Karlsruher Institut für Technologie

Vorlesung Einführung in Rechnernetze
9. Sicherheit

Prof. Dr. Martina Zitterbart

Dipl.-Inform. Martin Florian, Markus Jung (M.Sc.), Matthias Flittner (M.Sc.)

[zitterbart | florian | m.jung | flittner]@kit.edu

Institut für Telematik, Prof. Zitterbart

Kapitelübersicht

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Einführung
2. Schutzziele
3. Angriffe
4. Bausteine
5. Protokolle

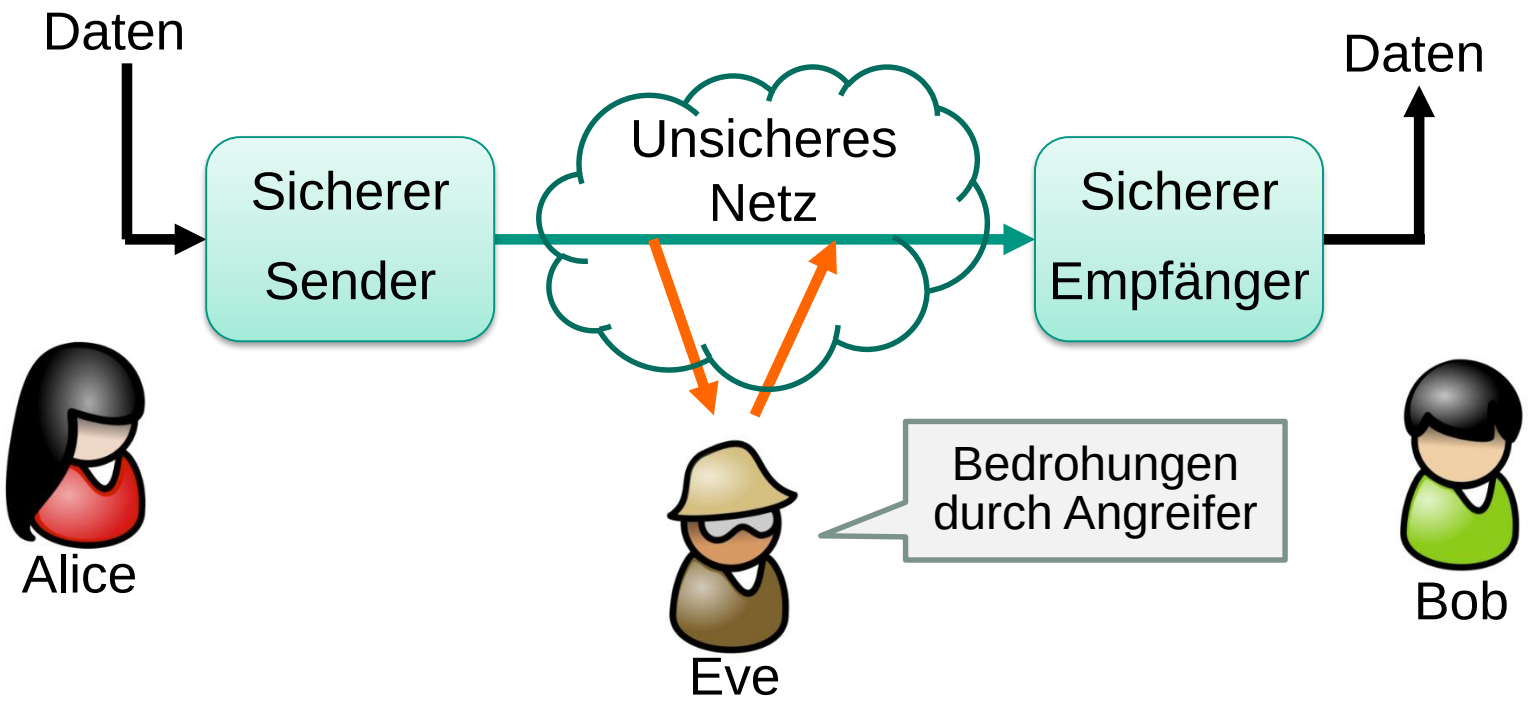
Kapitelübersicht

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Einführung
2. Schutzziele
3. Angriffe
4. Bausteine
5. Protokolle

9.1 Einführung

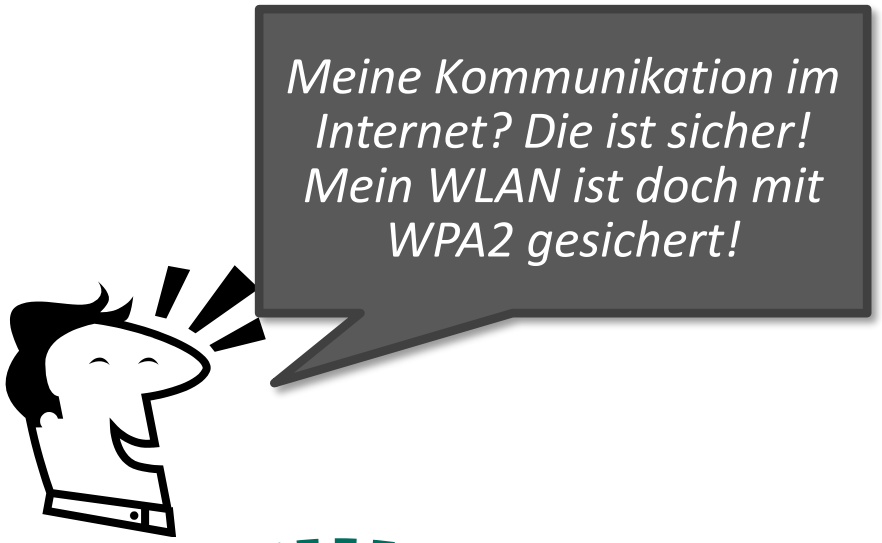
- Alice und Bob wollen „*sicher*“ über unsicheres Netz kommunizieren

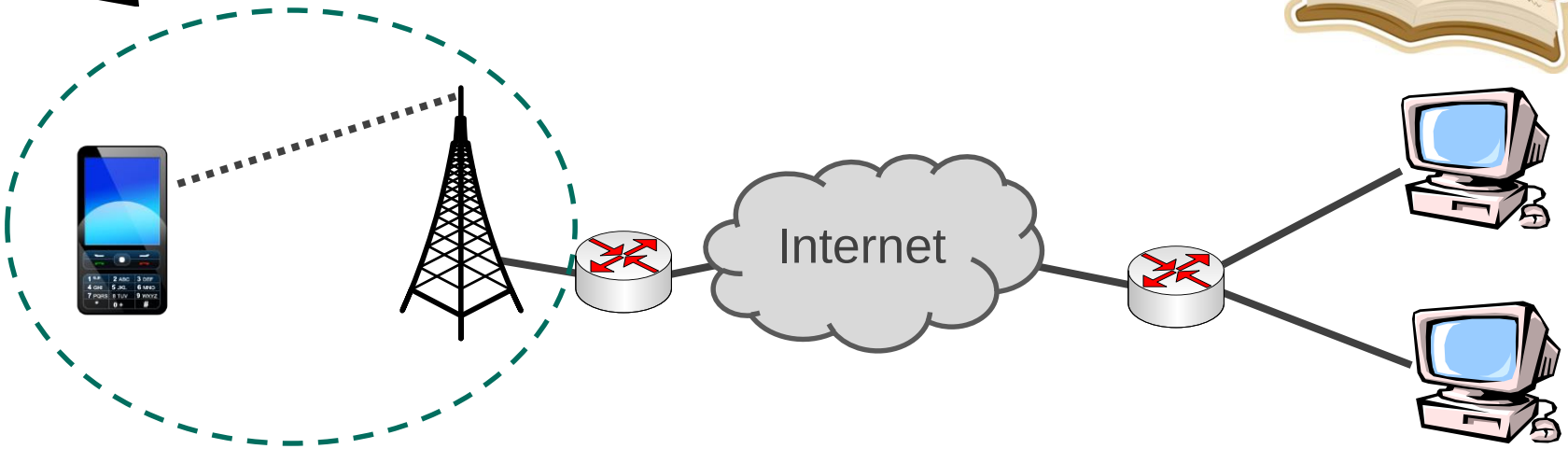


Was heißt „*sicher*“?
Welche Bedrohungen können vorliegen?
Und wie kann Sicherheit realisiert werden?

Wie/was/wo wird geschützt? Fallstricke

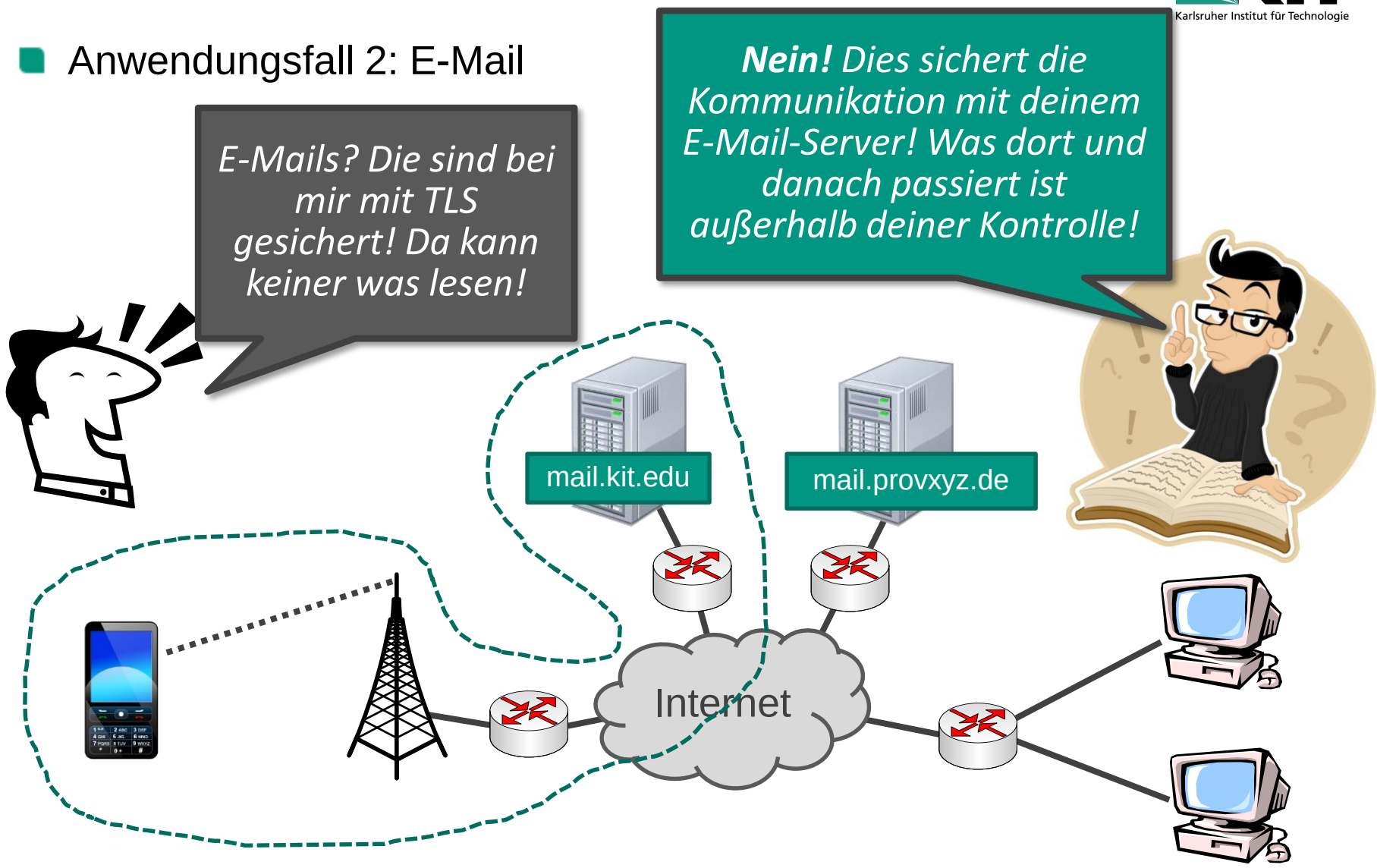
- Anwendungsfall 1: Kabelloser Zugang





Wie/was/wo wird geschützt? Fallstricke

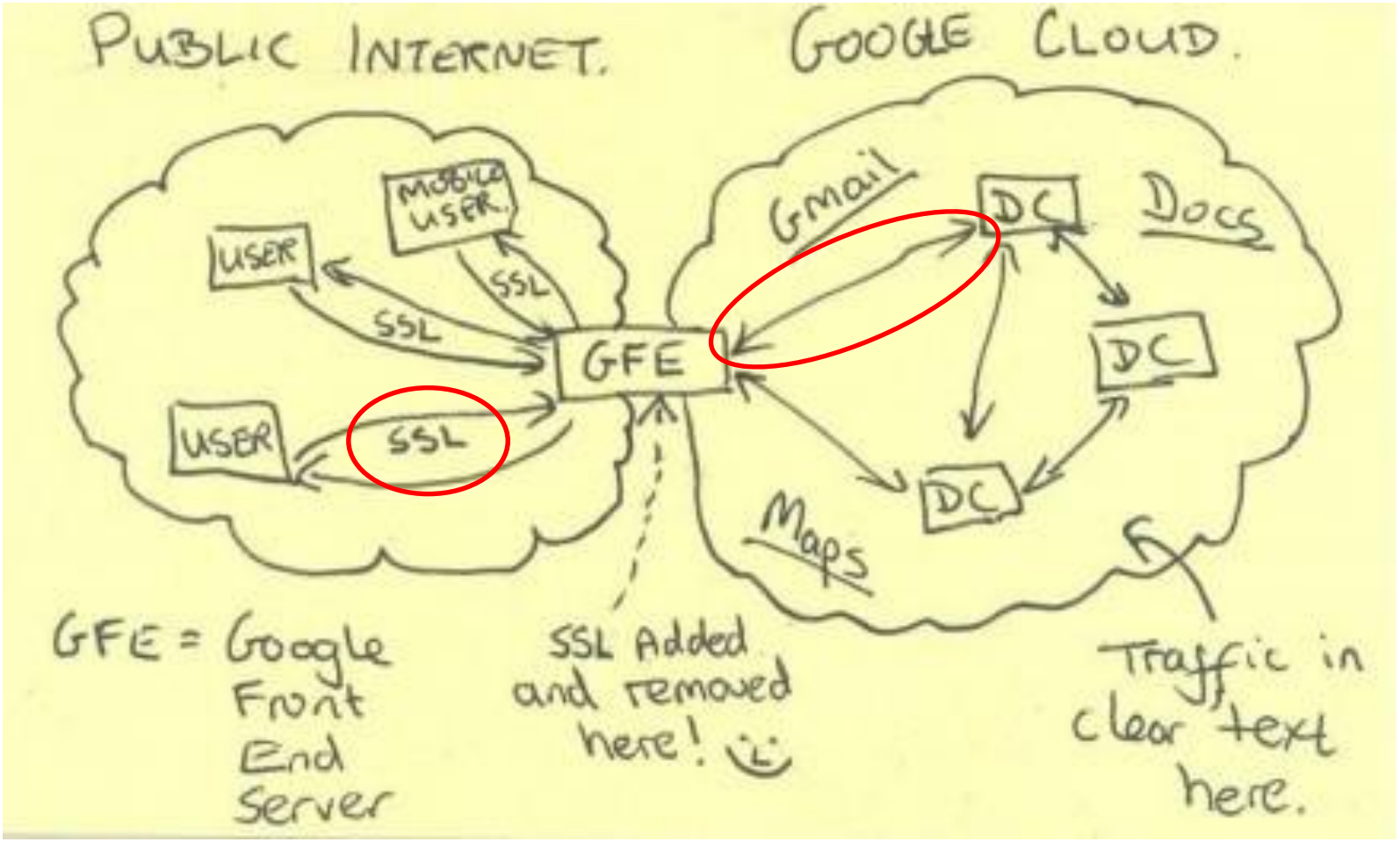
- Anwendungsfall 2: E-Mail



Oder so: scheinbare Sicherheit…

- SSL zum Frontend, Klartext im Backend

[WaPo13]



NSA und die Google-Infrastruktur

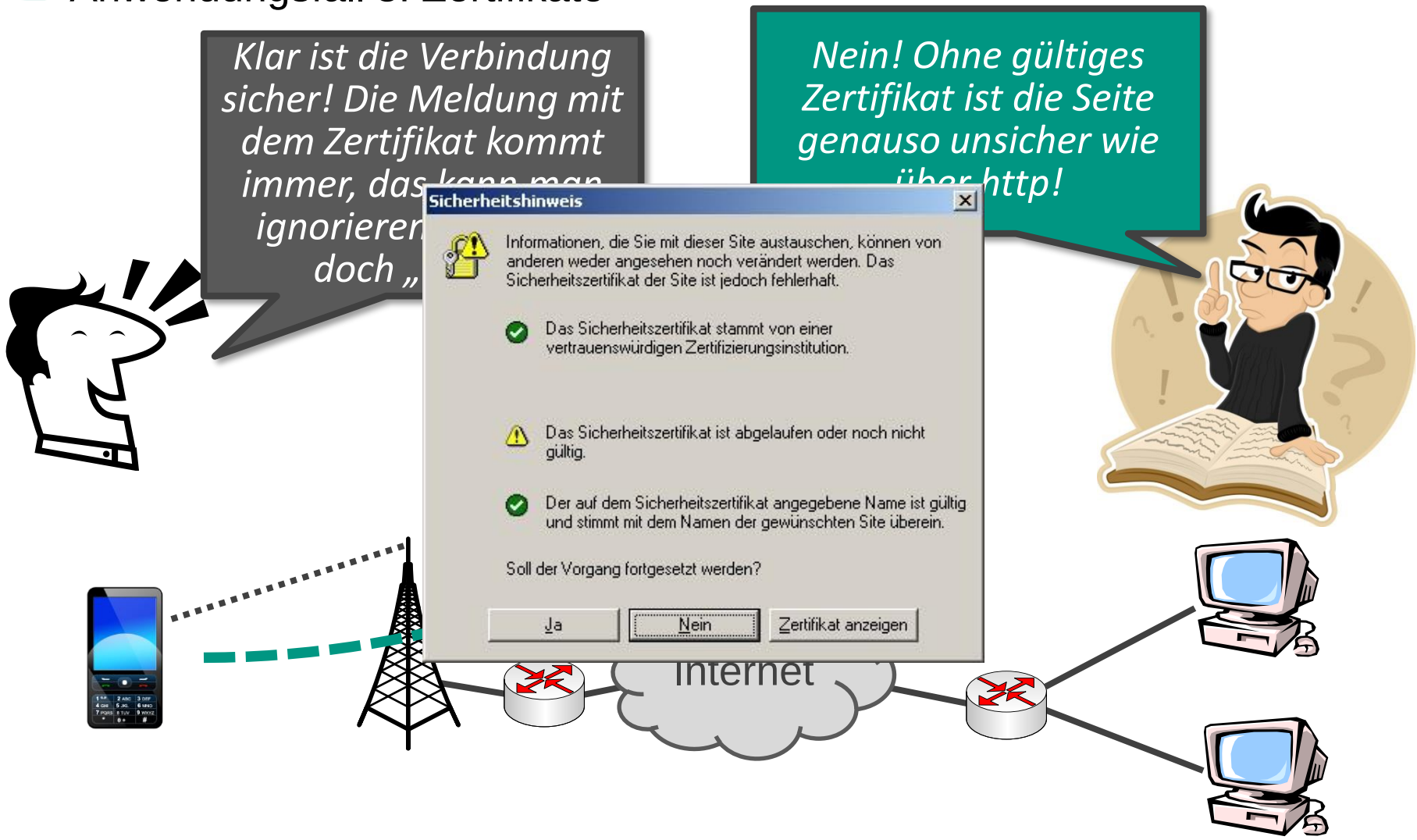
- Nutzer wiegt sich in Sicherheit
  - TLS-Verbindung zum Google-Server, z.B. bei Mailabruf
- TLS aber nur bis zum Frontend-Server
  - danach unverschlüsselt innerhalb des Google-Rechenzentrums

… NSA hat einfach dort die Daten abgegriffen!


[WaPo13]

Wie/was/wo wird geschützt? Fallstricke

- Anwendungsfall 3: Zertifikate



Netzsicherheit: Vielzahl an Themen und Begriffen

- Signatur
- Verschlüsselung
- Hashfunktion
- Message Authentication Code
- Schlüsselaustausch

Bausteine

- Vertraulichkeit
- Integrität
- Authentizität
- Privatheit
- (Nicht)-Abstreitbarkeit

Schutzziele

- Abhören
- Einfügen
- Manipulieren
- Man-in-the-Middle
- Replay
- Denial-of-Service

Angriffe

- TLS/SSL
- IPsec
- Kerberos

System/Protokoll

System/Protokoll verwendet Bausteine um Schutzziele zu realisieren und vor Angriffen zu schützen

- Signatur
- Verschlüsselung
- Hashfunktion
- Message Authentication Code
- Schlüsselaustausch

Bausteine

- Vertraulichkeit
- Integrität
- Authentizität
- Privatheit
- (Nicht)-Abstreitbarkeit

9.2 Schutzziele

- Abhören
- Einfügen
- Manipulieren
- Man-in-the-Middle
- Replay
- Denial-of-Service

Angriffe

- TLS/SSL
- IPsec
- Kerberos

System/Protokoll

System/Protokoll verwendet Bausteine um Schutzziele zu realisieren und vor Angriffen zu schützen

9.2 Schutzziele

- Schutzziel
  - Anforderungen an eine Komponente oder ein Systems, die erfüllt werden müssen, um schützenswerte Güter vor Bedrohungen zu schützen

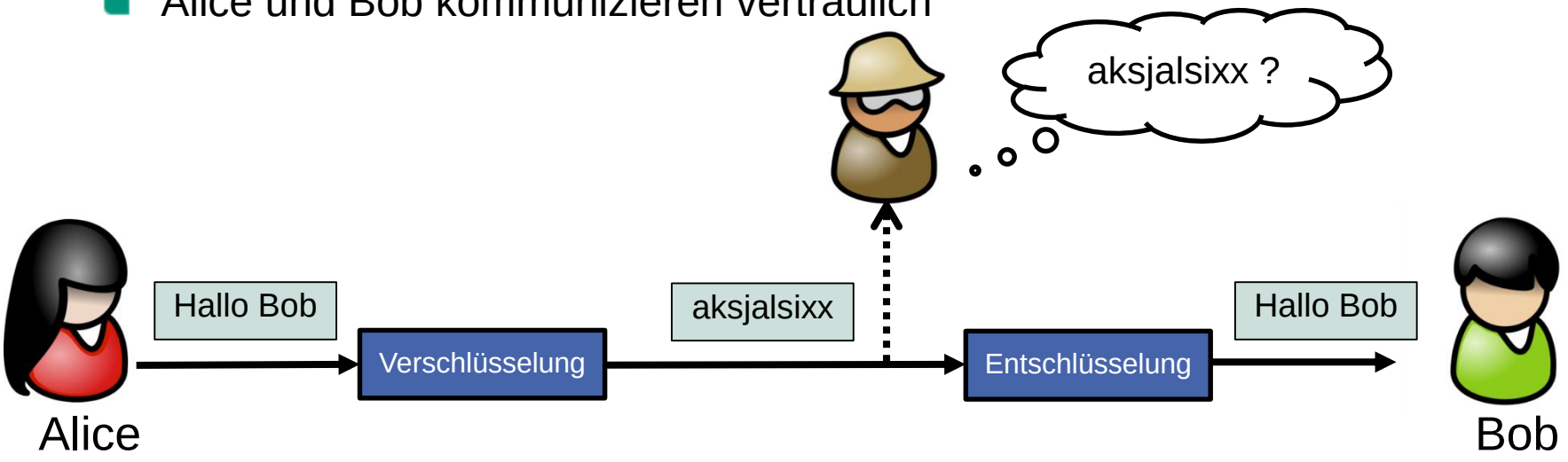
- Häufige Kategorisierung in
  - **C**onfidentiality (Vertraulichkeit)
  - **I**ntegrity (Integrität)
  - **A**vailability (Verfügbarkeit)

- Weitere Schutzziele
  - Authentizität
  - Privatheit
  - (Nicht-)Abstreitbarkeit

Schutzziel Vertraulichkeit

C I A

- Vertraulichkeit
  - Ein System bewahrt Vertraulichkeit, wenn es keine unautorisierte Informationsgewinnung ermöglicht
- Baustein zur Umsetzung von Vertraulichkeit
  - Symmetrische oder asymmetrische Verschlüsselung
- Beispiel
  - Alice und Bob kommunizieren vertraulich



Schutzziel Integrität

- Integrität
  - Ein System bewahrt *starke* Integrität, wenn es nicht möglich ist, Daten unautorisiert zu manipulieren
  - Ein System bewahrt *schwache* Integrität, wenn unautorisierte Manipulationen an Daten nicht unbemerkt möglich sind

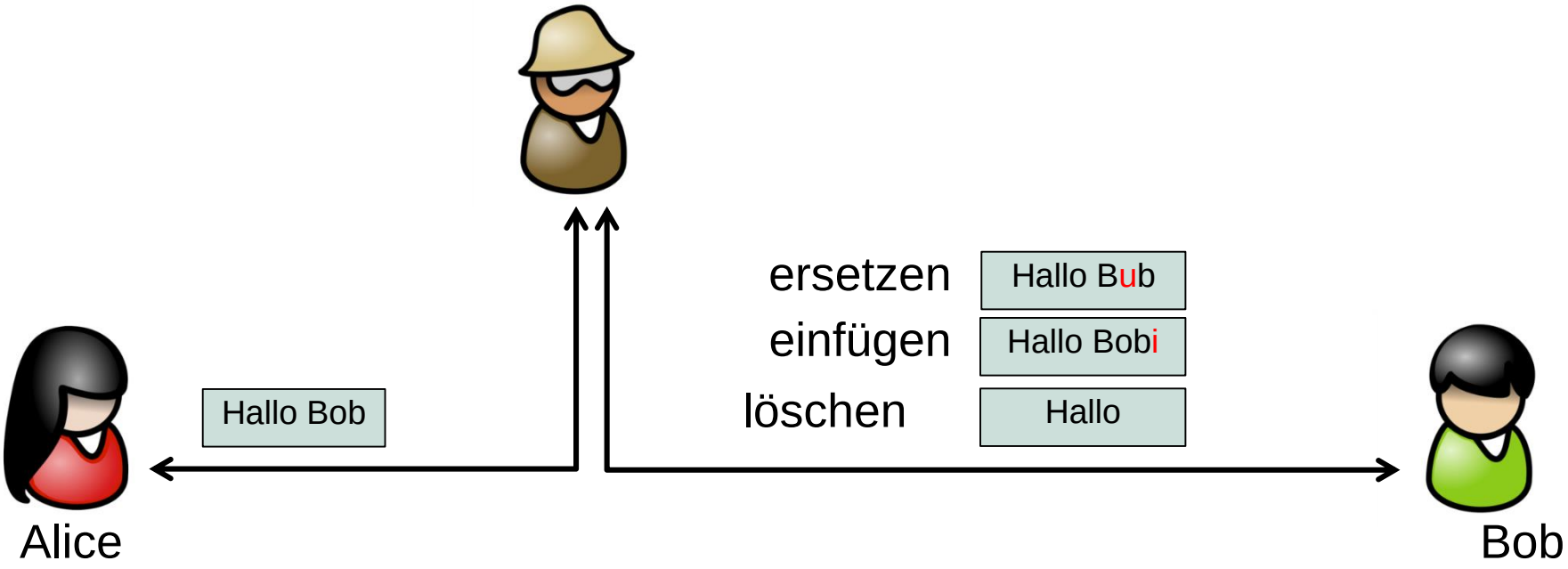
- Bausteine zur Umsetzung von Integrität
  - Tamper Proof Module
  - Message Authentication Codes (MAC)
    - Hashfunktionen mit Schlüssel

… nur schwache Integrität ?

- Manipulationen in vielen Fällen nicht zu verhindern
- Sollten dann wenigstens nicht unbemerkt bleiben

Schutzziel Integrität

- Mögliche Manipulationen z.B.
  - Ersetzen von Daten
  - Einfügen in Daten
  - Löschen von Daten



Schutzziel Authentizität

- *Was ist Integrität der Daten wert, wenn nicht sicher ist von wem die Daten kommen?*

- Authentizität
  - Angegebene Quelle von Daten entspricht tatsächlicher Quelle + Datenintegrität
  - Echtheit von *Subjekten*
    - Bob will sicherstellen, dass er wirklich mit Alice kommuniziert
      → Authentifikation
  - Echtheit von *Daten*
    - Bob will sicherstellen, dass die Daten wirklich von Alice sind
- Bausteine zur Umsetzung von Authentizität
  - Zertifikate, Signaturen, gemeinsames Geheimnis

Schutzziel Verfügbarkeit

- Verfügbarkeit
  - Beschreibt, in welchem Maße die Funktionalität des Systems von berechtigten Subjekten unabhängig von Einflüssen in Anspruch genommen werden kann
  - Beschreibt in Gegenwart von Angreifern, in welchem Maße die Funktionalität des Systems von berechtigten Subjekten unabhängig von gezielten Einflüssen in Anspruch genommen werden kann

- Signatur
- Verschlüsselung
- Hashfunktion
- Message Authentication Code
- Schlüsselaustausch

Bausteine

- Vertraulichkeit
- Integrität
- Authentizität
- Privatheit
- (Nicht)-Abstreitbarkeit

Schutzziele

- Abhören
- Einfügen
- Manipulieren
- Man-in-the-Middle
- Replay
- Denial-of-Service

9.3 Angriffe

- TLS/SSL
- IPsec
- Kerberos

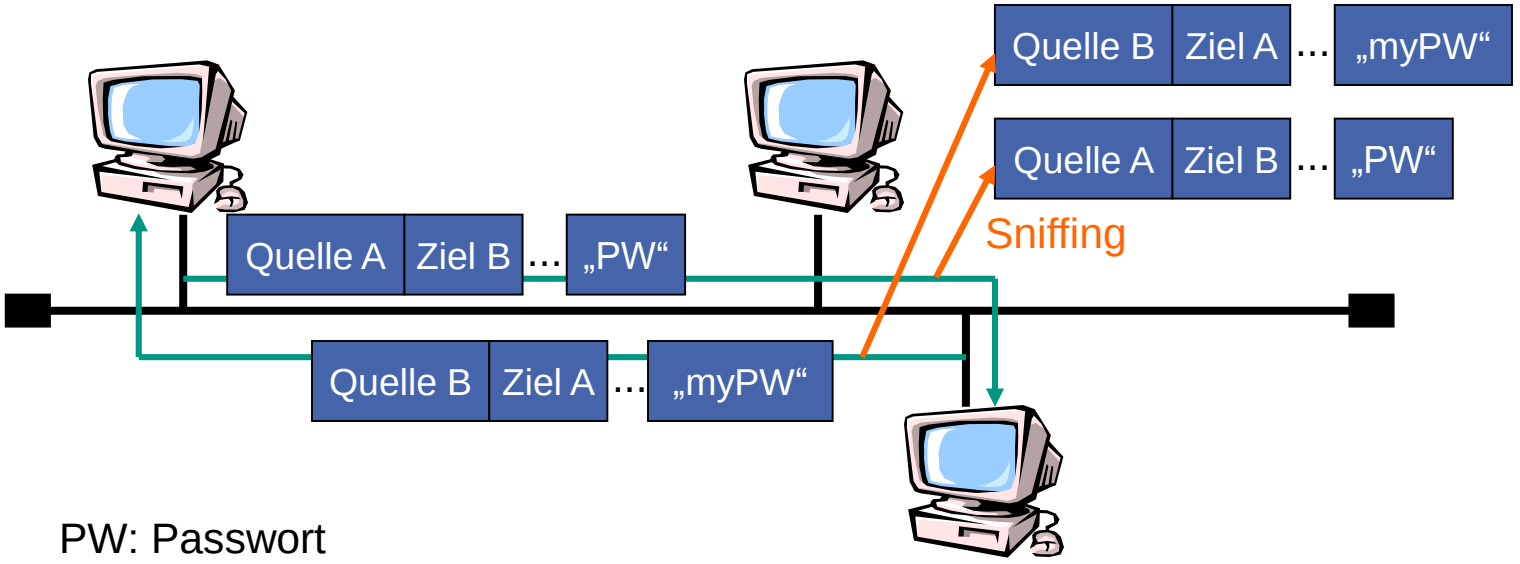
System/Protokoll

System/Protokoll verwendet Bausteine um Schutzziele zu realisieren und vor Angriffen zu schützen

Beispiel für Angriff: Abhören von Daten

- Empfang aller Dateneinheiten, die den Netzanschluss des Systems passieren
  - Besonders leicht in Broadcast-Netzen, z.B. Ethernet
    - Bsp.: Ethernet-Adapter auf Promiscuous Mode stellen
  - … auch als Sniffing bezeichnet

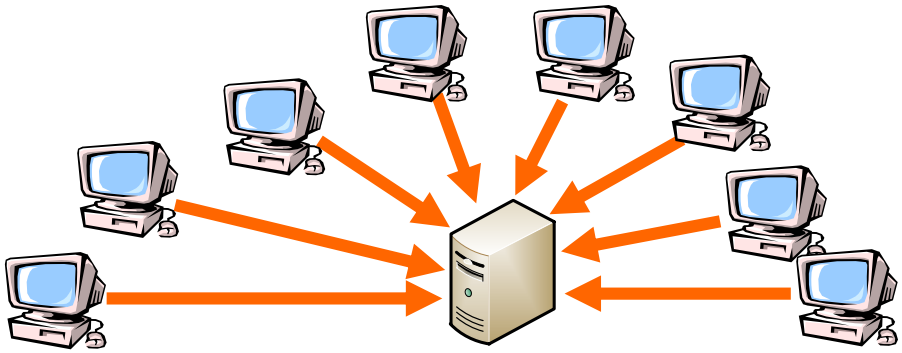
- Z.B. Mitprotokollieren von Daten und Klartextpasswörtern
  - Telnet, FTP, etc.



PW: Passwort

Beispiel für Angriff: Denial-of-Service (DoS)

- Prinzip
  - Möglichst viel Arbeit für die Infrastruktur erzeugen, so dass diese ihren normalen Aufgaben nicht nachgehen kann



- Beispiel: SYN-Flooding
  - Senden von TCP-SYN-Dateneinheiten zu einem Server
    - IP-Adressen gefälscht
  - Server allokiert Ressourcen
    - partiell geöffnete Verbindungen

Pingo!

- http://pingo.upb.de/6466

Pingo

http://pingo.upb.de/

- Signatur
- Verschlüsselung
- Hashfunktion
- Message Authentication Code
- Schlüsselaustausch

9.4 Bausteine

- Vertraulichkeit
- Integrität
- Authentizität
- Privatheit
- (Nicht)-Abstreitbarkeit

Schutzziele

- Abhören
- Einfügen
- Manipulieren
- Man-in-the-Middle
- Replay
- Denial-of-Service

Angriffe

- TLS/SSL
- IPsec
- Kerberos

System/Protokoll

System/Protokoll verwendet Bausteine um Schutzziele zu realisieren und vor Angriffen zu schützen

9.4 Kryptografische Bausteine

- Verschlüsselung
  - Symmetrische Verschlüsselung
    - Kommunikationspartner verfügen über selben Schlüssel zum Ver- und Entschlüsseln
    - Hohe Effizienz, da meist einfache Operationen (Bit-Shift, XOR)
  - Asymmetrische Verschlüsselung
    - Kommunikationspartner verfügen über öffentliche und private Schlüssel
    - Basis: Faktorisierung von Zahlen gilt als schwer
- Integritätssicherung
  - Kryptografische Hashfunktion
    - Verwendet symmetrische Kryptografie
  - Digitale Signatur
    - Verwendet asymmetrische Kryptografie
- Schlüsselaustausch

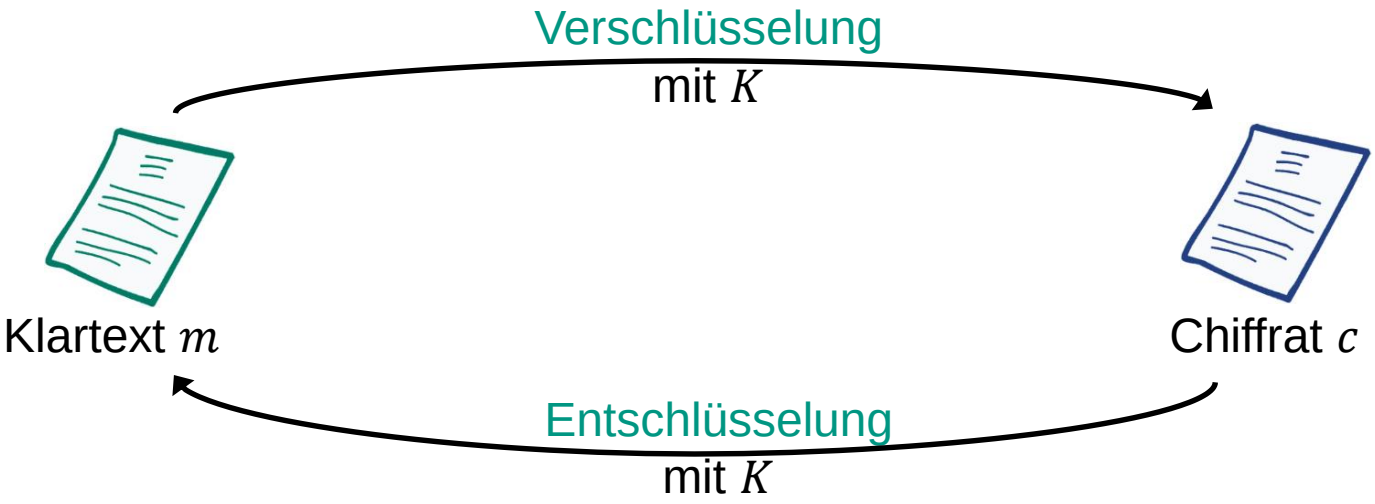
9.4.1 Verschlüsselung

- Symmetrische Verschlüsselung

- Asymmetrische Verschlüsselung

Symmetrische Verschlüsselung

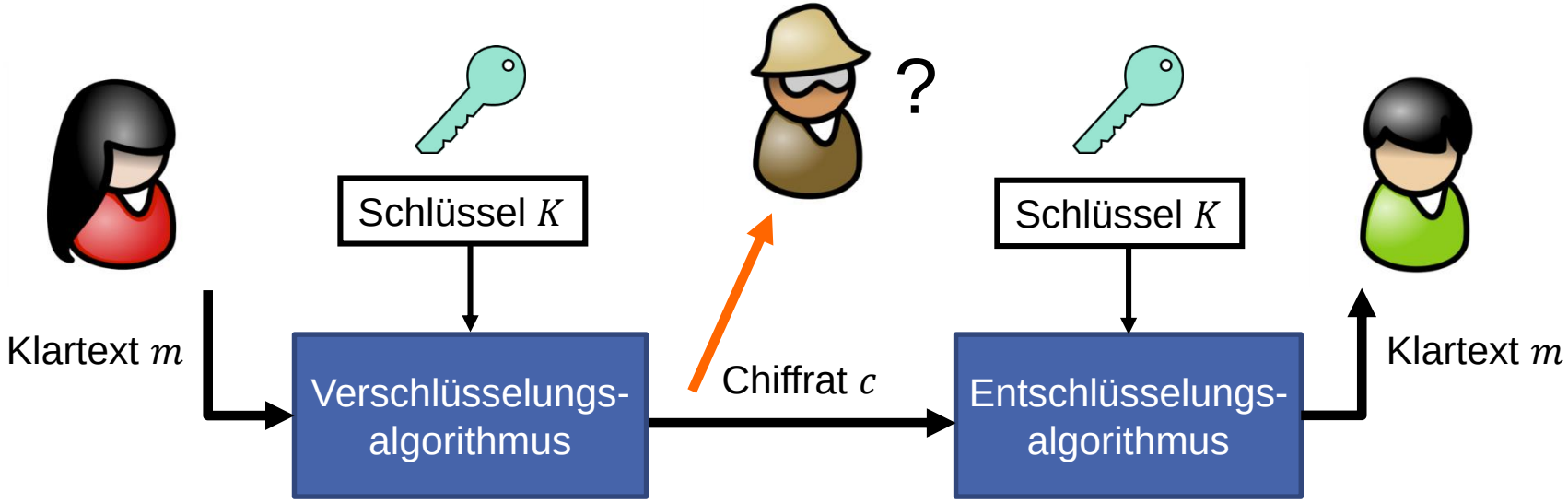
- Gemeinsames Geheimnis der Kommunikationspartner: Schlüssel K



- Verschlüsselungsalgorithmus Enc
  - Eingabe: gemeinsamer Schlüssel K, Klartext m
  - Ausgabe: Chiffrat $\mathbf{c}$ - verschlüsselter Klartext. Es gilt: $c = \mathrm{Enc}(K, m)$
- Entschlüsselungsalgorithmus Dec
  - Eingabe: gemeinsamer Schlüssel K, Chiffrat $\mathbf{c}$
  - Ausgabe: Klartext m. Es gilt: $m = Dec(K, c)$
    - $m = f^{-1}(K, f(K, m))$
    - Mit f^{-1} ist Umkehrfunktion von f

Symmetrische Verschlüsselung

- Schema

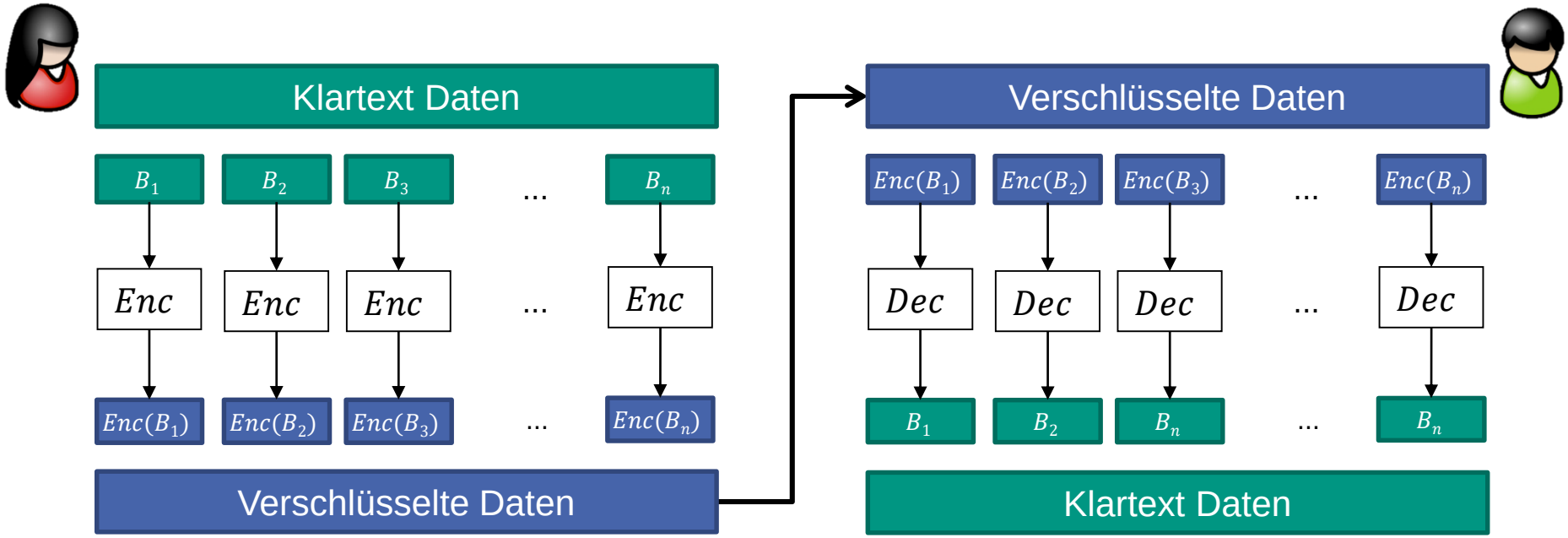


Symmetrische Verschlüsselung

- Zwei Klassen von Verfahren
  - Blockchiffren
    - Aufteilung der zu verschlüsselnden Daten in Blöcke der Länge k
  - Stromchiffren
    - Bit- oder Zeichenweises Verschlüsseln der Daten

Blockchiffren

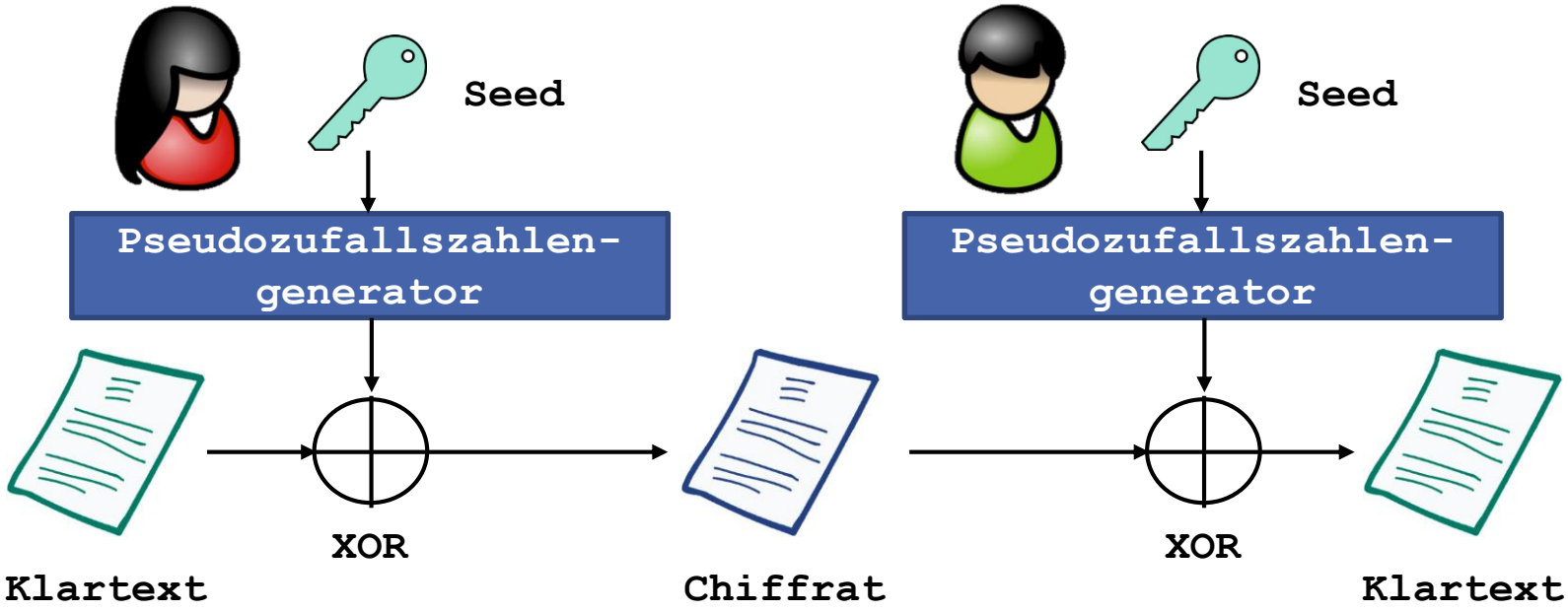
- Blockweises Verschlüsseln der Daten
  - Blöcke der verschlüsselten Daten haben gleiche Länge



- Beispiele
  - AES (Advanced Encryption Standard)
    - Blocklänge 128 Bit, Schlüssellänge 128, 192, 256 Bit
  - DES (Data Encryption Standard), 3DES
    - DES: Blocklänge 64 Bit, Schlüssellänge 56 Bit

Stromchiffren

- Bitweises (bzw. Zeichenweises) Verschlüsseln der Daten
  - Muss nicht warten bis ein Block von Daten bereit steht (z.B. Mobilfunk)
  - Gleicher Schlüssel als *Seed* für Zufallszahlengenerator bei Sender und Empfänger



- Pseudozufallszahlengenerator generiert mit gleichem *Seed* deterministische Folge

Asymmetrische Verschlüsselung

- Auch als Public-Key-Kryptografie bezeichnet

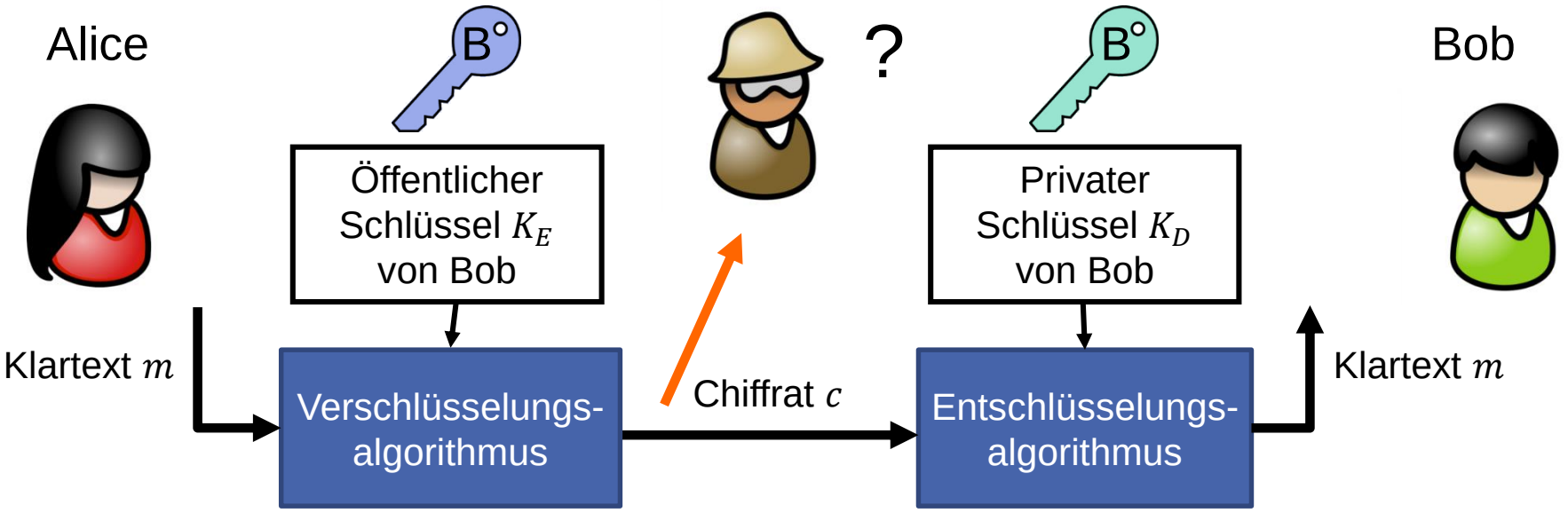
- *Kein* gemeinsames Geheimnis, zwei *unterschiedliche* Schlüssel
  - Öffentlicher Schlüssel (public key)
    - Ist allen bekannt (auch dem Angreifer)
  - Privater Schlüssel (private key)
    - Ist nur der kommunizierenden Instanz, z.B. Bob, bekannt

- Verschlüsselungsalgorithmus Enc
  - Eingabe: öffentlicher Schlüssel K_E des Ziels, Klartext m
  - Ausgabe: Chiffrat $\mathbf{c}$ - verschlüsselter Klartext. Es gilt: $c = Enc_{K_E}(m)$

- Entschlüsselungsalgorithmus Dec
  - Eingabe: privater Schlüssel K_D des Ziels, Chiffrat $\mathbf{c}$
  - Ausgabe: Klartext m. Es gilt: $m = Dec_{K_D}(c)$

Asymmetrische Verschlüsselung

- Schema



- Grundlage
  - Die Berechnung des privaten Schlüssels bei Kenntnis des öffentlichen Schlüssels, des Verschlüsselungsalgorithmus und ggf. des Chiffrats ist praktisch nicht möglich

Beispiel: RSA

Karlsruher Institut für Technologie

- Entworfen von Ron Rivest, Adi Shamir und Len Adleman

- Schlüsselerzeugung
  - Auswahl zweier großer Primzahlen p und q
  - Berechne

$$n = p * q$$

und

$$z = (p-1)(q-1)$$

  - Wähle eine Zahl $1 < e < z$, die teilerfremd zu z ist
  - Finde d, so dass $ed - 1$ durch z dividierbar ist, also

$$e * d \equiv 1 \, mod \, z$$

  - Öffentlicher Schlüssel: (n, e) 
  - Privater Schlüssel: (n, d) 

Beispiel: RSA

C I A

- Verschlüsselung
  - Bitmuster m mit $m < n$ soll verschickt werden
    - In Blöcke unterteilt mit Blockgrößen kleiner oder gleich $\mathrm{ld}(n) + 1$
    - Praktisch verwendete Blockgröße: 2^k bit mit $2^k < n \leq 2^{k+1}$
  - Alice berechnet $c = m^e \bmod n$
  - c wird an Bob gesendet

- Entschlüsselung
  - Bob berechnet $m = c^d \bmod n$

Vergleich der kryptografischen Verfahren

- Symmetrische Verfahren: AES
  - (+) Geringe Komplexität, höhere Effizienz (HW-Implementierung)
  - (−) Aufwändige Schlüsselverteilung
  - (−) Aufwändige Realisierung von digitalen Unterschriften
- Asymmetrische Verfahren: RSA
  - (+) Einfache Schlüsselverteilung
  - (+) Digitale Unterschriften einfach zu realisieren
  - (−) Höhere Komplexität, geringere Effizienz (trotz HW-Impl.)
- Empfehlung: Kombination (Hybrides Verfahren)
  - Beginn einer Interaktion mit asymmetrischem Verfahren
    - Austausch von symmetrischen Schlüsseln über gesicherten Kanal
  - Wechsel zu symmetrischem Verfahren

Pingo

http://pingo.upb.de/

9.4.2 Integritätssicherung

- Ziel
  - Daten sollen exakt so beim Empfänger eintreffen wie vom Absender versendet
  - → Manipulationen können nicht verhindert werden, sollen aber eindeutig erkannt werden … schwache Integrität

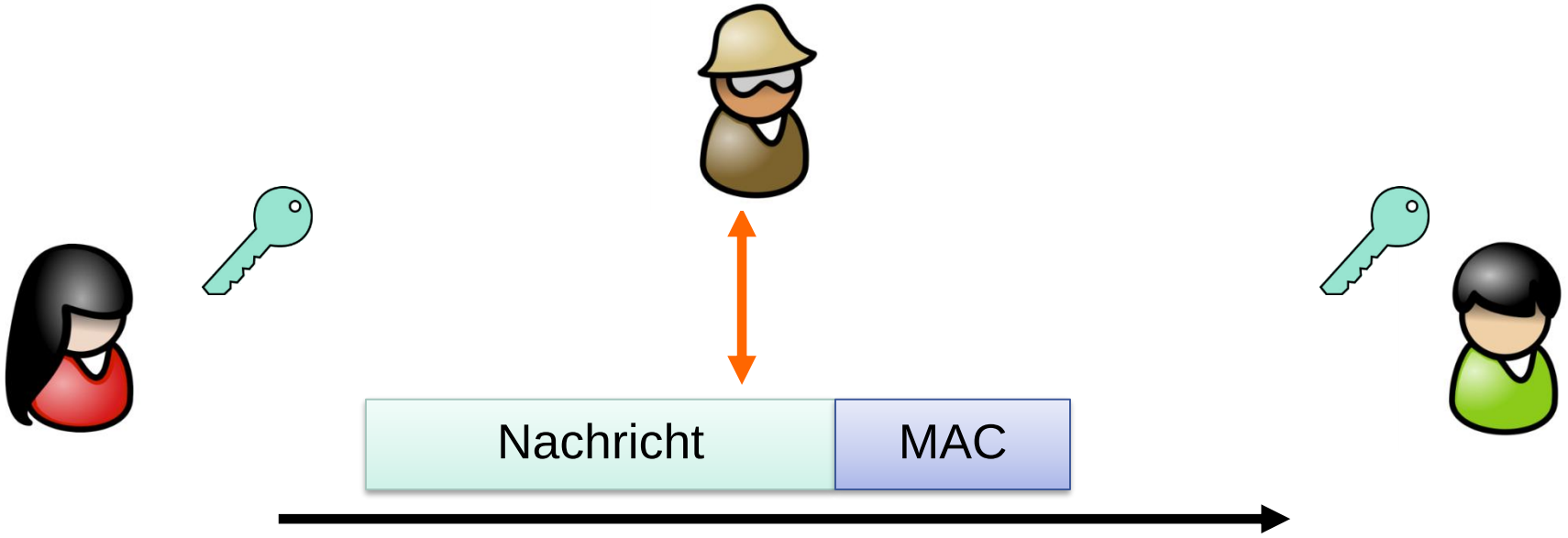
- Kryptografische Verfahren
  - Kryptografische Hashfunktion (Keyed Hash Function)
    - Hashfunktionen im Zusammenspiel mit gemeinsamem geheimem Schlüssel
  - Digitale Signaturen
    - Einsatz asymmetrischer Kryptografie

Kryptografische Hashfunktion

- Merkmal von Einwegfunktionen f
  - $f(a) = b$ ist leicht zu berechnen
  - $f^{-1}(b)$ darf nicht „effizient“ berechenbar sein … polynomialer Aufwand
  - Einwegfunktion/Urbildresistenz: zu gegebenem b ist es schwierig, ein a zu finden, so dass gilt $f(a) = b$
    - Anschauliches Beispiel: Telefonbuch
      - Nummer zu Name leicht
      - Name zu Nummer nicht leicht
  - Schwache Kollisionsresistenz
    - Für gegebenes a ist es schwierig, ein $\bar{a} \neq a$ zu finden, so dass gilt $f(a) = f(\bar{a})$
  - Starke Kollisionsresistenz
    - Es ist schwierig, zwei verschiedene Werte a und $\bar{a}$ aus der Urbildmenge A zu finden, so dass gilt $f(a) = f(\bar{a})$
- Kryptografische Hashfunktion
  - Bildet Nachrichten beliebiger Länge auf Hashwert fester Länge ab
    - Message Digest
  - Beispiele: MD5, SHA-1 (gelten als nicht mehr sicher), SHA-2, SHA-3 (noch in Standardisierung befindlich)

Message Authentication Code (MAC)

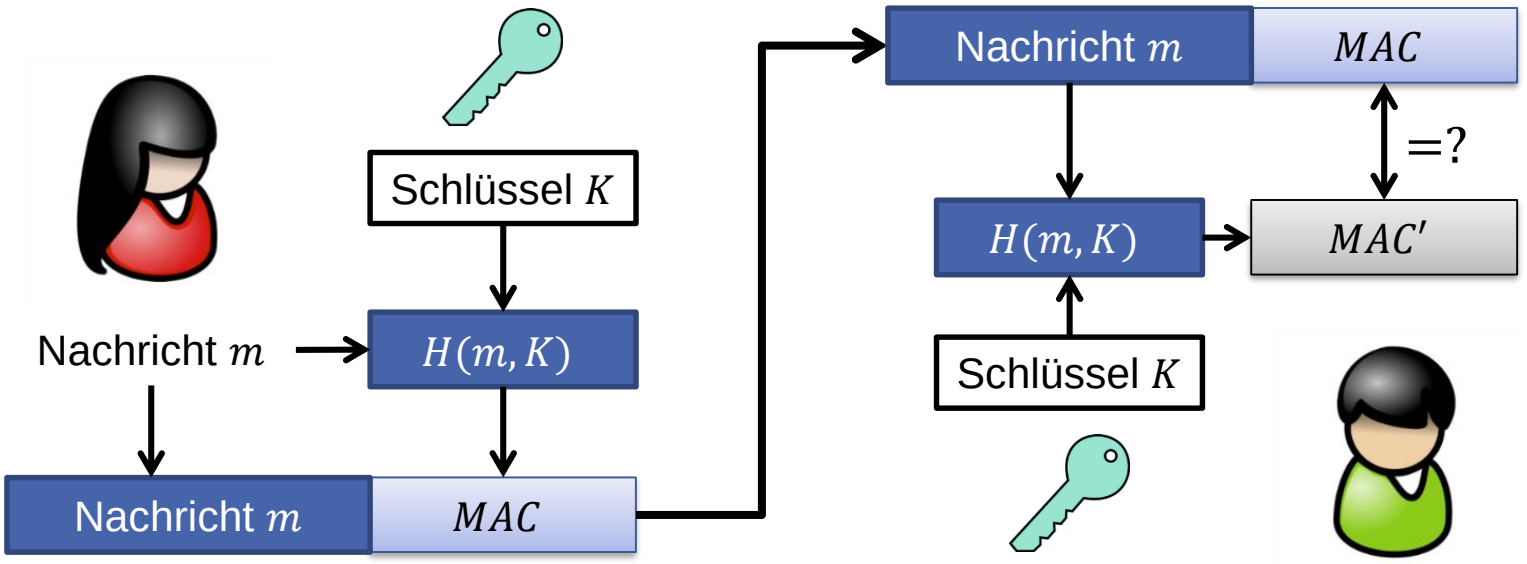
- Ziel
  - Empfänger erkennt Manipulationen an den gesendeten Daten
- Vorgehensweise
  - Alice erstellt Nachricht m und berechnet Hash $H(m)$
  - Alice hängt Hash an Nachricht an
- Voraussetzung
  - Alice und Bob besitzen gemeinsamen symmetrischen Schlüssel



Message Authentication Code

- Berechne Message Authentication Code auf Basis einer Hashfunktion H()
  - über zu sichernde Daten und
  - über geheimen Schlüssel
  - *Keyed-Hashing for Message Authentication*

→ Angreifer kann gültigen Hashwert nach Veränderung der Daten nicht neu berechnen

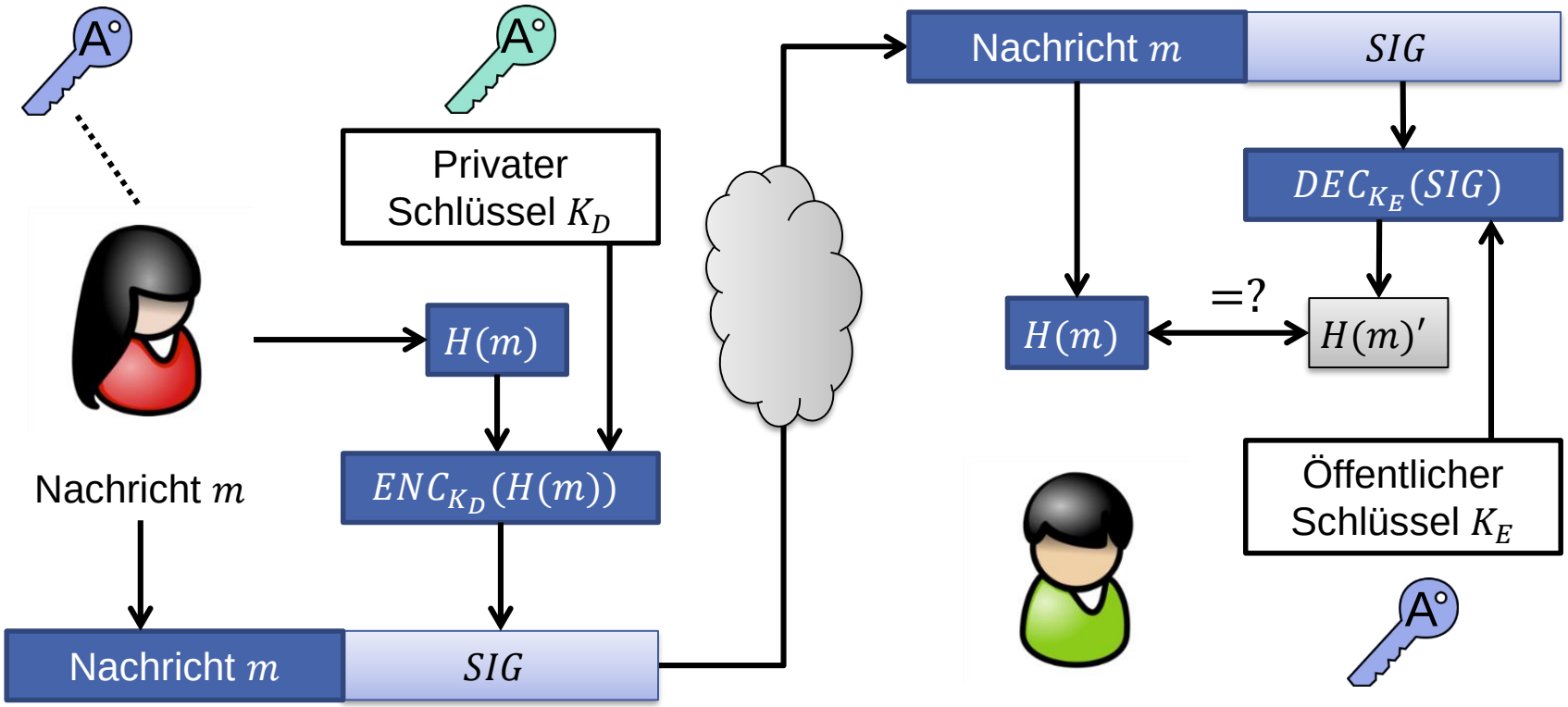


Digitale Signatur

- Eigenschaften
  - Vergleichbar denjenigen von „normalen“ Unterschriften
  - Nachweisbar, nicht fälschbar, kann nicht geleugnet werden
    - „Nichtabstreitbarkeit“
  - Basiert auf asymmetrischer Kryptografie
- Vorgehensweise
  - Dokument m wird mit privatem Schlüssel K_D verschlüsselt: $c = Enc_{K_D}(m)$
    - c dient als Signatur und wird mit m gesendet
  - Empfänger berechnet $m' = Dec_{K_E}(c)$ mit öffentlichem Schlüssel K_E
    - Signatur ist korrekt, falls gilt $m' = m$
  - Üblicherweise wird Signatur nicht über komplettes Dokument berechnet, sondern über wesentlich kürzeren Hash-Wert des Dokuments
- Alice kann damit verifizieren
  - Bob hat das Dokument unterschrieben
  - Kein anderer als Bob hat das Dokument unterschrieben
  - Bob hat das Dokument m und nicht das Dokument m' unterschrieben

Digitale Signatur

- Hier: Signatur über Hashwert des Dokuments



… also alles klar?

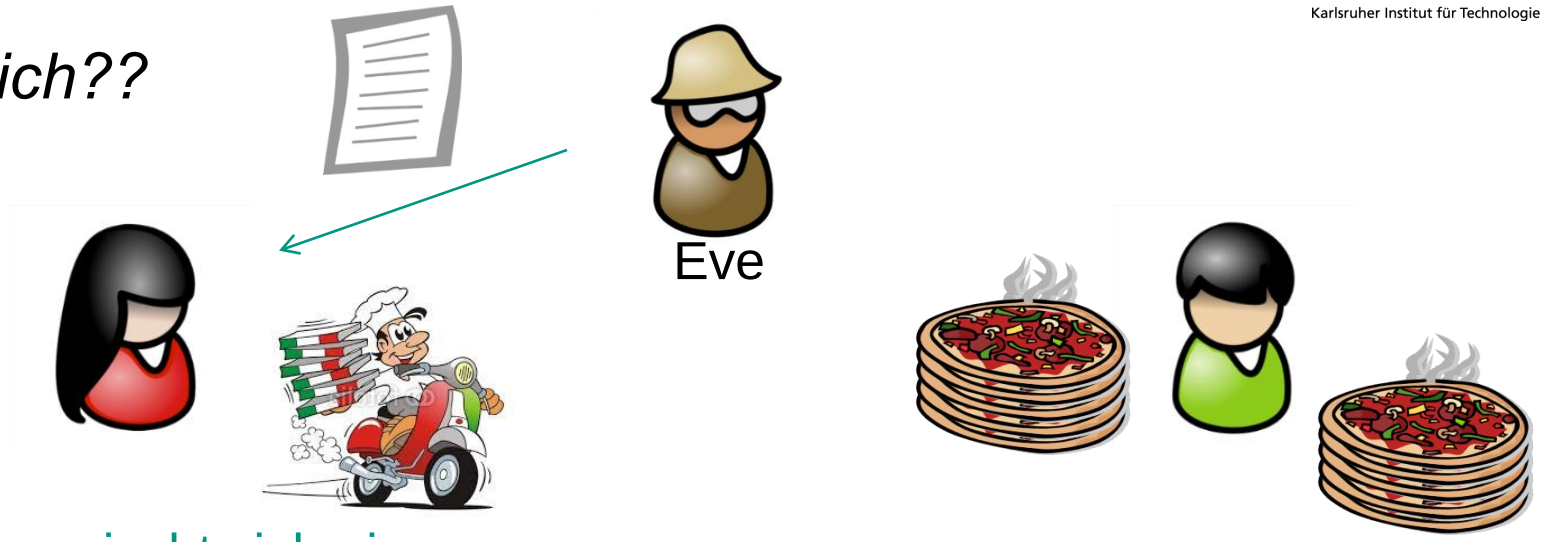
- Szenario: Eine ganz normale Pizza-Bestellung …

  - Alice
    - Arbeitet bei einem Pizzadienst der Bestellungen über das Internet erlaubt
  - Bob
    - Sendet Bestellung mit Heimadresse im Klartext
    - Nachricht enthält Bob‘s Signatur und seinen öffentlichen Schlüssel
  - Alice
    - *… ist die Bestellung wirklich von Bob ?*
    - Überprüft Signatur mit Bob´s, der Nachricht hinzugefügtem, öffentlichem Schlüssel

[KuRo10]

… also alles klar?

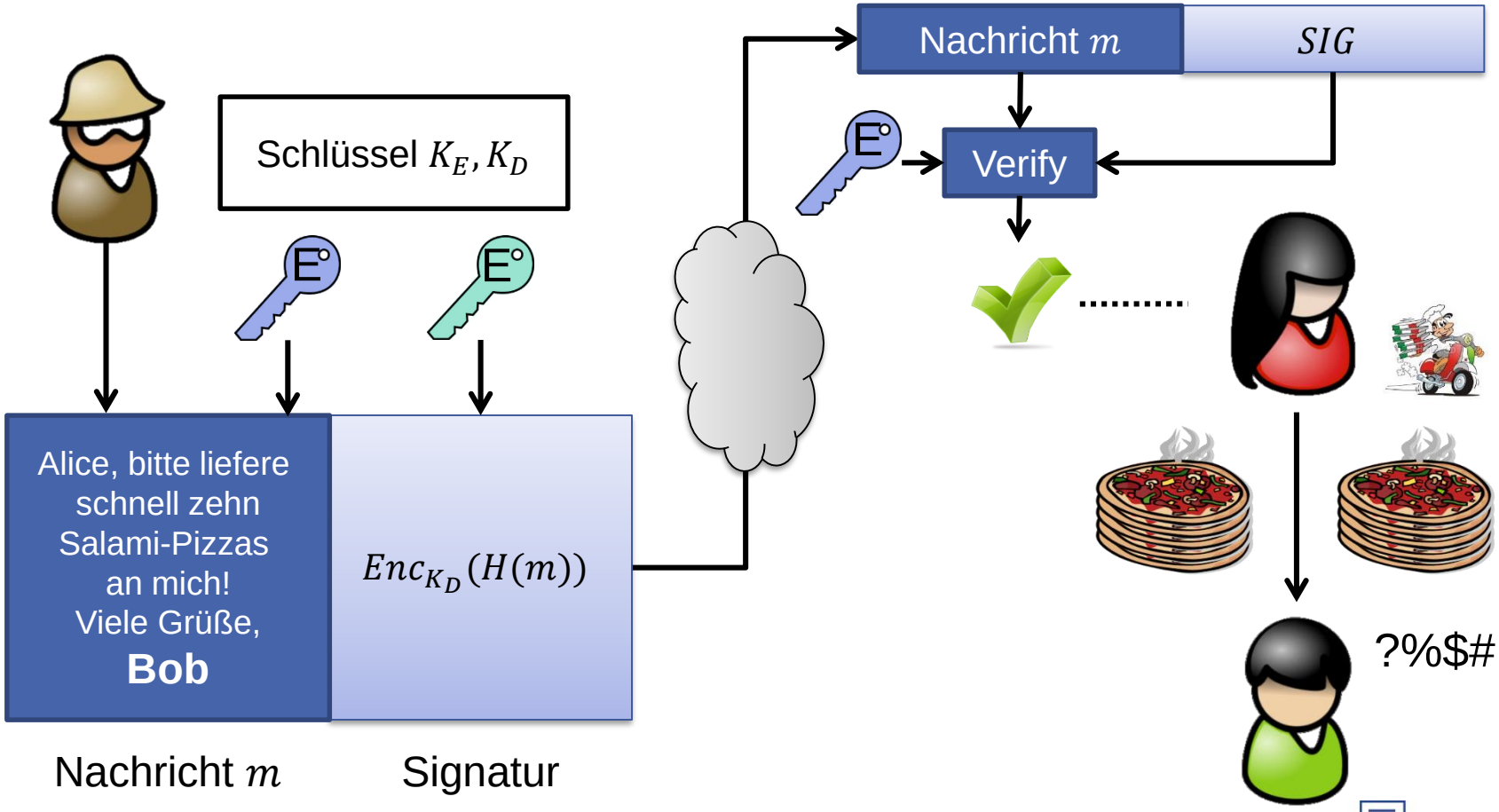
… wirklich??



- Eve mischt sich ein
  - Gibt vor Bob zu sein, schickt Bestellung an Alice
  - Nachricht enthält Eve´s öffentlichen Schlüssel und eine Signatur
    - Signatur mit Eve´s privatem Schlüssel erstellt
- Alice
  - Nutzt „Bob´s“ öffentlichen Schlüssel
    - … naja, also den von Eve
- Bob
  - … ist ziemlich überrascht über die Pizzalieferung

[KuRo10]

Eve´s Angriff schematisch dargestellt



[KuRo10]

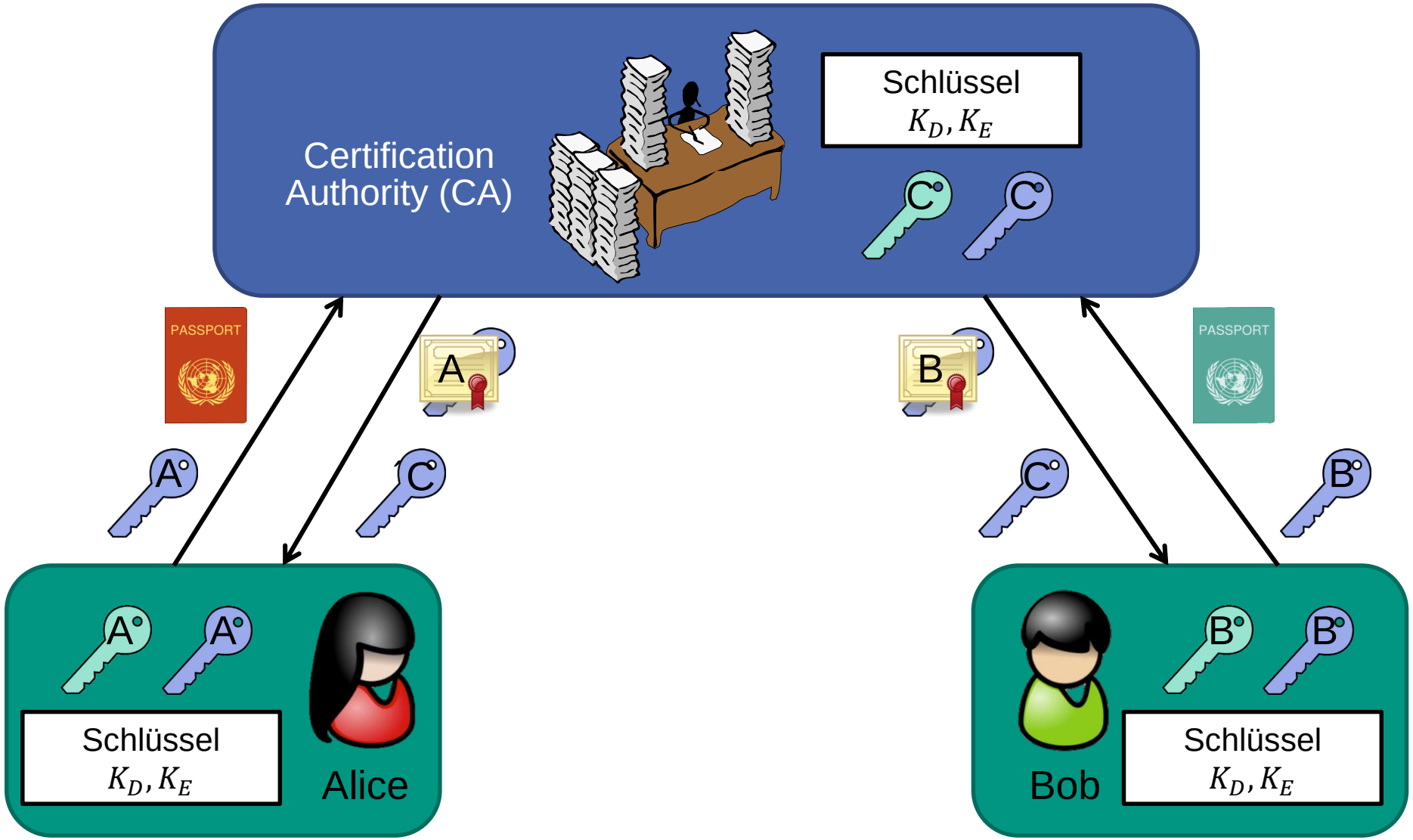
Grundlegendes Problem

- Signatur
  - Sichert Integrität der Nachricht
  - Authentizität wird nicht gesichert !
- Was fehlt?
  - Überprüfung ob der öffentliche Schlüssel demjenigen gehört, mit dem man kommunizieren möchte
  - Im Beispiel
    - Alice muss überprüfen können, ob der öffentliche Schlüssel zu Bob gehört
- Abhilfe
  - Digitale Zertifikate
    - Genauer ID-Zertifikate

Digitale Zertifikate

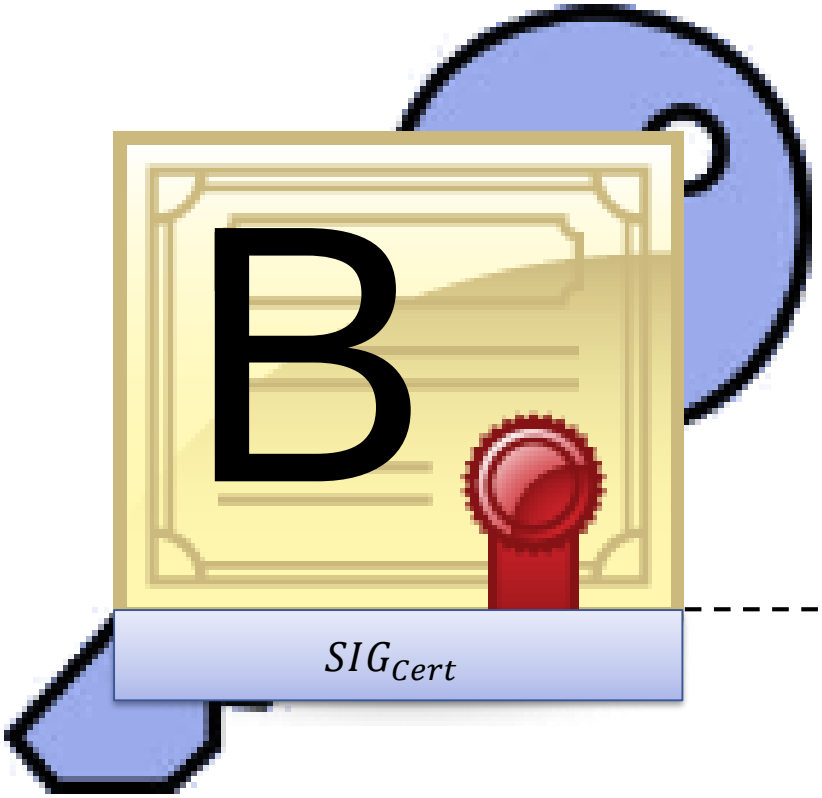
- Problemstellung
  - Authentifizierung eines Sachverhaltes, den man nicht selbst überprüfen kann
  - man verlässt sich auf vertrauenswürdige Dritte, die ihn schon kontrolliert haben
- Zertifikat ist digitales Dokument, in dem eine Instanz einen bestimmten Sachverhalt mittels digitaler Signatur bestätigt
  - erzeugt Vertrauen in den Sachverhalt
- Zertifikate werden von vertrauenswürdiger Instanz erstellt
  - Certification Authority (CA)
- ID-Zertifikate
  - öffentlicher Schlüssel → eindeutiger Name (*Identität*)
  - Authentifikation von öffentlichen Schlüsseln

ID-Zertifikate für Bob und Alice



Grober Aufbau eines ID-Zertifikates

- … hier für Bob



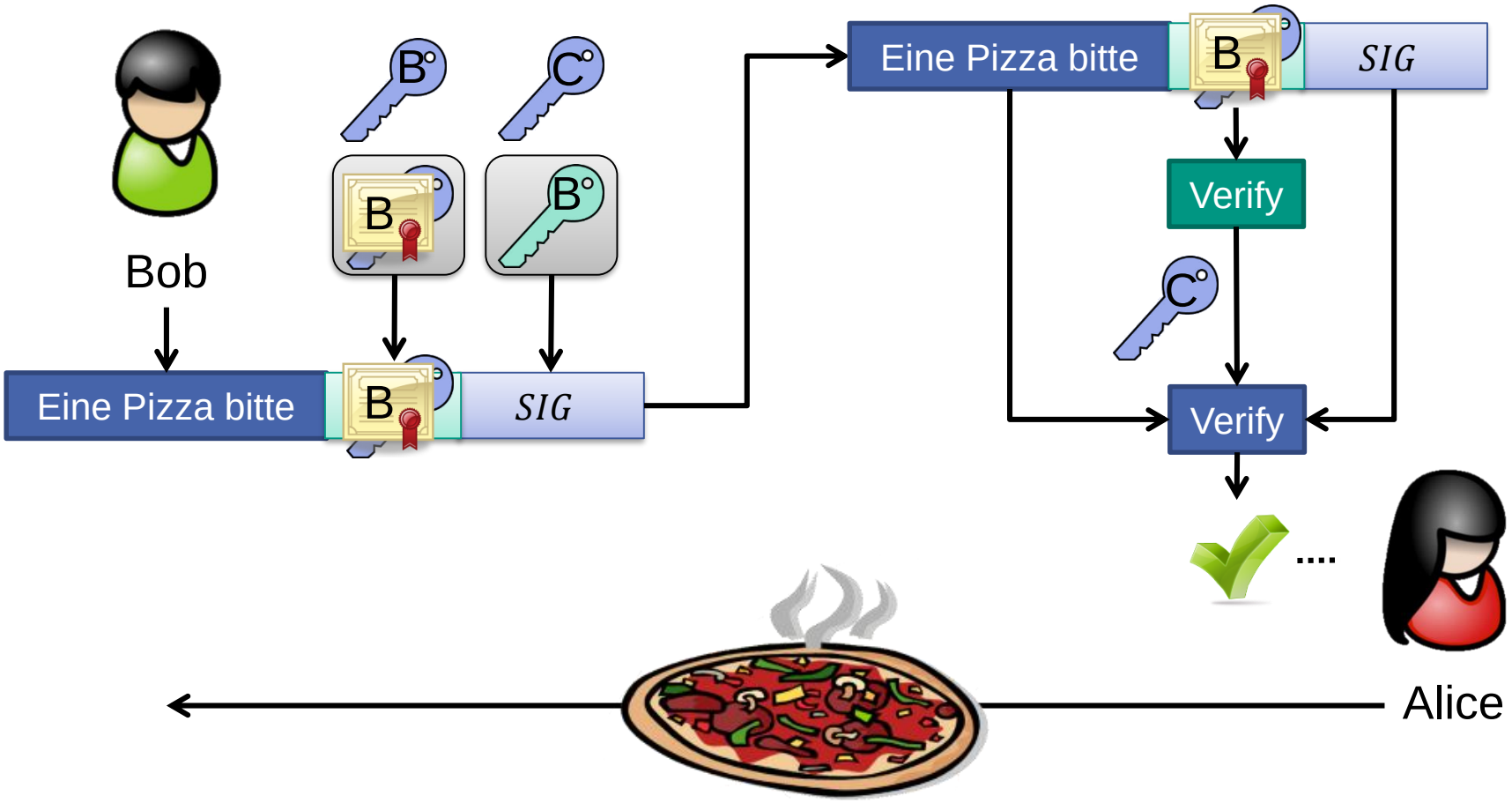
Eindeutiger Name der Identität

Öffentlicher Schlüssel der Identität K_E

Digitale Signatur der CA

Beispiel mit ID-Zertifikaten

- „Gelungene“ Pizza-Bestellung



Pingo

http://pingo.upb.de/

Netzsicherheit: Vielzahl an Themen und Begriffen

- Signatur
- Verschlüsselung
- Hashfunktion
- Message Authentication Code
- Schlüsselaustausch

Bausteine

- Integrität
- Authentizität
- Vertraulichkeit
- Autorisierung
- (Nicht)-Abstreitbarkeit

Schutzziele

- Abhören
- Einfügen
- Manipulieren
- Man-in-the-Middle
- Replay
- Denial-of-Service

Angriffe

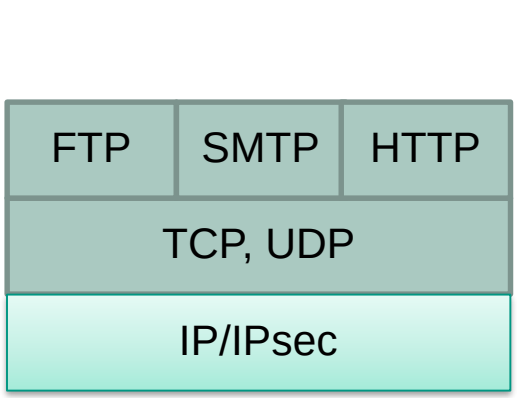
- TLS/SSL
- IPsec
- Kerberos

9.5 System/Protokoll

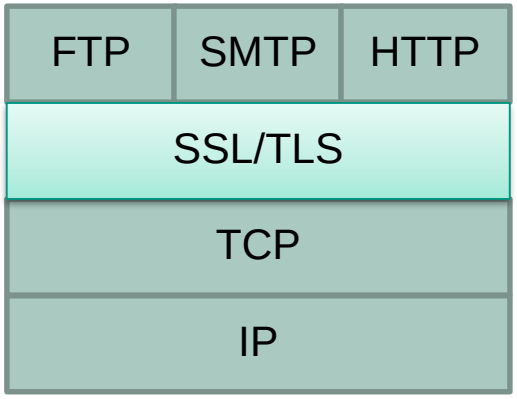
System/Protokoll verwendet Bausteine um Schutzziele zu realisieren und vor Angriffen zu schützen

9.5 Sicherheitsprotokolle

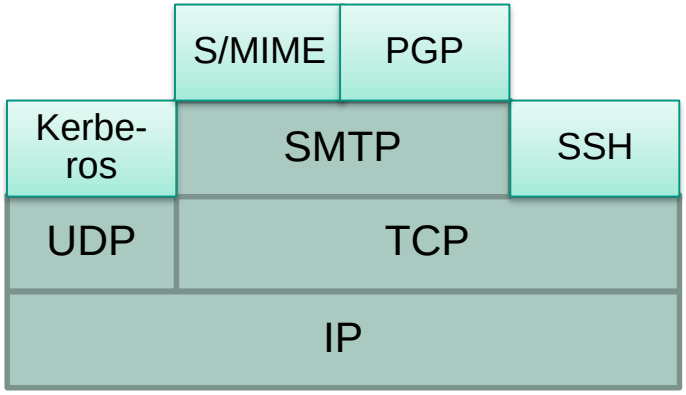
- Es gibt eine Vielzahl von Sicherheitsprotokollen, z.B.



Auf Vermittlungsschicht



Auf Transportschicht



Auf Anwendungsschicht

- Mehr dazu in weiterführenden Vorlesungen
  - Netzsicherheit: Architekturen und Protokolle
    - Aus der Forschungsgruppe von Prof. Dr. Zitterbart, im Sommersemester
  - IT-Sicherheitsmanagement für vernetzte Systeme
    - Aus der Forschungsgruppe von Prof. Dr. Hartenstein, im Wintersemester

Zusammenfassung

- Wichtige Schutzziele sind
  - **C**onfidentiality (Vertraulichkeit)
  - **I**ntegrity (Integrität)
  - **A**vailability (Verfügbarkeit)
- Verschlüsselung
  - Symmetrisch
  - Asymmetrisch
- Sicherung der Integrität und Authentizität
  - Message Authentication Code
  - Digitale Signatur
- Digitale Zertifikate

Literatur

[BBCH05] R. Bless, E.-O. Blaß, M. Conrad, H.-J. Hof, K. Kutzner, S. Mink, M. Schöller; Sichere Netzwerkkommunikation: Grundlagen, Protokolle und Architekturen; Springer-Verlag, 2005

[DiHe76] W. Diffie, M. E. Hellman; New Directions in Cryptography. In: IEEE Transactions on Information Theory. 22, Nr. 6, 1976, S. 644–654

[KaPS02] C. Kaufman, R. Perlman, M. Speciner; Network Security: Private Communication in a Public World; Prentice Hall, 2002

[KuRo10] J. Kurose, K. Ross; Computer Networking; Addison Wesley, 2010, 5th Edition

- Kapitel 8

[RFC2104] H. Krawczyk, M. Bellare, R. Canetti; HMAC: Keyed-Hashing for Message Authentication; RFC 2104, Internet Engineering Task Force, Februar 1997

[RFC4301] S. Kent, K. Seo; Security Architecture for the Internet Protocol; RFC 4301, Internet Engineering Task Force, Dezember 2005

[RFC5246] T. Dierks, E. Rescorla; The Transport Layer Security (TLS) Protocol Version 1.2; RFC 5246, Internet Engineering Task Force, August 2008

Literatur

[Schm09] K. Schmeh; Kryptografie: Verfahren, Protokolle, Infrastrukturen; Dpunkt-Verlag, 2009

[Schn04] B. Schneier; Secrets & Lies – IT-Sicherheit in einer vernetzten Welt; dpunkt.Verlag, 2004

- Interessante Diskussion zum Thema Sicherheit – keine Präsentation technischer Verfahren

[Sta10a] W. Stallings; Cryptography and Network Security: Principles and Practices; 5. Auflage, Prentice Hall, 2010

[Sta10b] W. Stallings; Network Security Essentials: Applications and Standards; 4. Auflage, Prentice Hall, 2010