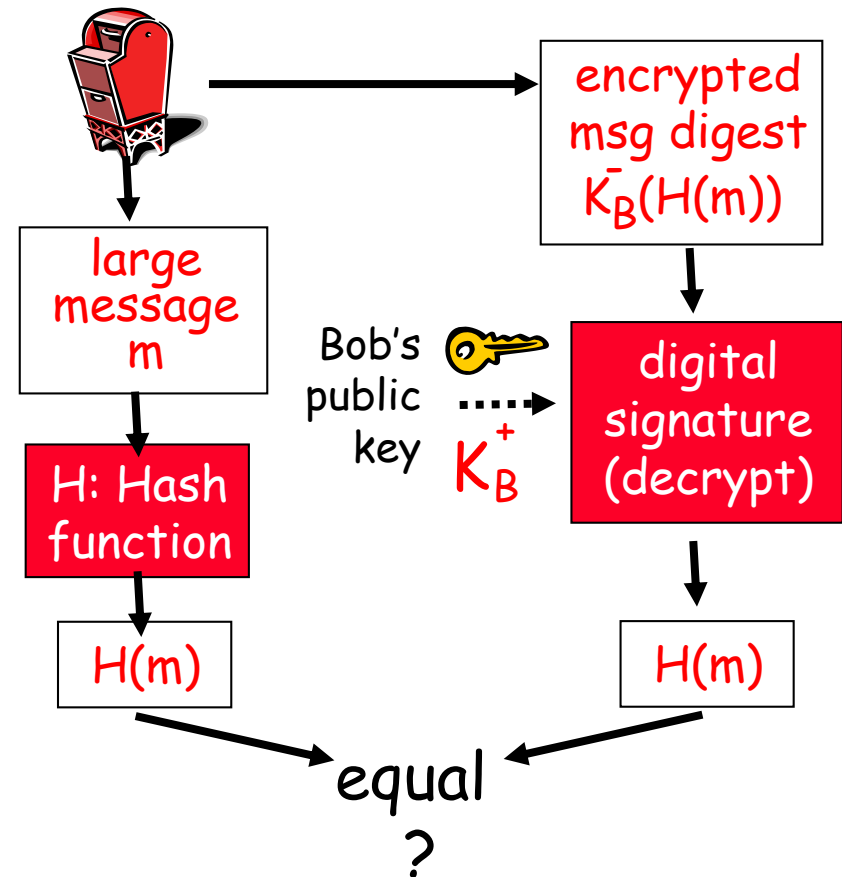
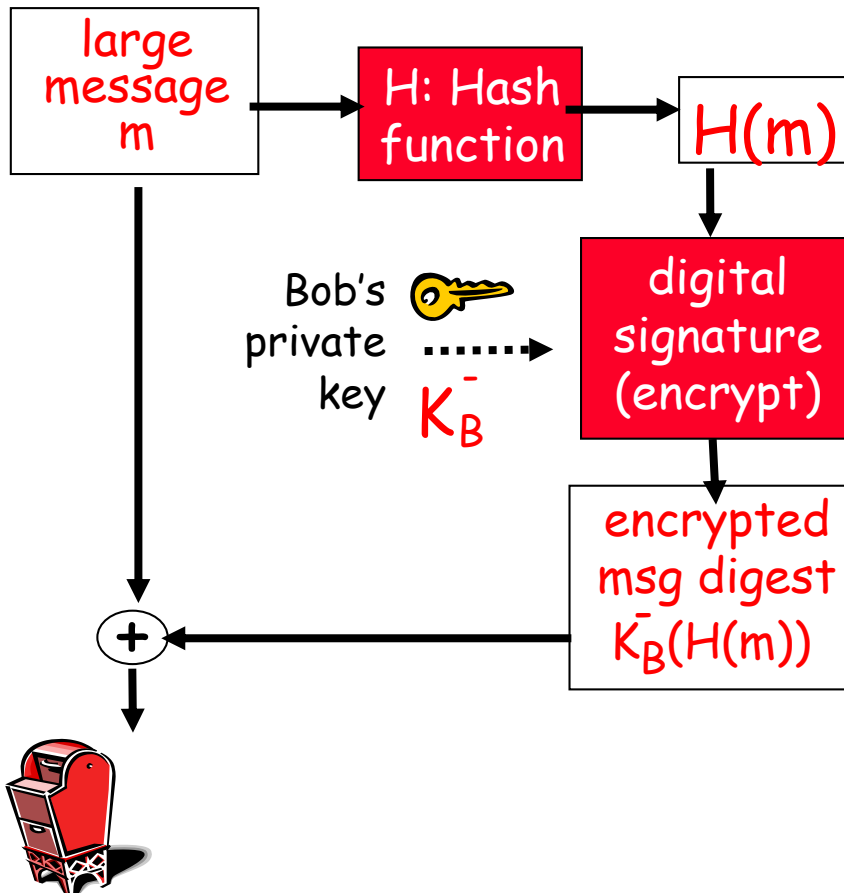
Eigenschaften kryptographischer Hashfunktionen:

- ◆ Abbildung langer Bytefolgen auf kürzere Folge
- ◆ Nicht umkehrbar: Gegeben $x = H(m)$, so ist es sehr aufwändig daraus m zu berechnen
- ◆ Gegeben m und $x = H(m)$, so ist es sehr aufwändig ein $m' \neq m$ zu finden, so dass $x = H(m')$ gilt.
- ◆ Es ist sehr aufwändig, überhaupt zwei m, m' zu finden, so dass $H(m) = H(m')$ gilt

Digitale Signatur = Signierter Message Digest

Bob sendet digital signierte Nachricht

Alice verifiziert die Signatur und die Integrität der signierten Nachricht



Vertrauenswürdige dritte Parteien

Verwaltung symmetrischer Schlüssel:

- ◆ Wie können 2 Parteien im Netz ein Shared Secret etablieren?

Lösung:

- ◆ Key Distribution Center (KDC) wirkt als Mittler zwischen den Parteien
 - statt n^2 Shared Secrets zwischen allen Paaren sind initial nur n Shared Secrets zwischen KDC und den Parteien einzurichten
 - KDC generiert bei Bedarf Sitzungsschlüssel für 2 Parteien

Public Key Zertifizierung:

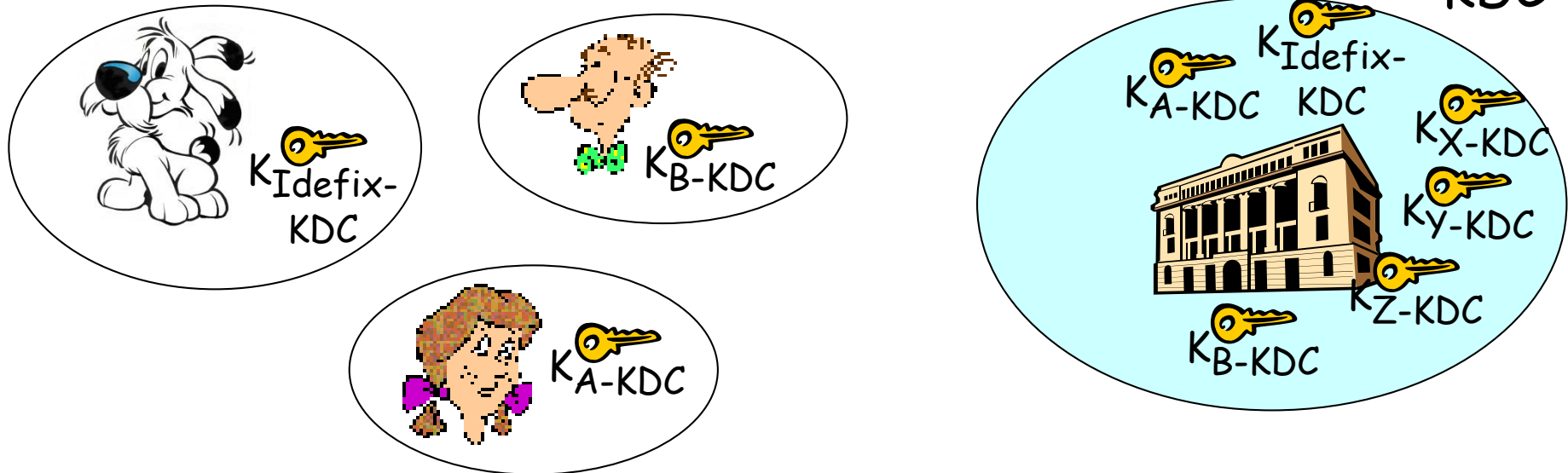
- ◆ Wenn Alice den öffentlichen Schlüssel von Bob erfährt, wie kann sie sicher sein, dass das wirklich Bobs öffentlicher Schlüssel ist

Lösung:

- ◆ Zertifizierungsstelle (Certification Authority CA)

Key Distribution Center (KDC)

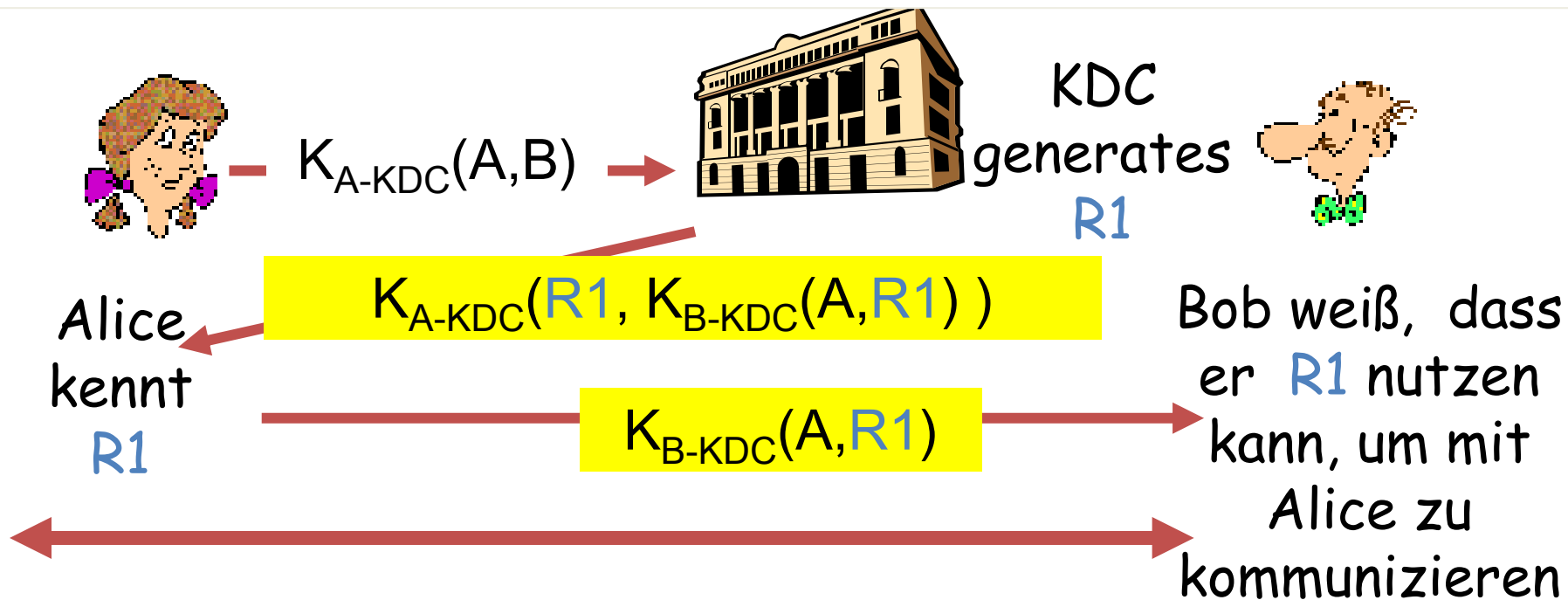
- Alice, Bob brauchen ein Shared Secret zur effizienten sicheren Kommunikation
- **KDC:** Server verwaltet je Partei einen geheimen Schlüssel
- Alice und Bob kennen jeweils ihre eigenen geheimen Schlüssel, K_{A-KDC} K_{B-KDC} , mit deren Hilfe sie mit dem KDC authentifiziert kommunizieren können.
- Wenn Alice eine Sitzung mit Bob durchführen will, lassen sie sich vom KDC einen Sitzungsschlüssel als Shared Secret zwischen Alice und Bob erzeugen



Key Distribution Center (KDC)

Wie erfährt Bob den Sitzungsschlüssel R1?

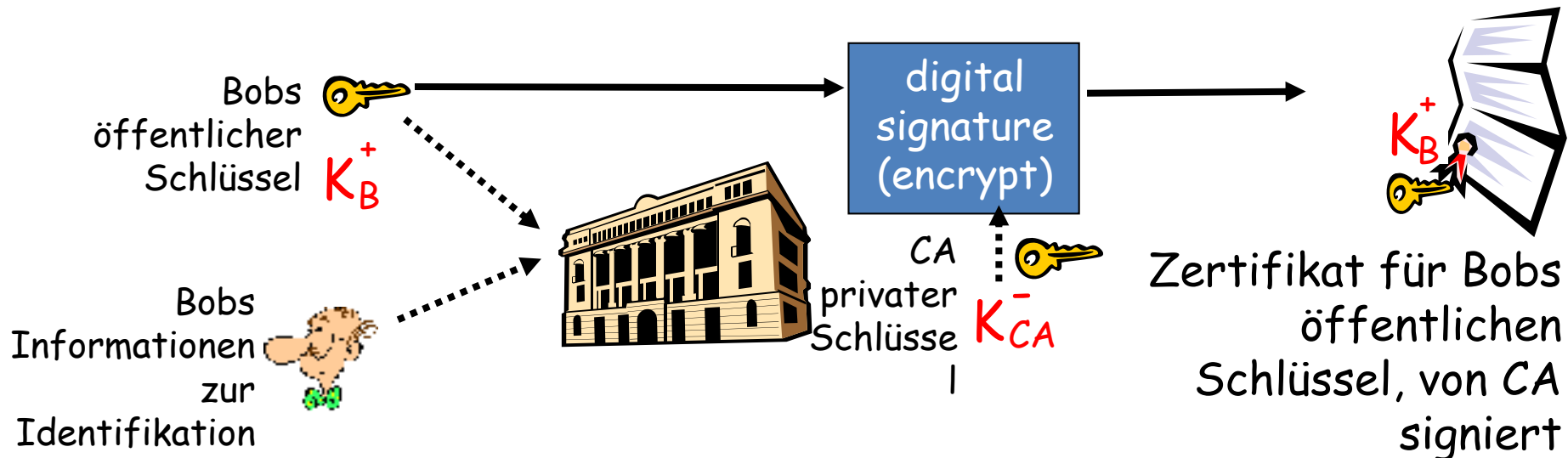
KDC erzeugt "Ticket", das von Alice unveränderbar an Bob weitergegeben wird



Alice und Bob kommunizieren effizient: Sie nutzen **R1** als **Session Key** für die symmetrische Verschlüsselung

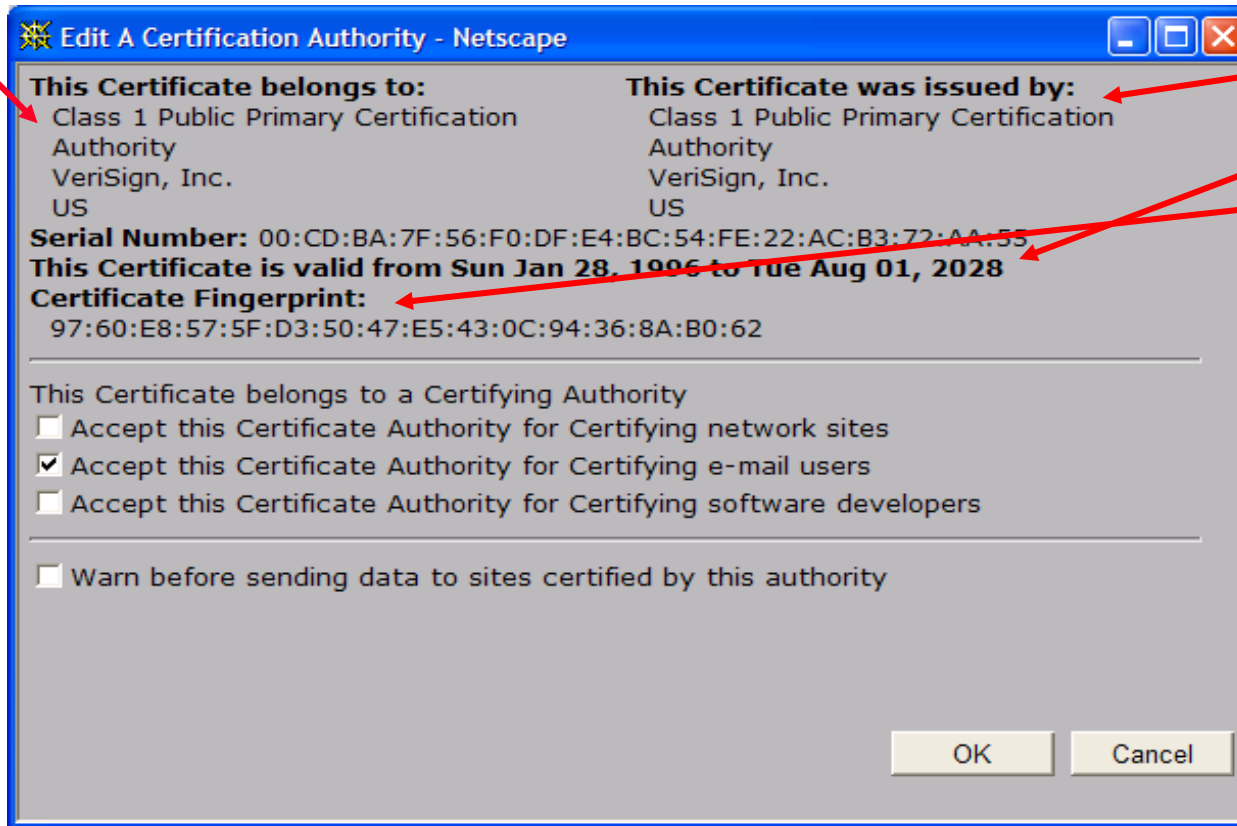
Zertifizierungsstellen (Certification Authorities CAs)

- **Certification Authority (CA):** Verwaltet die Bindung eines öffentlichen Schlüssels an Person / Partei E.
- E registriert seinen öffentlichen Schlüssel bei CA.
 - E weist sich bei CA aus (z.B. mit dem Personalausweis)
 - CA erzeugt einen Datensatz, das Zertifikat, das die Bindung von K_E^+ an E dokumentiert
 - Zertifikat: " K_E^+ ist öffentlicher Schlüssel von E" digital signiert von CA



Inhalt eines Zertifikats

- ◆ Seriennummer (eindeutig für alle Zertifikate derselben CA)
- ◆ Information zur Partei: Name, Art
 - auch (hier nicht sichtbar) öffentlicher Schlüssel sowie Angaben zu unterstützten Kryptoalgorithmen



Info zu CA

Gültigkeitszeitdauer

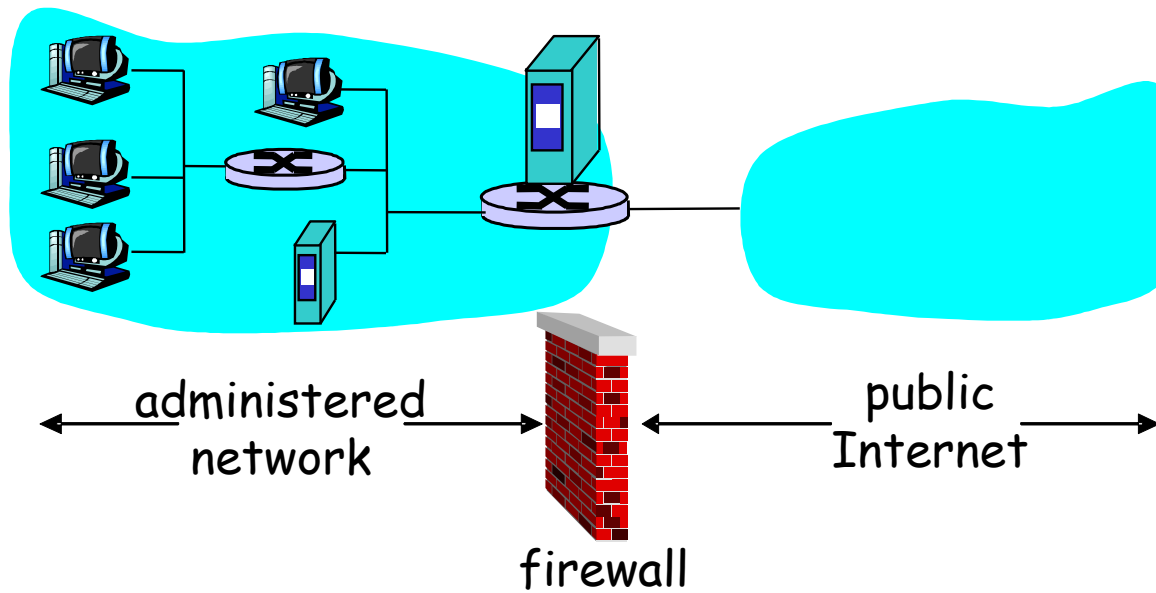
Signatur der CA

Weitere Aufgaben einer CA

- ◆ Zeitstempel
- ◆ Rückruf-Listen

Firewalls

Verkehrskontrolleinrichtung an Grenze eines Firmennetzes zum öffentlichen Netz hin (auch an Innennetzgrenzen zu sensiblen Subnetzen): Lässt manche Kommunikation zu, manche nicht.



Firewalls: Motivation

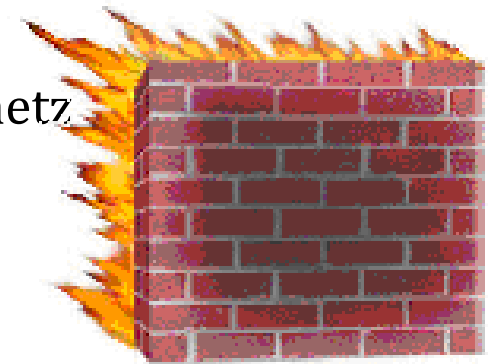
Eigentlich sind Firewalls nicht nötig, weil alle Hosts und Router nur vorgesehene Dienste an vorgesehene Nutzer erbringen sollen und dies durch die Autorisierungs- und Authentifikationsdienste der Rechner kontrolliert wird.

Aber es gibt immer wieder unvorhergesehene Schwachstellen, die aus Programmier- und Administrationsfehlern resultieren.

Deshalb sollen Firewalls zusätzlich unabhängig von den anderen Diensten unerwünschten Verkehr abblocken und damit die Angriffsfläche verkleinern.

Ferner

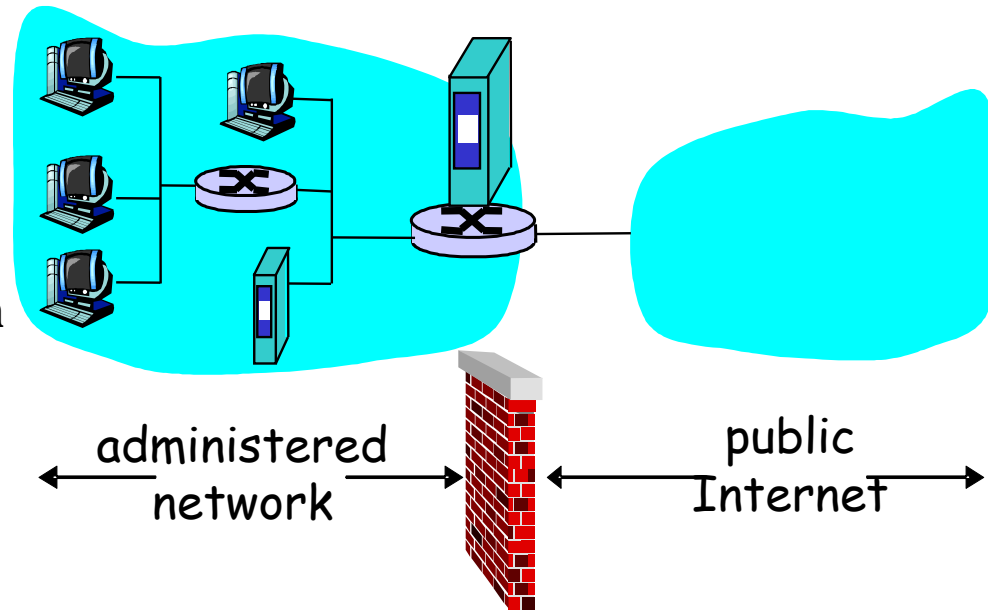
- ◆ Abwehr von Verfügbarkeitsangriffen auf das Innennetz
- ◆ Abwehr von IP-Spoofing-Angriffen
- ◆ Oft in Verbindung mit NAT
- ◆ Oft in Verbindung mit VPN



Firewalls: Architektur

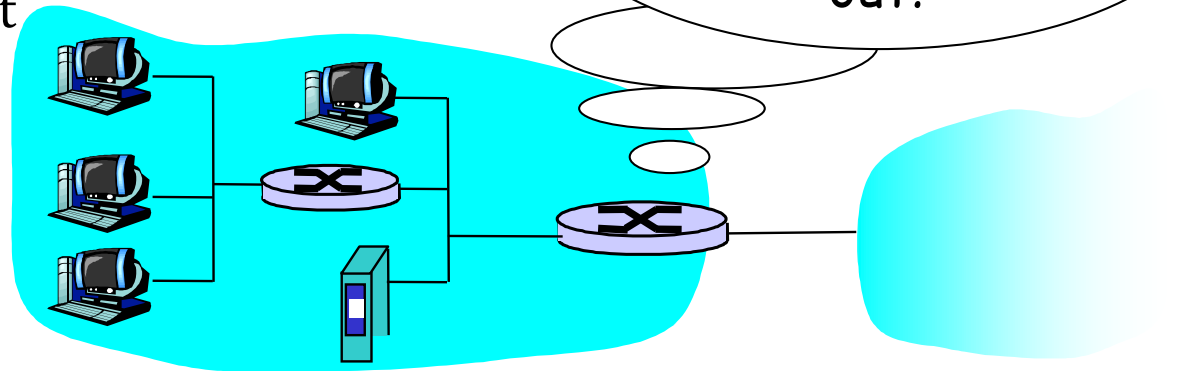
Drei Aspekte

- ◆ **Netztopologie**
 - Innennetz – Außennetz,
Firewall an Verbindungswegen
- ◆ **Filterfunktion**
3 Filtertypen
 - Applikationsfilter
 - Verbindungsfilter
 - Paketfilter (statisch / dynamisch)
- ◆ **Filteranordnung**
 - nur ein Router mit Paketfilter
 - mehrere zusammenwirkende Filter und Knoten
 - » Dual homed Bastion Host
 - » Screened Subnet



Paket-Filter

- ◆ Router, der Innen- und Außennetz verbindet, hat Paketfilterfunktion
- ◆ Liste aus Filterregeln der Form
“Interface, Bedingung über Paket-Header, Aktion”
- ◆ Bedingung:
 - source IP address, destination IP address, TCP/UDP source and destination port numbers
 - ICMP message type, TCP SYN and ACK bits
- ◆ Aktion: Paket durchlassen, verwerfen (mit / ohne Alarm)
- ◆ Statische und dynamische Filter

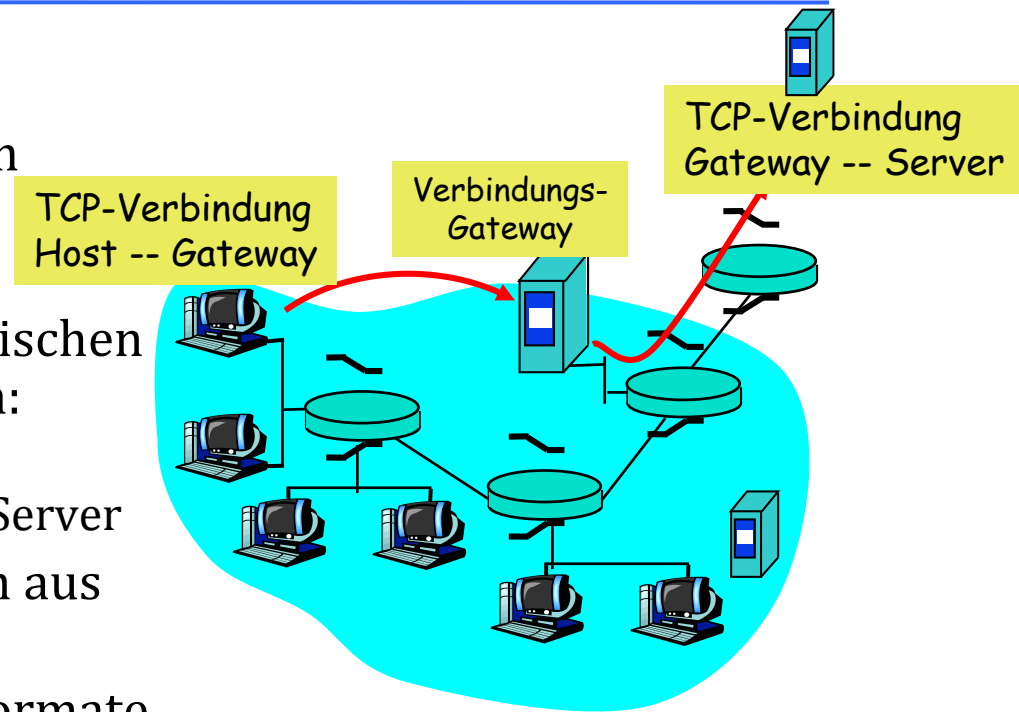


Filterlisten – Aufbau

- Vorne:** Anti-Spoofing Regeln verbieten, dass von außen Pakete mit Innenadressen als Absenderadresse durchkommen
- Mitte:** Nur positive Regeln für den notwendigen Verkehr
- Hinten:** Negative Regeln, die den ganzen Rest verbieten.

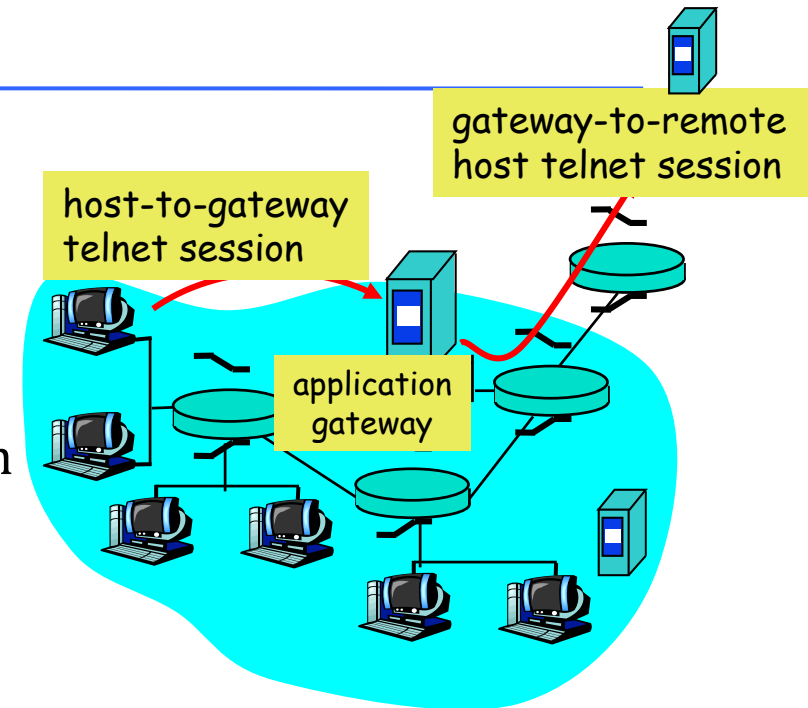
Verbindungsfilter

- ◆ Realisierung durch einen Prozess “Verbindungs-Gateway” auf einem Firewall-Host
- ◆ Es werden keine direkten Transportverbindungen mehr zwischen Außen- und Innennetz zugelassen:
 - Stattdessen 2 Verbindungen:
Client – Gateway und Gateway – Server
- ◆ Gateway packt die TCP-Nutzdaten aus und verpackt sie selbst wieder
- ◆ Prüfung der TCP-Adressen und Formate, Erschweren von Formatfehler- und Segmentierungsattacken
- ◆ Die eigentlichen Anwendungsdaten können nicht untersucht werden, weil das Verbindungsgateway das Anwendungsprotokoll nicht kennt



Applikationsfilter

- ◆ Realisierung durch einen Prozess “Applikationsgateway” auf einem Firewall-Host, z.B. Telnet-Gateway
- ◆ Es werden keine direkten Anwendungsverbindungen mehr zwischen Außen- und Innennetz zugelassen:
 - Stattdessen 2 Verbindungen:
Client – Gateway und Gateway – Server
- ◆ Gateway packt die Anwendungsnutzdaten aus und verpackt sie selbst wieder
- ◆ Gateway kann Anwendungsdaten interpretieren, da speziell für bestimmten Anwendungstyp erzeugt:
 - Nutzerkennungen, Authentifikation und Autorisierung
 - Zusatzdaten (z.B. Mail-Anhänge, Active X, Applets)



Ein Applikationsgateway wird oft auch Applikations-Proxy oder Applikationsfilter genannt

Typische Bedrohungen im Internet (Internet Security Threats)

Mapping und Scanning:

- Vor dem eigentlichen Angriff: Erkunde das Netz, finde heraus, welche Hosts, Dienste, Betriebssysteme vorhanden sind
- ping kann zeigen, welche Host-Adressen vergeben sind (auch Verzeichnisse sind nützlich)
- Port-Scanning: Versuch, zu jedem TCP Port eine Verbindung aufzubauen bzw. jeden UDP-Port anzusprechen
Kommt eine Reaktion, welche?
Bekannte Schwachstellen und Angriffsmuster durchspielen.
 - » nmap (<http://www.insecure.org/nmap/>) mapper: “network exploration and security auditing”
- Ferner: Versuch, sich einzuloggen, Versuch FTP-Server-Account anzusprechen. Nutzernamen und Passwörter raten.
Standardmäßig eingerichtete Accounts testen.

Schutzmaßnahmen?

Internet Security Threats: Schutzmaßnahmen

Verkleinere Angriffsfläche

- Firewalls
- Auf Desktop-PC: Personal Firewall
- Gehärtete Konfiguration

Bemerke Besonderheiten

- Log-Erzeugung und Prüfung (Logging and Audit)
- Verkehrsstatistiken führen und überwachen
- Systemkonfiguration und Dateien überwachen (Tripwire)
- IDS – Automatische Angriffserkennung (Intrusion Detection Systeme)

Entferne Schwachstellen

- Aktualisiere Systeme, wenn Patches verfügbar
- Scanne selbst, um Schwachstellen zu finden

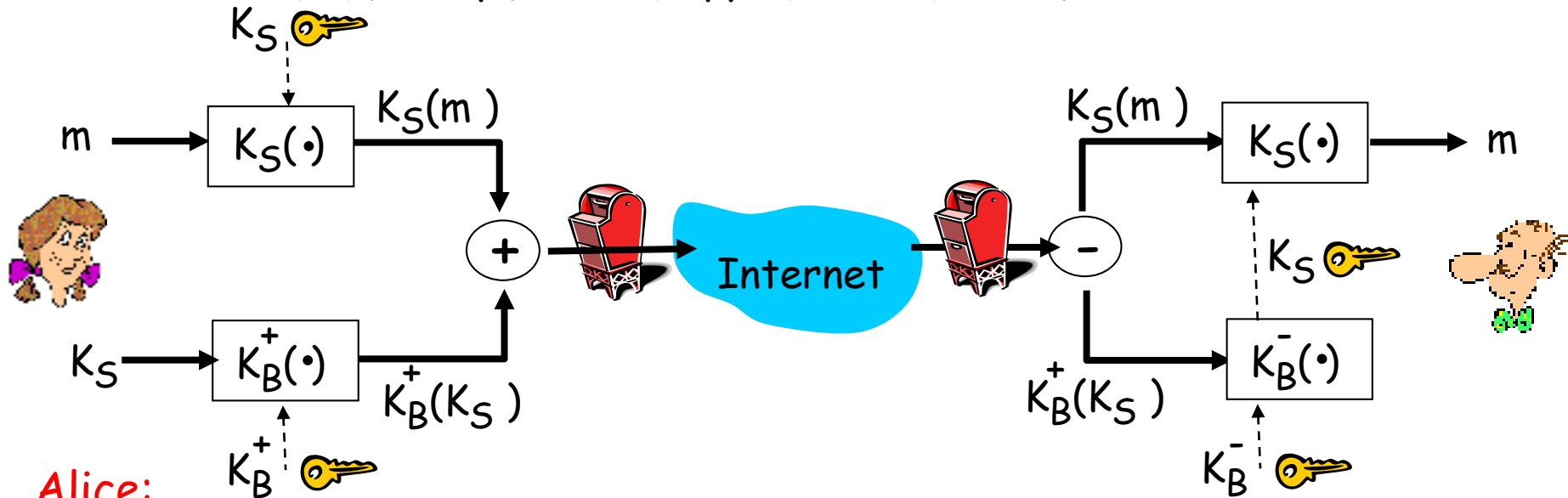
Wehre bösartigen Code ab

- Virens Scanner, Firewall, gehärtete Konfiguration, eingeschränkte Nutzeraccounts



Sichere E-Mail: Vertraulichkeit

- ❑ Alice will vertrauliche Mail m an Bob senden
- ❑ Bob hat einen zertifizierten öffentlichen Schlüssel

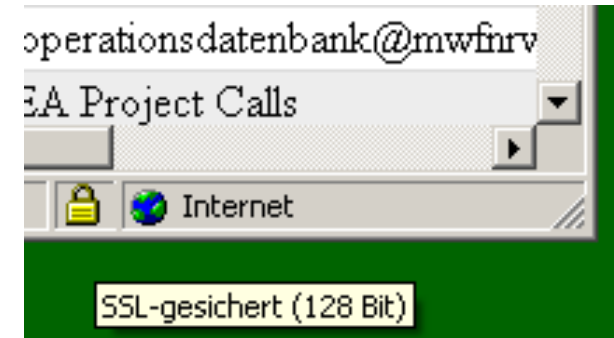


Alice:

- ❑ Prüft Bobs Zertifikat: Gültig?
- ❑ Generiert per Zufallsgenerator *symmetrischen* Secret Key K_S
- ❑ Verschlüsselt Nachricht mit K_S (Effizienz)
- ❑ verschlüsselt K_S mit Bobs öffentlichem Schlüssel
- ❑ sendet beides, $K_S(m)$ und $K_B(K_S)$, in E-Mail an Bob
- ❑ Bob entschlüsselt erst $K_B(K_S)$, dann $K_S(m)$

TLS / SSL: Transport Layer Security / Secure Socket Layer

- ◆ “Aufsatz” auf TCP-Verbindungen:
 - (optionale) Authentifikation der Partnerprozesse
 - Vertraulichkeit, Integrität und Authentizität der Nachrichten per Verschlüsselung



- ◆ in Anwendungsprozessen zu implementieren, z.B. im Web-Browser und im Web-Server (shttp)

- ◆ Betrieb in 2 Phasen

1. Vorbereitung

- Authentifikation, Kryptoparameterabstimmung, Sitzungsschlüsselaustausch

2. Kommunikation “Wie TCP” über Sockets

- ◆ Server Authentifikation:

- SSL-Enabled Browser enthält Zertifikate vertrauenswürdiger CAs.
- Browser fordert von einem kontaktierten Server dessen Zertifikat an, das von einer dieser CAs ausgestellt sein muss
- Browser prüft mit dem CA-Zertifikat, ob das Server-Zertifikat gültig ist (Problem: Rückrufe)

- ◆ Schauen Sie mal in die Einstellungen Ihres Browsers um die CA-Liste einzusehen

IPsec: Network Layer Security

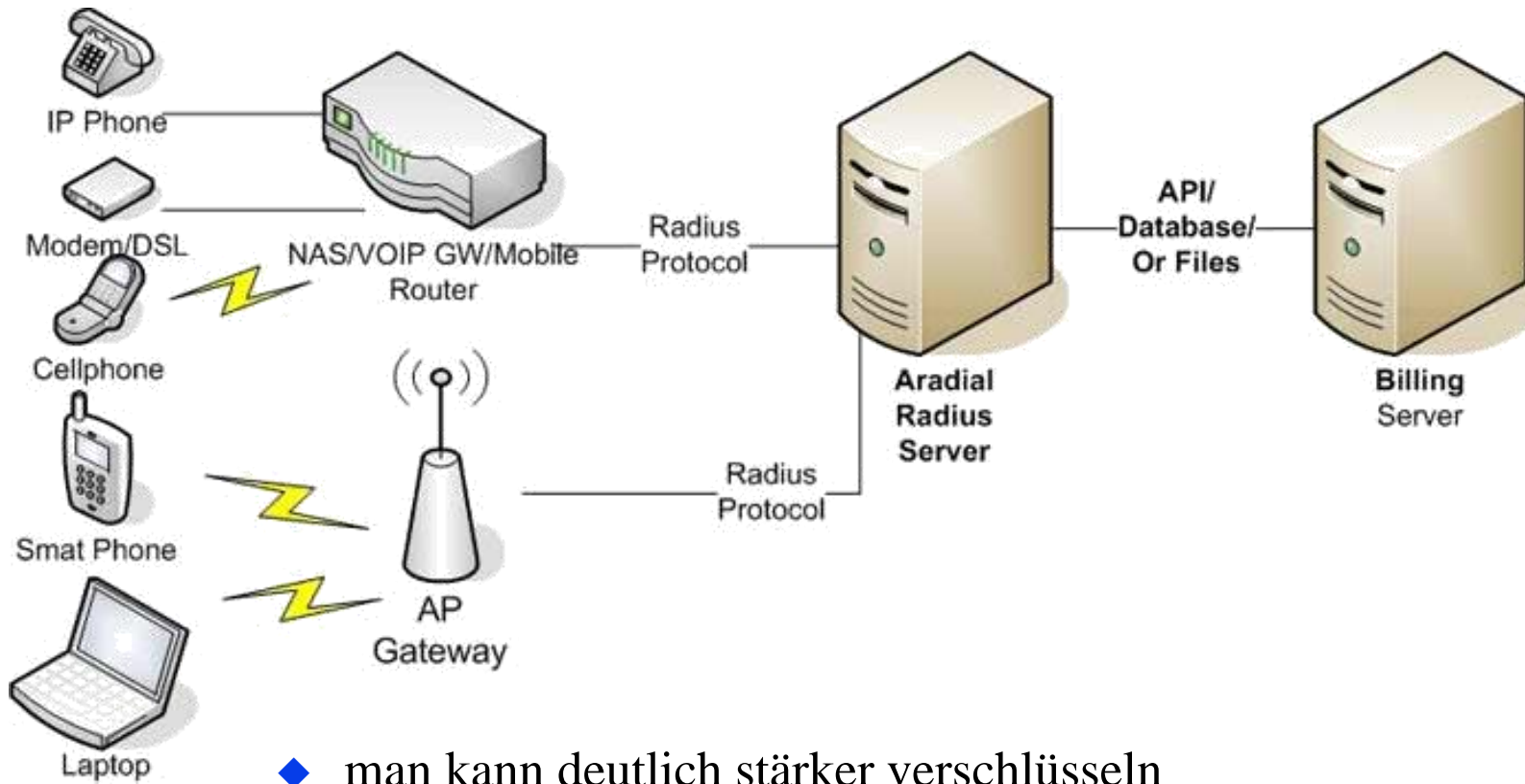


- ◆ IPsec ist im Protokoll IP V6 enthalten
Es kann auch in IP V4 eingesetzt werden
- ◆ IPsec sichert den IP-Paketaustausch zwischen Netzknoten
- ◆ IPsec wird als “Aufsatz” auf IP im Kern des Host-Betriebssystems implementiert und durch Administrationsparameter aktiviert
 - Vorteil: Keine Änderungen oder Ergänzungen der Anwendungsprozesse nötig
 - Nachteil: Knoten und nicht individuelle Anwendungsprozesse bilden die Endpunkte der gesicherten Kommunikation
- ◆ Problem:
 - IP ist verbindungslos/sitzungslos
 - Effiziente Kommunikation verlangt Sitzungsschlüssel als Shared Secret
- ◆ Lösung: Konzept der Security Association SA
 - Je Paar aus Quelle und Ziel (also auch je Richtung) wird SA definiert
 - Alle passenden IP-Pakete gehören zur SA, solange SA existiert
- ◆ Betrieb ähnlich SSL: 2 Phasen
 - SA Aufbau
 - Paketaustausch
- ◆ SA-Aufbau wird durch Knotenadministration gesteuert:
Security Policy Definition (SPD) legt für “Quelle → Ziel” fest, ob und mit welchen Parametern eine SA einzurichten ist, so dass die IP-Pakete, die diesem Muster folgen, nur innerhalb einer solchen SA ausgetauscht werden.

IEEE 802.11 Wireless LAN – Security

- ◆ *WLAN-Frames können leicht abgehört werden*
 - Funkwellen halten sich nicht an die Grundstücksgrenzen
 - es gibt Richtantennen
- ◆ **Sicherheitsfunktionen**
 - Authentifikation und Verschlüsselung
- ◆ **Wired Equivalent Privacy (WEP): Ein schwacher Versuch**
 - Authentifikation a la *ap4.0*, *Shared Secret* und *Challenge Response* basiert
 - » Host sendet Request an Access Point, der antwortet mit 128-Bit N_{once}
 - » Host sendet verschlüsselte N_{once} zurück
 - Keine dynamische Schlüsselverteilung
 - Es gibt für Access Point und alle Hosts ein Gruppen-“Shared Secret“
Daraus werden alle benötigten Schlüssel abgeleitet.
 - Verschlüsselung ist relativ leicht zu brechen

802.11i: Verbesserte Sicherheit im WLAN



- ◆ man kann deutlich stärker verschlüsseln
- ◆ dynamische Schlüsselverteilung wird unterstützt
- ◆ bindet einen separaten Authentifikationsserver ein, der nicht mit dem Access Point zusammenfällt (z.B. Kerberos, RADIUS)