Kapitel VI – Sicherheit in verteilten Systemen

1. **Definitionen**
2. Firewalls, Proxies & VPNs
3. Grundlagen der Kryptographie
4. Kerberos
5. Diffie-Hellman & Perfect Forward Secrecy
6. TLS
7. 802.11 Wireless LAN
8. Anonymisierung mit TOR

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Definitionen: Sicherheit (1)

Sicherheit ≠ Schutzmechanismus!

Sicherheit (-ziele)
CIA:

- **C**: Confidentiality
  Vertraulichkeit
- **I**: Integrity
  Integrität
- **A**: Availability
  Verfügbarkeit

Allgemeine Bedrohungen

- **Leakage**
  unerlaubtes Lesen
- **Tampering**
  unerlaubtes Schreiben
- **Vandalism**
  Störung des Betriebs

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Definitionen: Sicherheit (2)

Allgemeine Bedrohungen *Vernetzte* Bedrohungen

- **Leakage**
  unerlaubtes Lesen
- **Tampering**
  unerlaubtes Schreiben
- **Vandalism**
  Störung des Betriebs

- **Eavesdropping** (Lauschen)
- **Masquerading** (Senden/Empfangen unter falschen Namen)
- **Message Tampering** (Verfälschung bzw. Veränderung von Nachrichten)
- **Replaying** („Reflexion“, Abhören & Wiedergabe von Nachrichten)
- **Denial of Service** (Lahmlegen durch Überflutung)
- **Mobile Code** (falls böswillig)

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Kapitel VI – Sicherheit in verteilten Systemen

Maßnahmen: Firewalls (1)

Ein Firewall soll unerlaubte Pakete fern halten. Wie?

- **Paketfilter** – schließt z.B. manche Ports (von außen)

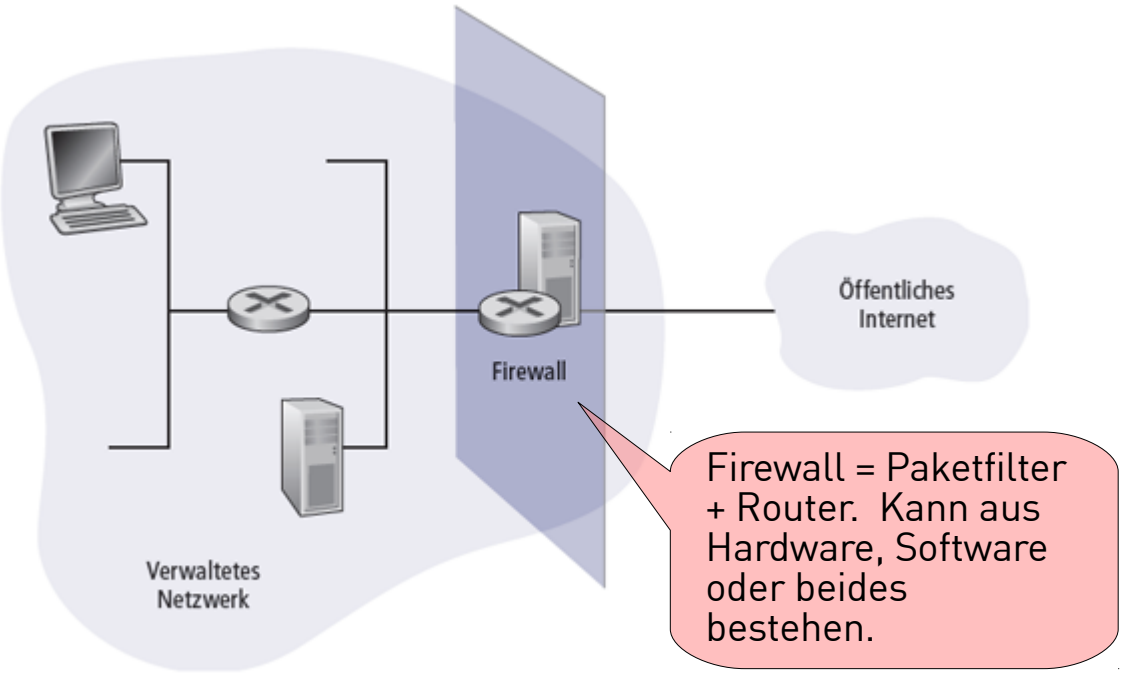


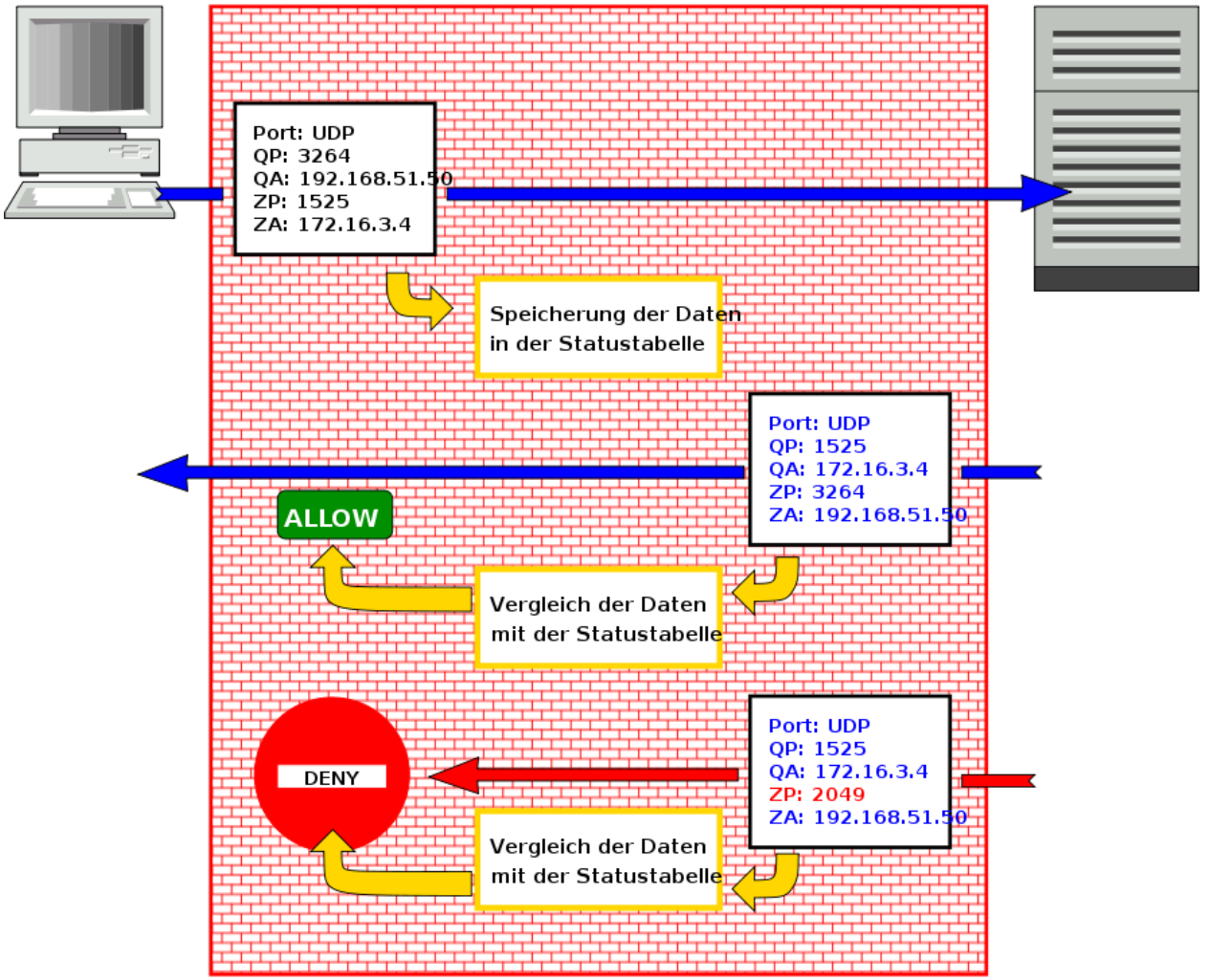
Abbildung 8.35: Firewall, platziert zwischen dem durch einen Administrator verwalteten Netzwerk und der Außenwelt

Quelle:Kurose & Ross, 4. Auglage, Kapitel 8

Maßnahmen: Firewalls (2)

Ein Firewall soll unerlaubte Pakete fern halten. Wie?

- **Paketfilter** – schließt z.B. manche Ports (von außen)
  - **Stateless** (einfach)
  - **Stateful** (komplizierte) – verwenden Wissen über Protokolle (wer sendet wann was an wen)



Quelle:http://de.wikipedia.org/w/index.php?title=Stateful_Packet_Inspection&oldid=53701082

Maßnahmen: Firewalls (3)

Ein Firewall soll unerlaubte Pakete fern halten. Wie?

- **Paketfilter** – schließt z.B. manche Ports (von außen)
- **Anwendungs-Gateway –** verwendet Wissen über Anwendungen z.B. Web-Application-Gateway (welche Parameter nehmen welche Web-Seiten?)

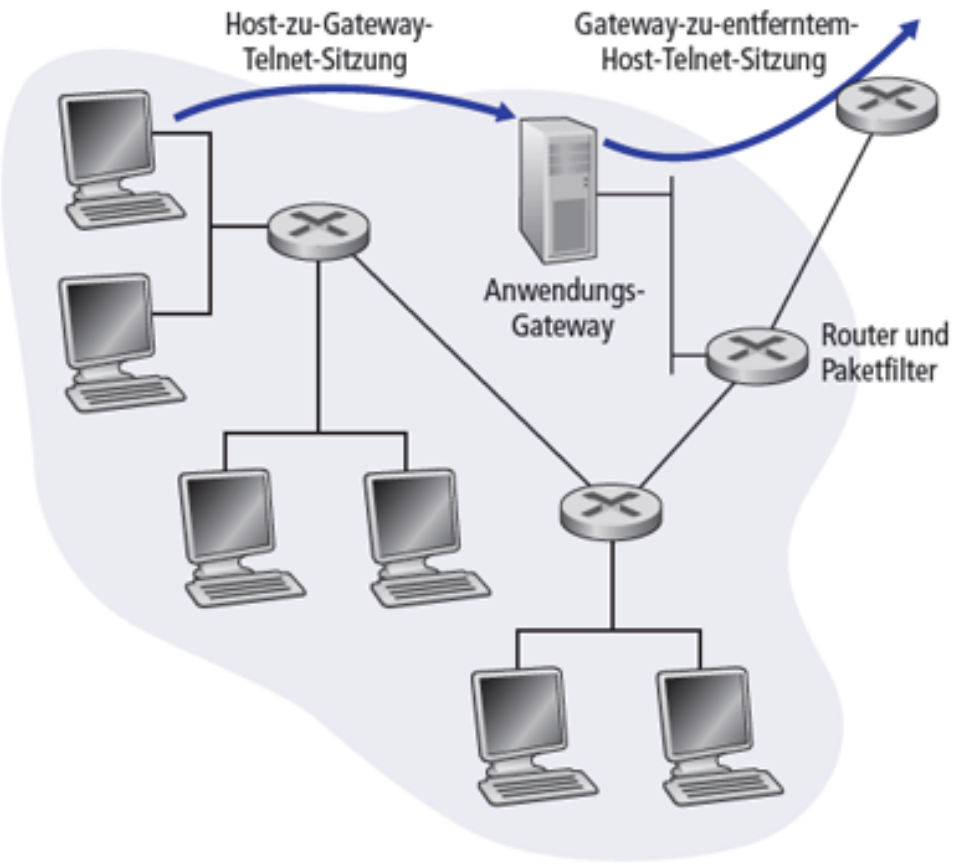


Abbildung 8.36: Firewall, bestehend aus einem Anwendungs-Gateway und einem Filter

Quelle:Kurose & Ross, 4. Auglage, Kapitel 8

Maßnahmen: Firewalls (3)

Ein Firewall soll unerlaubte Pakete fern halten. Wie?

- **Paketfilter** – schließt z.B. manche Ports (von außen)
- **Anwendungs-Gateway** – verwendet Wissen über Anwendungen
- **IDS/IPS** – Intrusion Detection bzw. Prevention System.

  IDS versucht Einbrecher an Hand von bekannten Muster zu erkennen.

  IPS versucht sie auch zu blockieren.

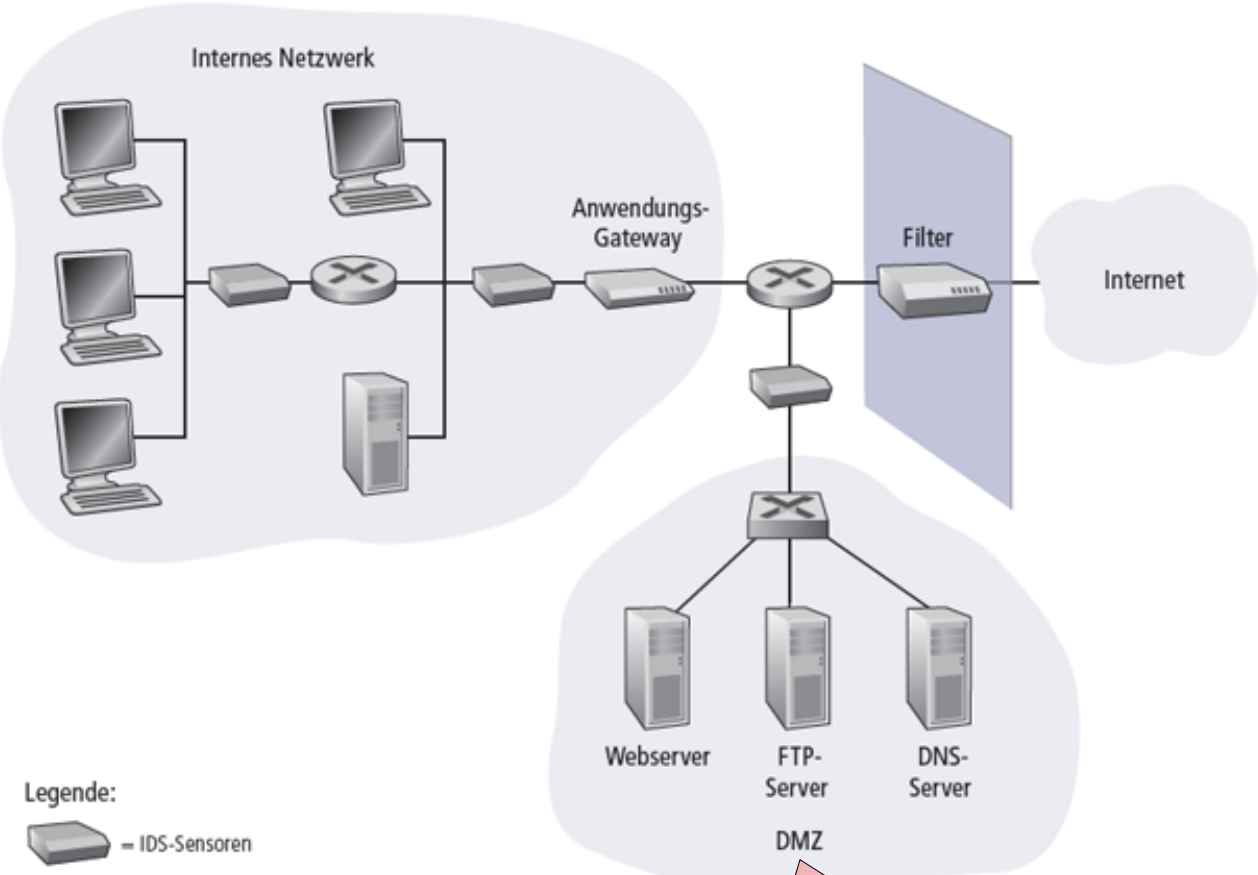


Abbildung 8.37: Eine Organisation, die Filter, Anwendungs-Gateway und IDS-Sensoren einsetzt

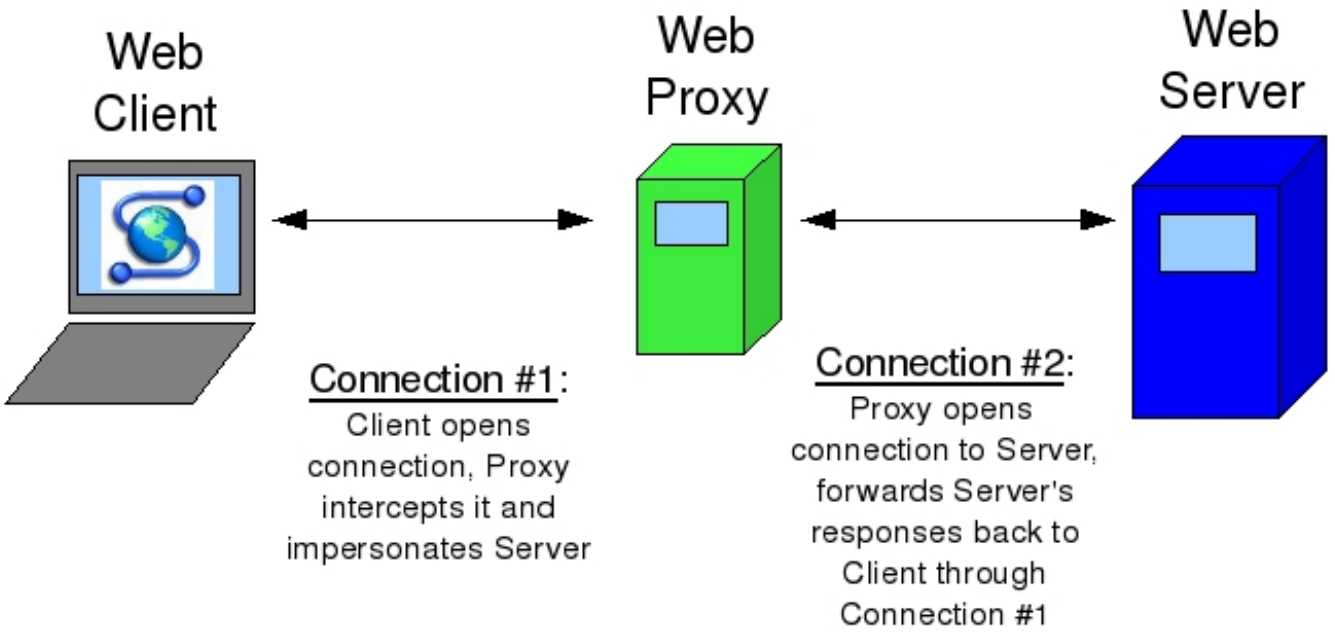
DMZ = Demilitarized Zone = Niemandsland = Server, die von außen erreichbar sein muss, mit eingeschränkte Zugang „nach innen“

Quelle:Kurose & Ross, 4. Auglage, Kapitel 8

Maßnahmen: Proxies

Proxy Server

- Leiten internen Netzwerkverkehr an externe Server weiter
- Z.B. HTTP Proxy für Webtraffic
- Vergleich Anwendung Gateway (Firewall)!?!



Maßnahmen: Proxies

- Frage: Kann ein Proxy auch Kommunikation mit verschlüsselten Webseiten bearbeiten?

  Der Proxy musste ja den Inhalt der Nachrichten lesen... oder?

- Antwort: Es geht (ohne den Inhalt zu lesen) – mit einem Trick.

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Maßnahmen: Proxies

Spezielle HTTP Nachricht: HTTP Connect (RFC 2817)
Beispiel: `CONNECT www.google.de:443 HTTP/1.1`

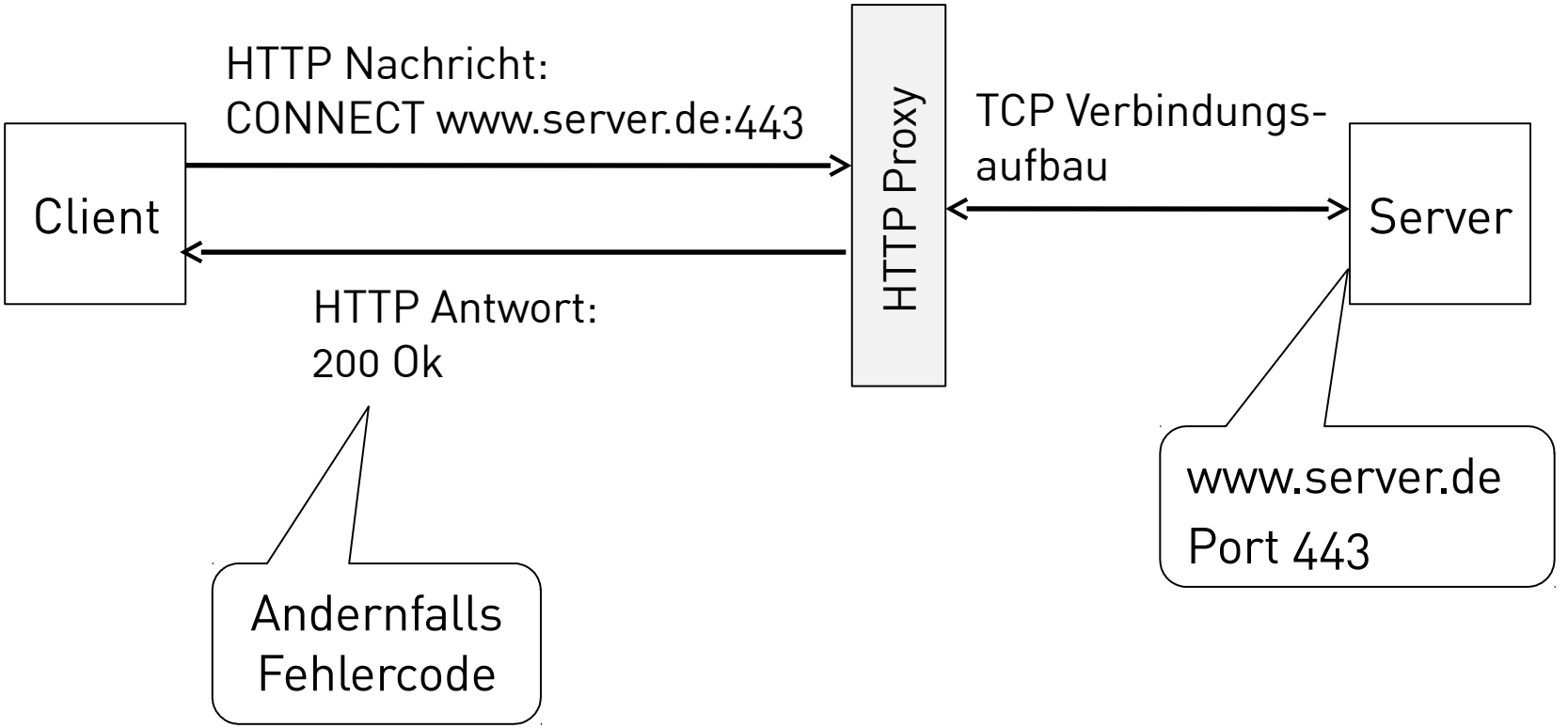
1. Client baut TCP Verbindung zum Proxy auf
2. Client sendet HTTP CONNECT mit IP und Port des Zielservers an den Proxy
3. Proxy öffnet normale TCP Verbindung zum Server und leitet danach einfach die gesendeten Nachrichten vom Client an den Server weiter (und umgekehrt), ohne in die Pakete zu schauen

Damit ist es möglich, beliebige TCP-basierte Protokolle durch einen HTTP Proxy zu *tunneln*, z.B. auch RPC Aufrufe (TCP)

Quelle:Kurose & Ross, 4. Auglage, Kapitel 8

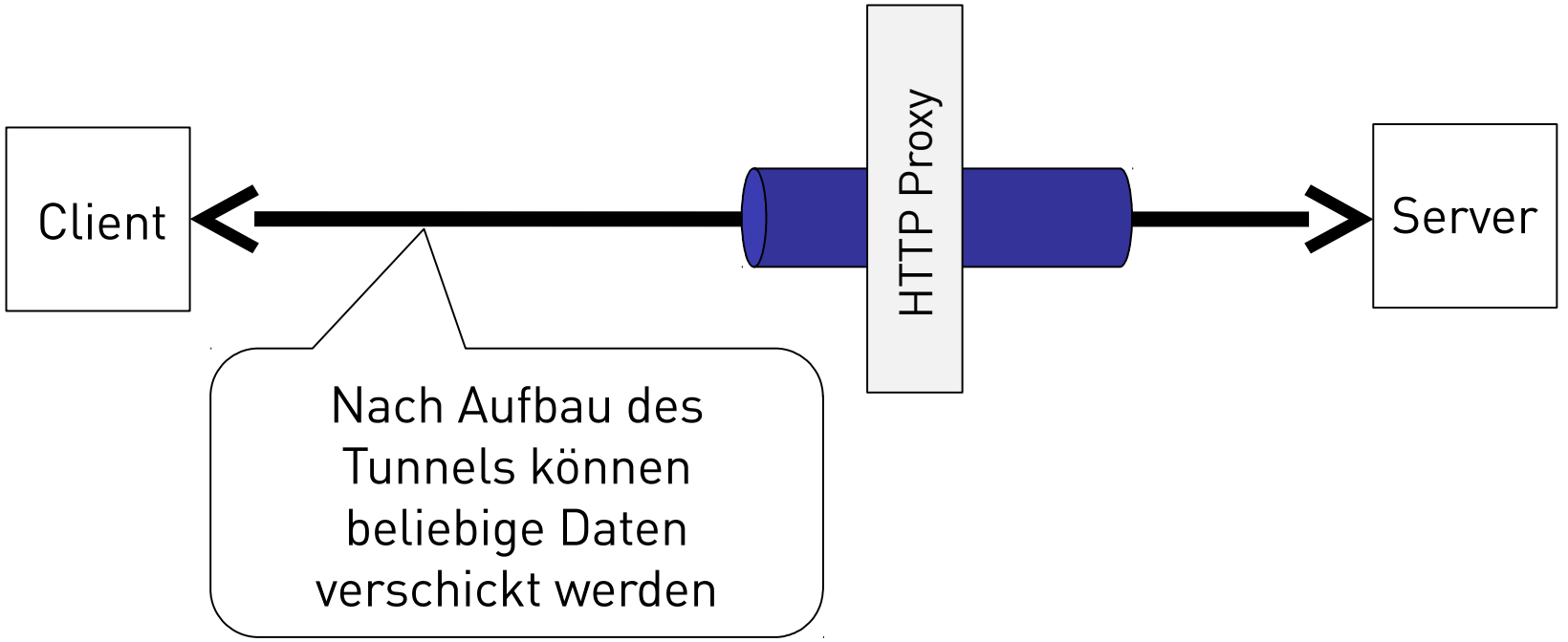
Maßnahmen: Proxies

- Client nutzt TCP Verbindung
- Geschickt wird zuerst eine HTTP Nachricht



Maßnahmen: Proxies

- TCP Verbindung zum Server kann danach für beliebige Daten genutzt werden

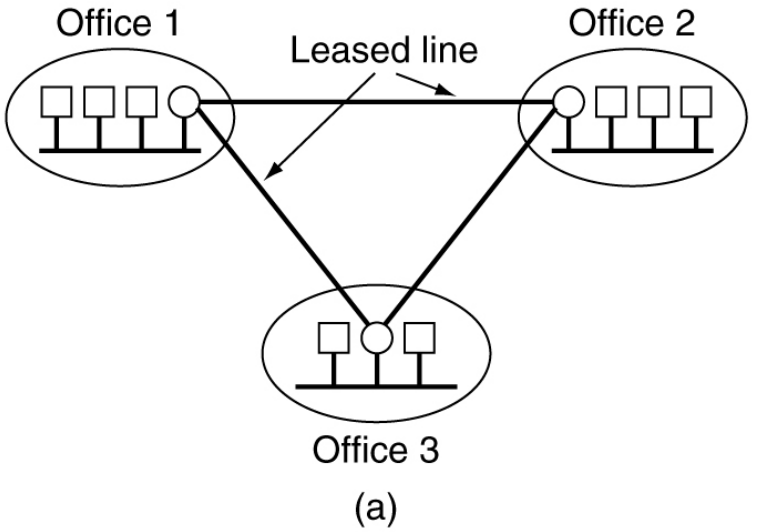


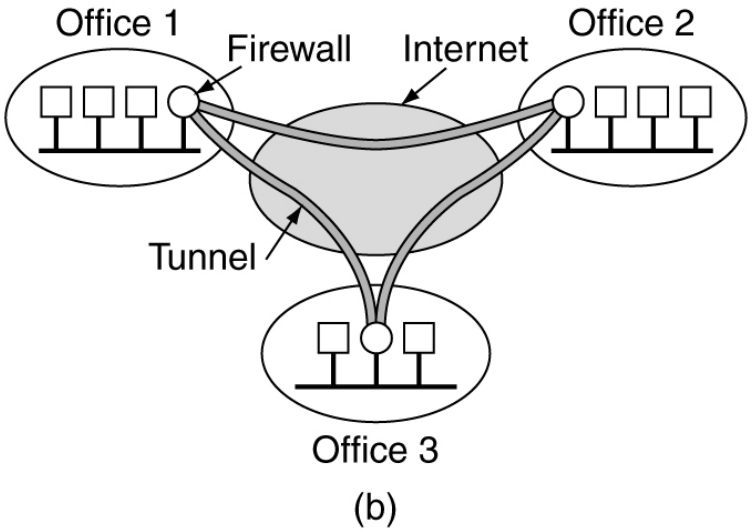
- Frage: Wie beendet der Client den Tunnel?

Quelle:Kurose & Ross, 4. Auglage, Kapitel 8

Maßnahmen: VPNs

Ein VPN (*Virtual Private Network*) ist eine Hardware- bzw. Software-Lösung, die Netzverkehr **verschlüsselt**, sodass offene Netze wie private Netze verwendet werden können.





(a) Private Network

(b) Virtual Private Network

Quelle:Tanenbaum, Computer Networks, 4. Edition bzw. Companion Website

Kapitel VI – Sicherheit in verteilten Systemen

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Begriffe der Kryptographie (1)

- Verschlüsselung: **Klartext → Chiffretext**
- Entschlüsselung: **Chiffretext → Klartext**
- **One-Time Pad:**
  Die *beweisbar* einzige sichere Methode für Symmetrische Kryptographie (s.u.):

  Zufälliger (sehr, sehr wichtig!) Schlüssel, genauso lang wie die zu verschlüsselnde Botschaft

  Beispiel: Nachricht XOR One-Time-Pad

Nachricht	1 1 0 1 0 1
Schlüssel	0 1 0 0 1 0
Verschlüsselte Nachricht	1 0 0 1 1 1
Schlüssel	0 1 0 0 1 0
Entschlüsselte Nachricht	1 1 0 1 0 1

Wichtig: Schlüssel muss geheim bleiben, und darf (daher) höchstens ein mal verwendet werden.

Der Beweis stammt von Claude Shannon, der Vater der Informationstheorie!

Problem: Kommunikation bzw. Schutz der Schlüssel.
Trotzdem: Wurde während des Kalten Krieges tatsächlich verwendet!

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Begriffe der Kryptographie (1,5)

- **Symmetrische Kryptographie (Secret-Key Kryptographie)** :
  Verschlüsselung und Entschlüsselung werden mit demselben Schlüssel gemacht, der also geheim gehalten werden muss (aber trotzdem kommuniziert werden).

- **Public-Key Kryptographie:**
  Ein Schlüssel für Verschlüsselung, ein anderer für Entschlüsselung. Jeder Teilnehmer hat einen *Public* und einen *Private Key.*

  Ein Key wird verwendet für Verschlüsselung, der andere Key wird für Entschlüsselung verwendet.

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Begriffe der Kryptographie (2)

Alice	1. Teilnehmerin
Bob	2. Teilnehmer
Carol	3. Teilnehmer
Dave	4. Teilnehmer
Eve	Eavesdropper
Mallory	Malicious attacker
Sara	A Server

K_A	Alice's Secret Key
K_B	Bob's Secret Key
K_{AB}	Key shared by Alice & Bob
K_{Apriv}	Alice's private key (known only to Alice)
$\{M\}_K$	Message M encrypted with Key K
$[M]_K$	Message M signed with Key K

Alternativ: Tanenbaum *et. al.* verwenden $K(M)$ für M, mit K verschlüsselt.

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Grundlagen der Kryptographie

Definition: Eine Funktion

$$y = f(x)$$

ist ein Einwegfunktion wenn:

- *y* leicht (effizient) zu berechnen ist – gegeben *x*.
- *x* sehr schwer ("computationally infeasible") zu berechnen ist – gegeben *y*.

Beispiel: Primfaktorzerlegung (allerdings nicht bewiesen!).

Sonderfall: Hash-Funktion
Wenn es mehr als ein x gibt, sodass

$$y = f(x_1) = f(x_2) = f(x_3) \ldots$$

I.d.R. ist Größe(y) << Größe(x)

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Grundlagen der Kryptographie

Definition: Blockverschlüsselung (Block cipher):

Wenn D eine Datenmenge ist, sodass $|D| >> |x|$, dann können wir keine Einweg-Funktion $y = f(x, k)$ verwenden, um D zu verschlüsseln (wo k ein Schlüssel ist).

Eine **Blockverschlüsselung** von D teilt D in *Blöcke* der Länge $|x|$, und verschlüsselt jede Block.
Wenn $D = d_0, d_1, d_2 \ldots$ ist
$\{D\} = \{d_0\}, \{d_1\}, \{d_2\} \ldots$

Welche Probleme haben alle Blockverschlüsselungen?

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Beispiel Algorithmus 1: TEA

TEA = Tiny Encyrption Algorithm

Symmetrishe Blockverschlüsselung. Nicht ganz sicher, aber sehr kleiner Verschlüsselung-Algorithmus (s. Wikipedia für Verbesserungen).

K: Key, Länge = 4, text hat Länge = 2.

```
void encrypt(unsigned long k[], unsigned long text[]) {
    unsigned long y = text[0], z = text[1];
    unsigned long delta = 0x9e3779b9, sum = 0; int n;
    for (n= 0; n < 32; n++) {
        sum += delta;
        y += ((z << 4) + k[0]) ^ (z+sum) ^ ((z >> 5) + k[1]);
        z += ((y << 4) + k[2]) ^ (y+sum) ^ ((y >> 5) + k[3]);
    }
    text[0] = y;  text[1] = z;
}
```

Nein, kein Zufallszahl!

Ziele (nach Schannon): *Diffusion & Confusion (bzw. Permutation & Substitution).*

Quelle:Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn 4, © Pearson Education 2005

Grundlagen der Kryptographie

Beispiel: **AES („Advanced Encryption Standard“)**

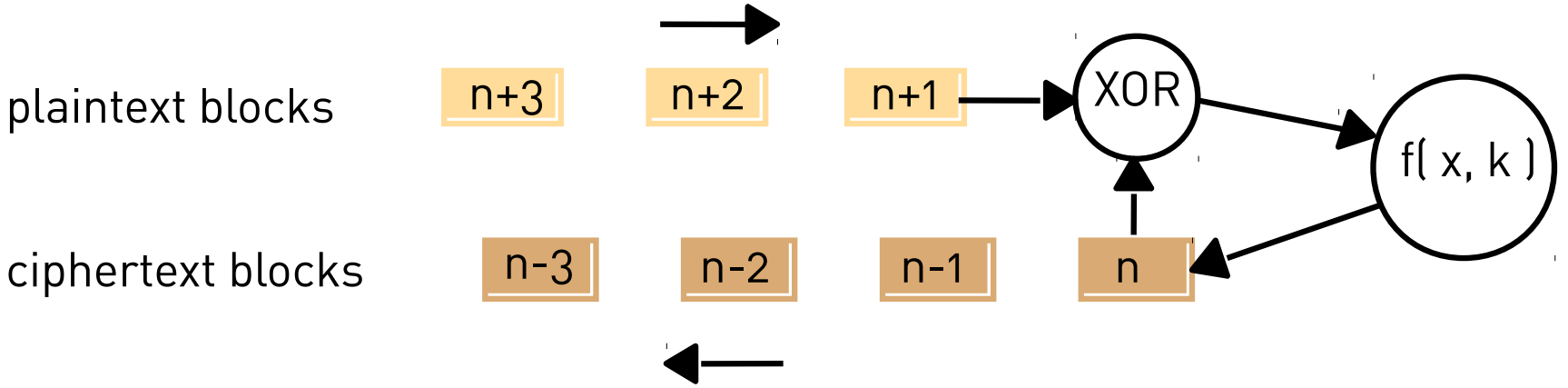
- Symmetrische Blockverschlüsselung
- Nachfolger von DES, 3-DES (beide gelten als sehr unsicher)
- Schritte (u.a.): Schlüsselerzeugung, rotieren, durchmischen... Vgl. https://de.wikipedia.org/wiki/Advanced_Encryption_Standard

- Empfohlen von der NSA: Suite B
  - http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Definition: Cipher Block Chaining

Wir können Blockverschlüsselung verbessern, wenn wir für jeden Block die letzte Blockverschlüsselung auch einbeziehen:



Cipher Block Chaining

Zweck

- Muster im Klartext werden verhindert
- Identische Klartextblöcke ergeben unterschiedliche Ciphertext Blöcke
- Sicherheit gegen weitere Angriffe...

Frage: Welche Auswirkung hat ein Übertragungsfehler in einem Block? Wie viele entschlüsselte Blöcke werden damit wertlos?

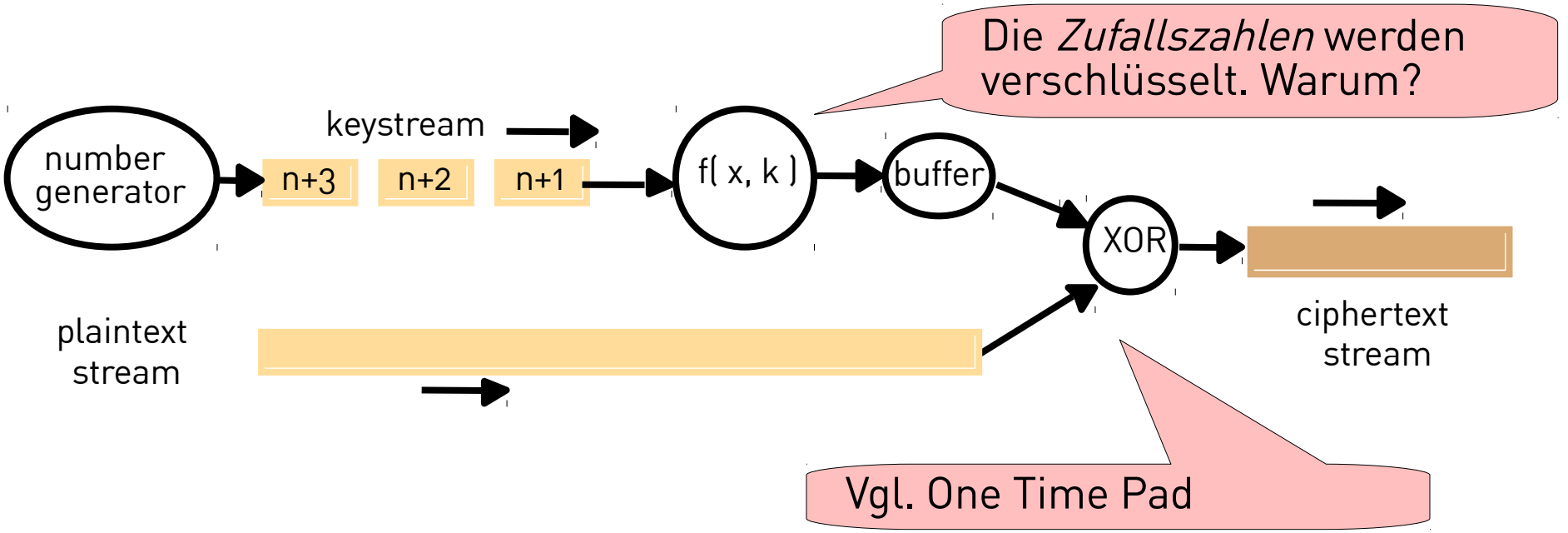
Frage: Wie verschlüsselt man den ersten Block?

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Definition: Stromverschlüsselung (Stream cipher):

Cipher Block Chaining läßt sich nicht in *Echtzeit* (zB für VoIP) verwenden.

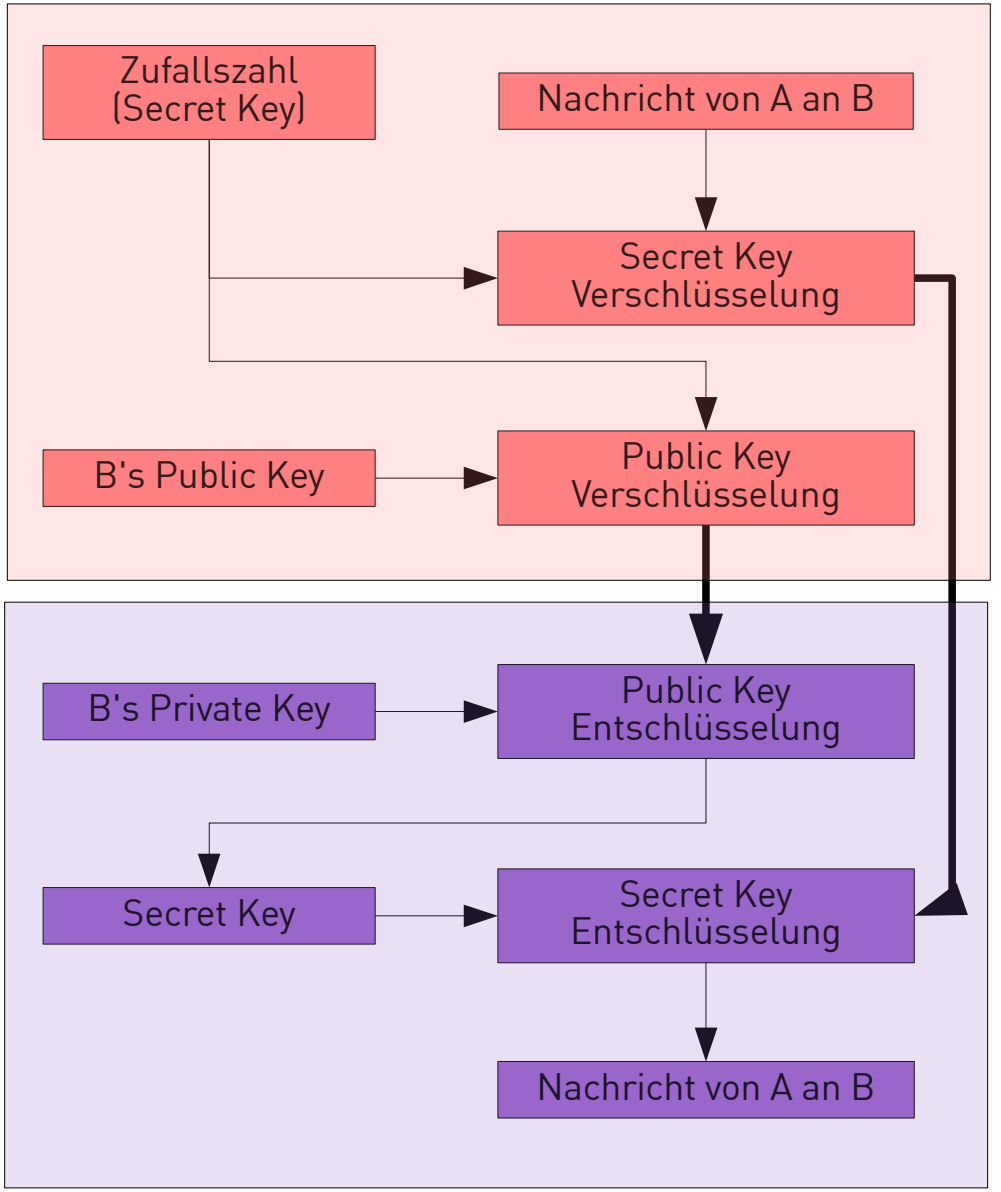
Eine **Stromverschlüsselung** von *D* verwendet zusätzlich einen **Schlüsselstrom**.



Public-Key: Verschlüsselung

Verschüsselung – für *Confidentiality*:

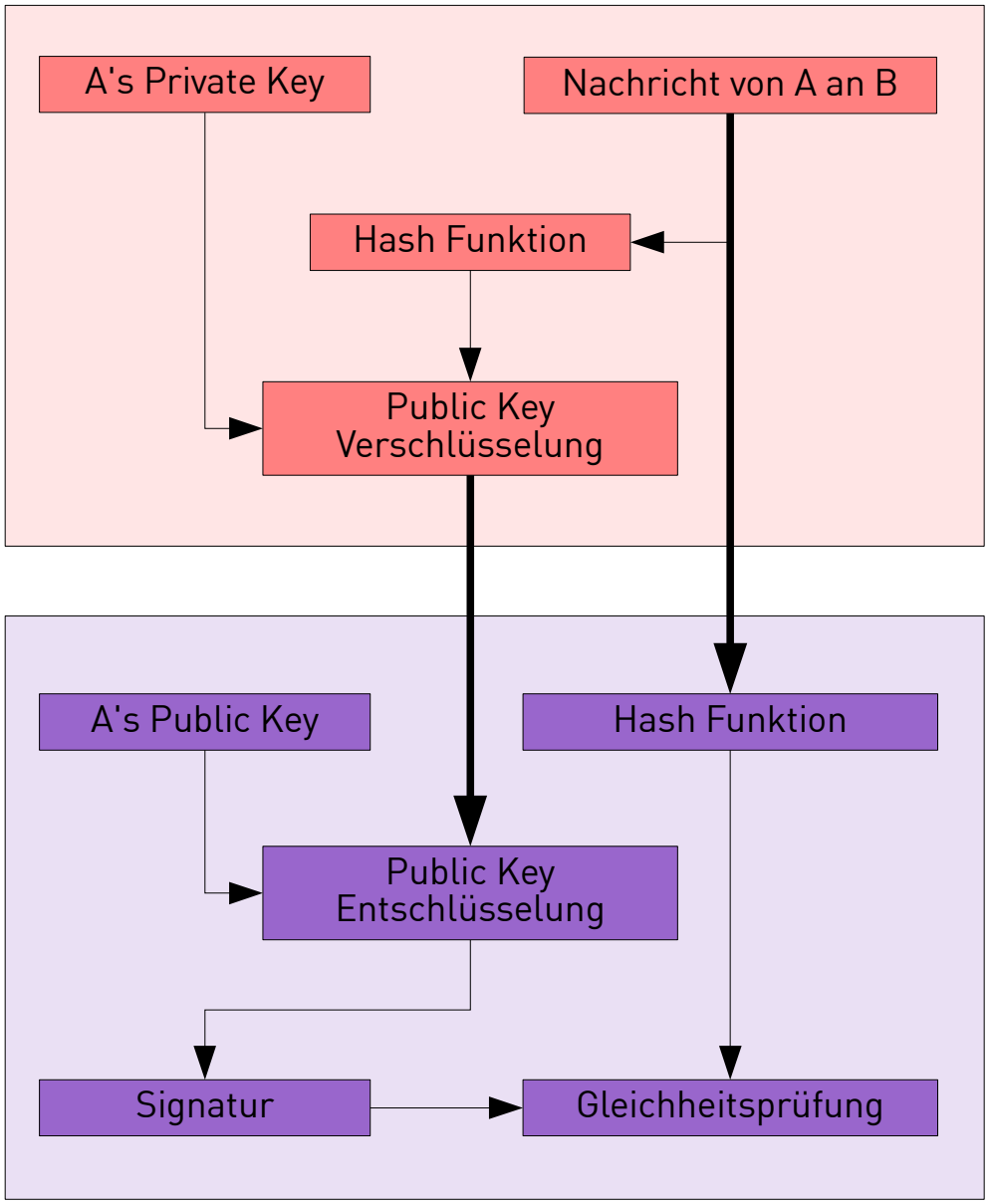
- **Alice will eine Email an Bob schicken, die nur Bob lesen kann.**
- Sie verwendet Bob's *Public Key* für Verschlüsselung.
- Nur Bob kann die Nachricht entschlüsseln.
- Eigentlich wird eine zufällig erzeugte *Secret Key* meistens kommuniziert (mit dem die Nachricht ver- bzw. Entschlüsselt wird).
- Allerdings: Bob weiß nicht sicher, ob die Email wirklich von Alice stammt bzw. nicht manipuliert wurde.



Public-Key: Signaturen

Digital Signatures – für *Integrity*:

- **Alice will eine Email an Bob schicken, sodass Bob weißt, dass die Nachricht von Alice stammt (und nicht manipuliert wurde).**
- Alice schreibt die Email.
- Alice berechnet eine Signatur mittels einer *kryptographische Hash Funktion.*
- Alice verschlüsselt die Signatur mit ihre *Private Key*.
- Nun kann jeder – auch Bob – die Signatur entschlüsseln mit Alice's *Public Key*, und sicher sein, dass Alice die Signatur verschlüsselt hat.
- Verschlüsselung und Signatur sind voneinander unabhängig.



Public-Key: Signaturen & Zertifikate

X.509 ist die *de-facto Industry Standard* für Digitale Zertifikate

- Bindet eine Name zu einer *Public Key.*
- Wird selbst unterzeichnet.

Subject	Distinguished Name, Public Key
Issuer	Distinguished Name, Signature
Period of validity	Not Before Date, Not After Date
Administrative information	Version, Serial Number
Extended Information	

X509 Certificate Format

Beispiel Algorithmus 2: RSA (1)

1.RSA Algorithm

Public-Key Blockverschlüsselung.
Nach Rivet, Shamir & Adelman (RSA) benannt.

To find a key pair *e*, *d*:

1. Choose two large prime numbers, *P* and *Q* (each greater than 10100), and form:
   $N = P \times Q$
   $Z = (P-1) \times (Q-1)$
2. For *d* choose any number that is relatively prime with *Z* (that is, such that *d* has no common factors with *Z*).
   We illustrate the computations involved using small integer values for *P* and *Q:*
   $P = 13$, $Q = 17$ –> $N = 221$, $Z = 192$
   $d = 5$
3. To find *e* solve the equation:

$$e \times d = 1 \bmod Z$$

That is, $e \times d$ is the smallest element divisible by *d* in the series $Z+1, 2Z+1, 3Z+1, \ldots$.

$e \times d = 1 \bmod 192 = 1, 193, 385, \ldots$
385 is divisible by *d*
e = 385/5 = 77

Quelle:Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn 4, © Pearson Education 2005

Beispiel Algorithmus 2: RSA (1)

RSA Algorithm *fortgesetzt*

To encrypt text using the RSA method, the plaintext is divided into equal blocks of length k bits where $2^k < N$ (that is, such that the numerical value of a block is always less than N; in practical applications, k is usually in the range 512 to 1024).

k = 7, since $2^7 = 128$

The function for encrypting a single block of plaintext M is:

$E'(e,N,M) = M^e \bmod N$

for a message M, the ciphertext is $M^{77} \bmod 221$

The function for decrypting a block of encrypted text c to produce the original plaintext block is:

$D'(d,N,c) = c^d \bmod N$

Rivest, Shamir and Adelman prove d that E' and D' are mutual inverses (that is, $E'(D'(x)) = D'(E'(x)) = x$) for all values of P in the range $0 \leq P \leq N$.

The two parameters e,N can be regarded as a key for the encryption function, and similarly d,N represent a key for the decryption function.

So we can write $\mathbf{K_e = <e,N>}$ and $\mathbf{K_d = <d,N>}$, and we get the encryption function:

$E(K_e, M) = \{M\}_K$ and $D(K_d, \{M\}_K) = M$.

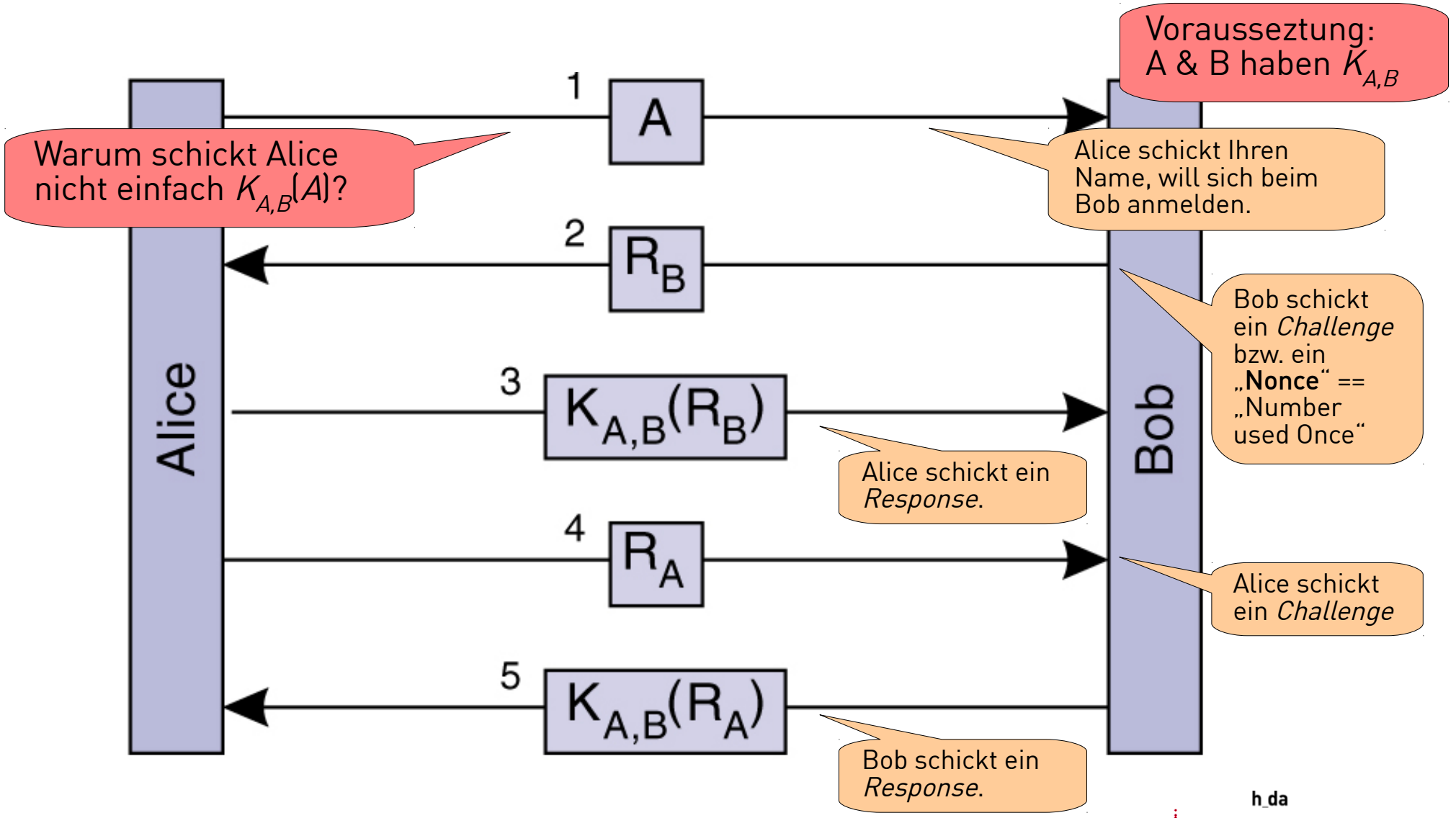
Quelle:Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn 4, © Pearson Education 2005

Kapitel VI – Sicherheit in verteilten Systemen

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Authentifizierung (1)

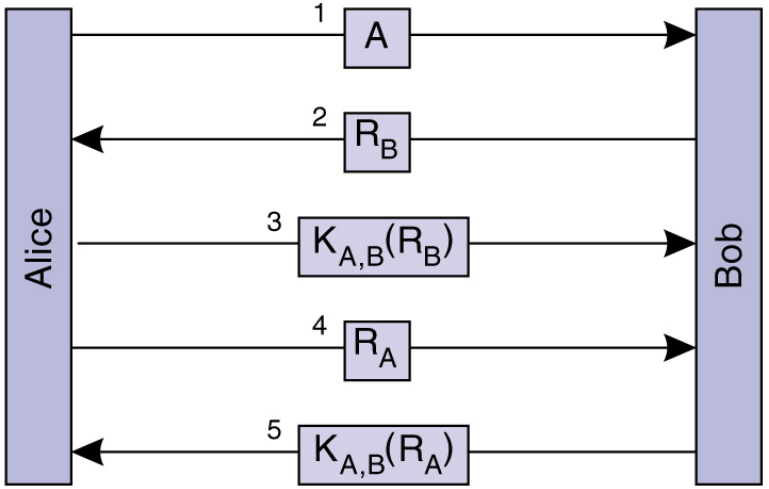
Ein einfacher (nicht empfehlenswerter) Algorithmus zur gegenseitige Authentifizierung (Voraussetzung zur Erzeugung eines Sitzungs-Schlüssels)



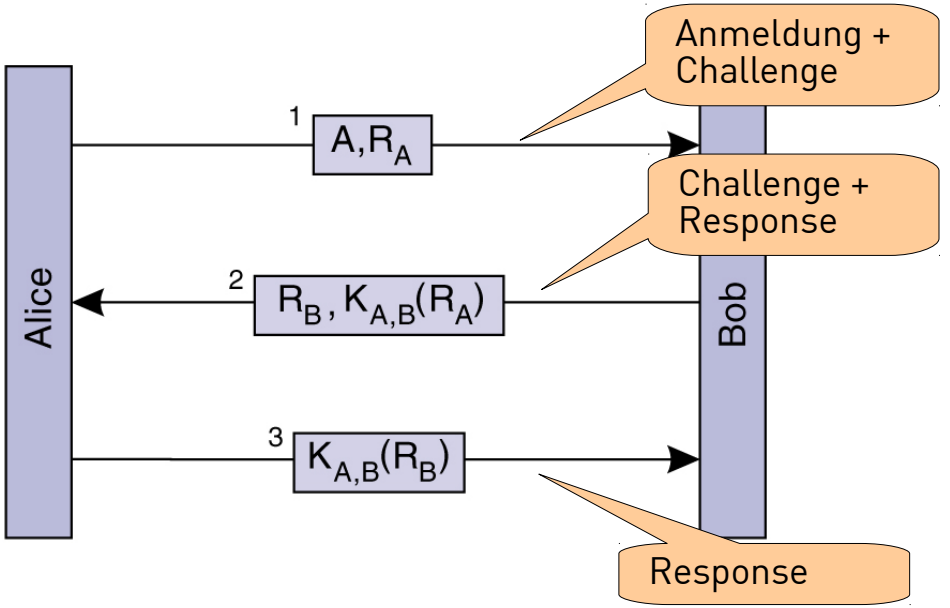
Authentifizierung (2)

Hier eine *fehlerhafte* Vereinfachung vom Protokoll: 3 Nachrichten statt 5.

Ursprungliche Authentifizierung



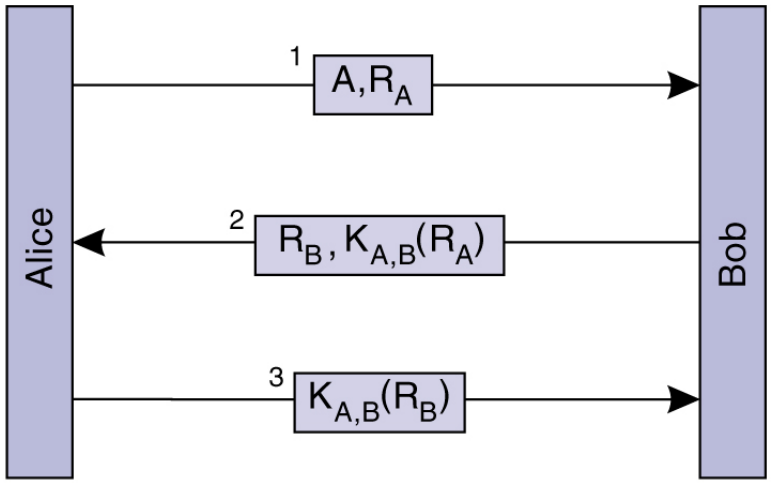
Vereinfachte Authentifizierung



Authentifizierung (3)

Die Vereinfachung wird vom **Reflektionsangriff** gefährdet.

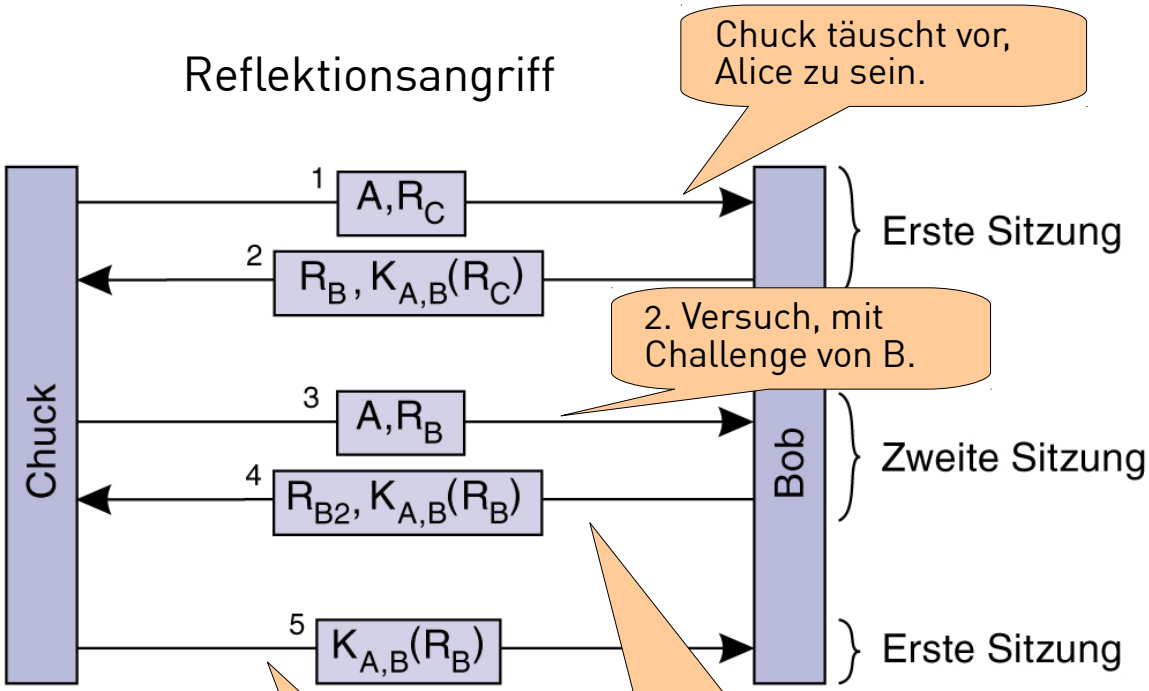
Vereinfachte Authentifizierung



Die ursprungliche (5-Nachrichten-) Protokoll hat allerdings andere Probleme, vgl. T&vS §9.2.1

Warum machen wir das Ganze? Warum verschlüsseln wir einfach nicht die ganze Sitzung mit $K_{A,B}$?

Reflektionsangriff

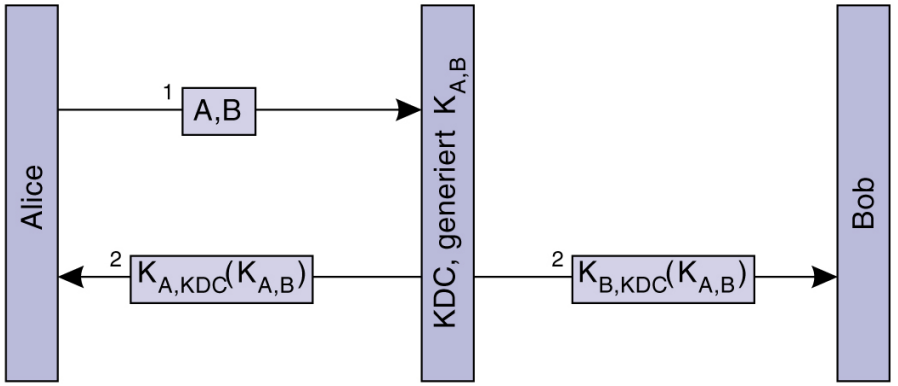


Authentifizierung (4)

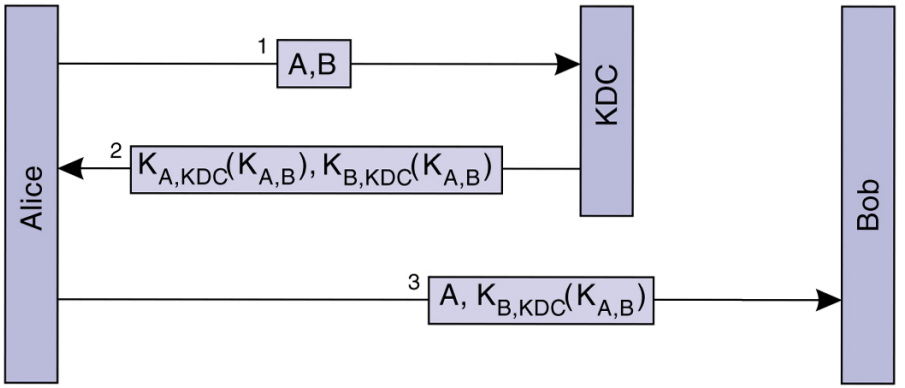
Problem: Wenn es *n* Teilnehmerinnen gibt, brauchen wir $O(n^2)$ Schlüssel.

Lösung: Verwendung eines **KDCs (Key Distribution Center).**
Jede Teilnehmerin hat einen gemeinsamen Schlüssel mit dem KDC.
Andere Schlüssel werden bei Bedarf erzeugt (*on the fly*).

Einfache Authentifizierung mit KDC



Verbesserte KDC Authentifizierung



Problem: Timing. A muss warten, bis B seinen Schlüssel hat.

Authentifizierung: Needham-Schroeder

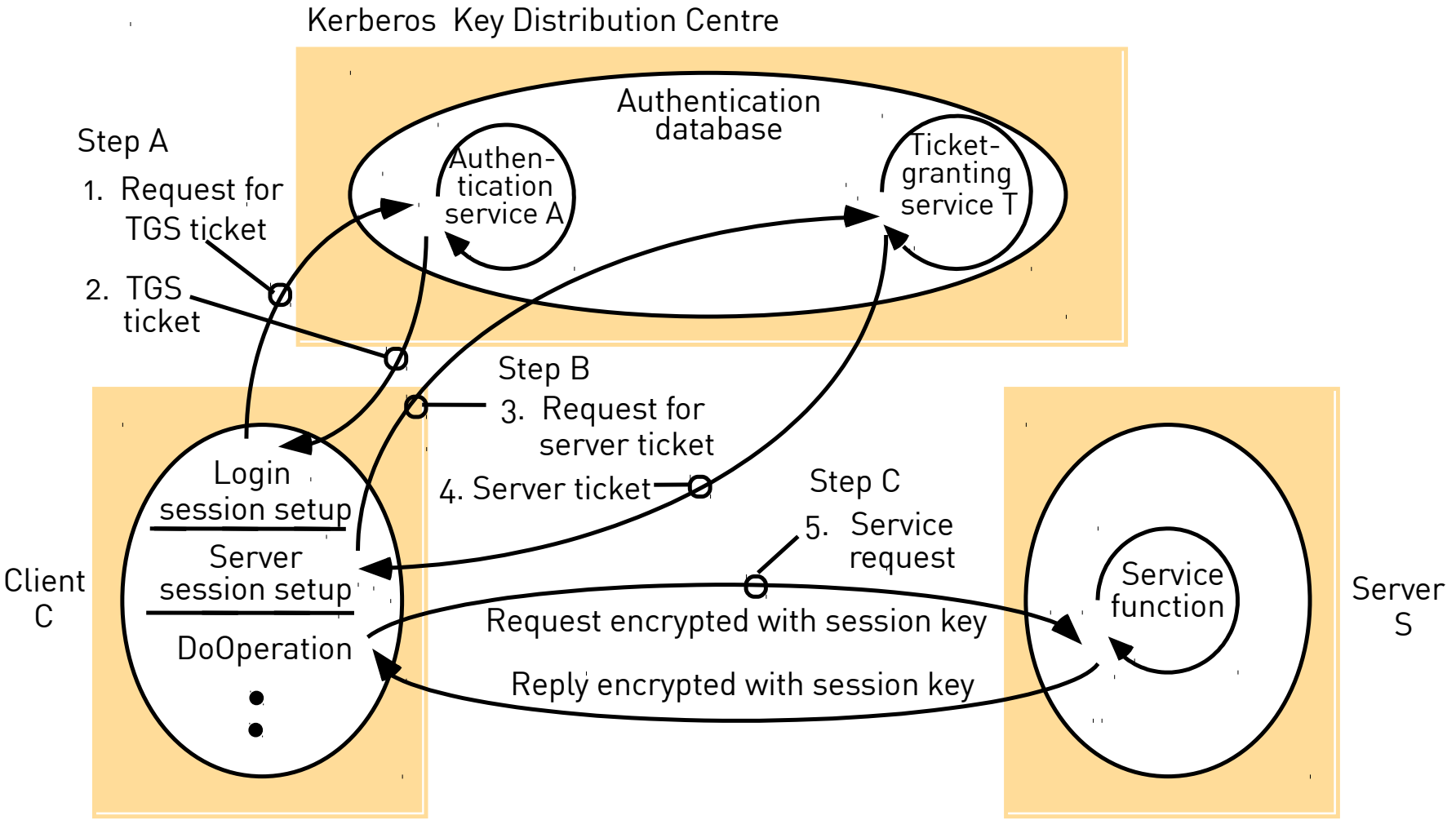
Direction	*Message*	*Notes*
1. A → S:	A, B, N_A	A requests S to supply a key for communication with B.
2. A ← S:	$\{N_A, B, K_{AB}, \{K_{AB}, A\}_{KB}\}_{KA}$	S returns a message encrypted in A's secret key, containing a newly generated key K_{AB} and a 'ticket' encrypted in B's secret key. The nonce N_A demonstrates that the message was sent in response to the preceding one. A believes that S sent the message because only S knows A's secret key.
3. A → B:	$\{K_{AB}, A\}_{KB}$	A sends the 'ticket' to B.
4. B → A:	$\{N_B\}_{KAB}$	B decrypts the ticket and uses the new key K_{AB} to encrypt another nonce N_B.
5. A → B:	$\{N_B - 1\}_{KAB}$	A demonstrates to B that it was the sender of the previous message by returning an agreed transformation of N_B.

Auch diese Protokoll hat eine Schwäche. Wie könnte man sie angreifen?

Authentifizierung in Kerberos (1)

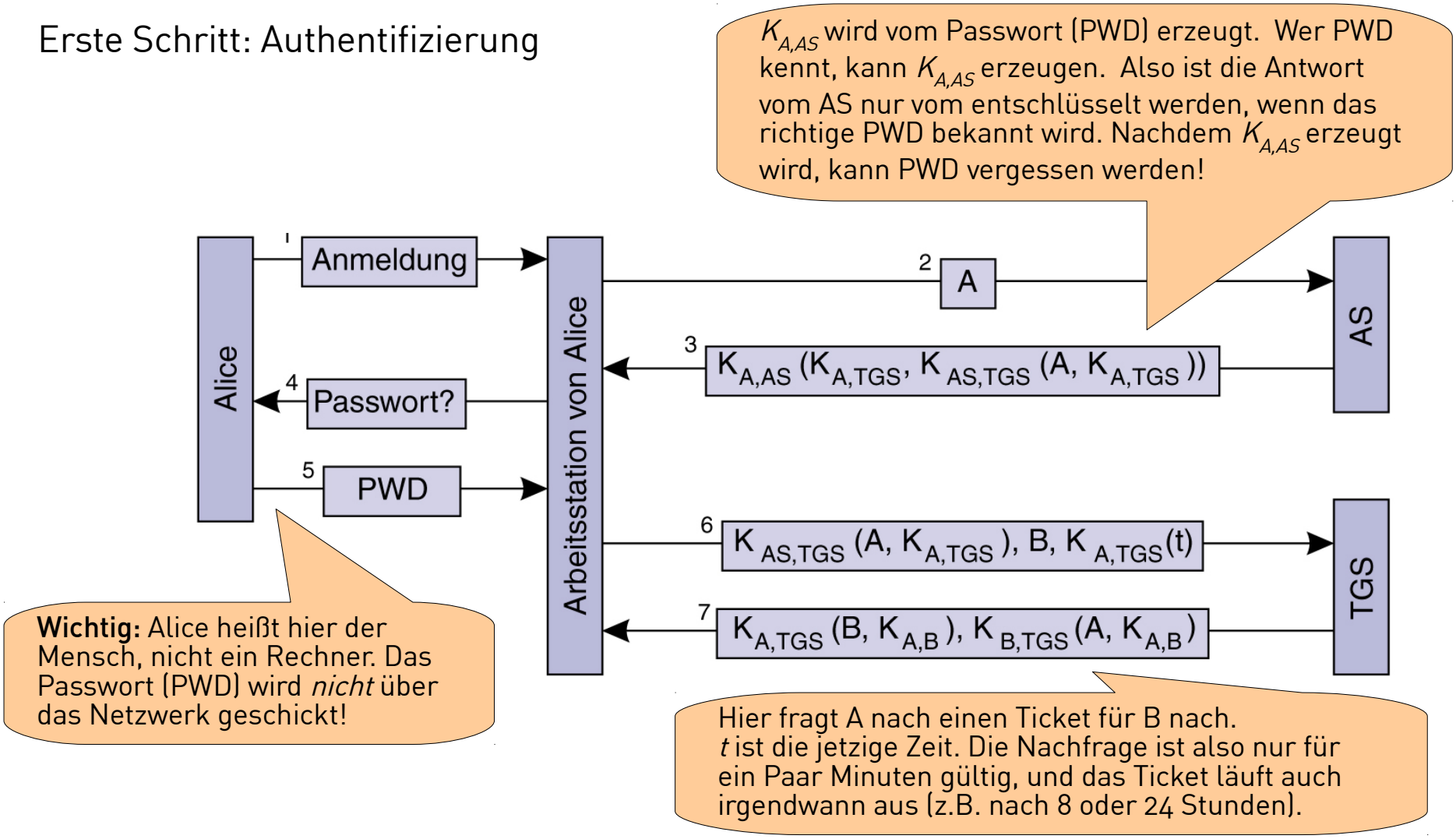
Kerberos ist ein KDC-Software, vom MIT entwickelt.
Verwendet einen verbesserten Needham-Schroeder Algorithmus.
Zur Zeit ist Version 5 verfügbar.

Architektur:



Authentifizierung in Kerberos (2)

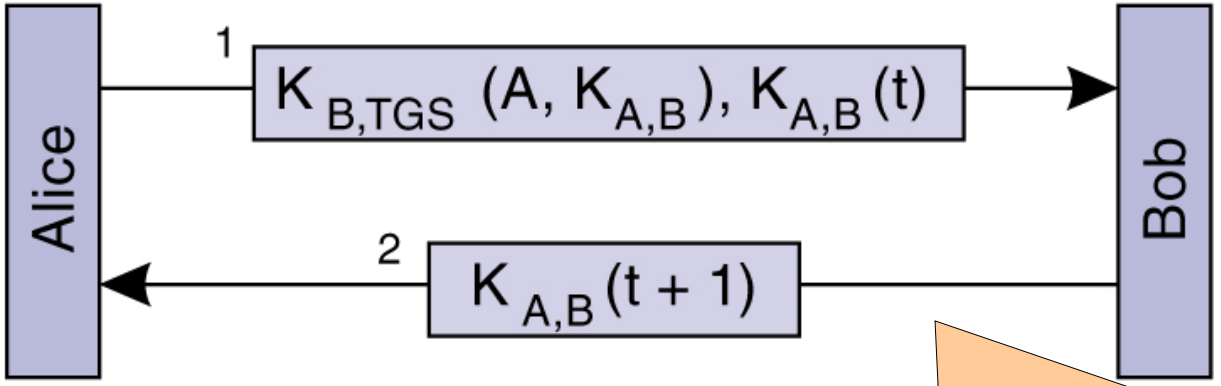
Erste Schritt: Authentifizierung



Authentifizierung in Kerberos (3)

Zweite Schritt: Einrichten eines sicheren Kanals

Ergebnis des 1. Schritts: A bekam ein Ticket mit $K_{A,B}$, einmal für sich, und einmal für B, verschlüsselt mit $K_{B,TGS}$



Single Sign-On: A kann nun eine sichere Verbindung zu andere Server einrichten, ohne As Passwort zu wissen, solange A sich die (entschlüsselte) Inhalte vom Nachricht 3 (s.o.) bemerkt hat (Nachrichten 6 & 7 müssen wiederholt werden - s.o.).

$K_{A,B}$ – schon für B vom TGS verschlüsselt – wird an B übermittelt. Auch ein Challenge wird B gegeben – die jetzige Zeit, mit dem Schlüssel $K_{A,B}$ verschlüsselt. Wenn B $K_{B,TGS}$ hat, kann er auch $K_{A,B}$ entschlüsseln, und das Response berechnen.

Kerberos: Kritik

- Wichtig: Session-Keys dürfen nur eine begrenzte Lebenszeit haben.
- Bis (inkl.) Kerberos 4 wurden *Time-Stamps* für *Nonces* verwendet.
  - Besser als Needham-Schroeder
  - Problem: Synchronisierung der Server!
- Seit Keberos 5 dürfen auch *Sequence-Numbers* verwendet werden.
  - Problem: Synchronisierung der Server nach einem Absturz
- Ob Version 4 oder 5:
  - Ist die Session-Zeit zu lang, ist sie unsicher
  - Ist sie zu kurz, ist sie nicht *user-friendly.*

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Kapitel VI – Sicherheit in verteilten Systemen

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

„Perfect“ Forward Secrecy

- **Das Problem:**
  - Nehmen wir an, dass ein Angreifer verschlüsselten Datenverkehr *heute* aufzeichnet.
  - In 4 Jahren bekommt Angreifer (irgendwie, s. Heartbleed Bug...) den privaten Schlüssel des Servers
  - Angreifer kann in 4 Jahren gesamten *aufgezeichneten* Datenverkehr von heute entschlüsseln
  - **Schwäche:** Wenn derselbe Schlüssel zur Authentifizierung und zur Verschlüsselung beim Austausch des geheimen Sitzungsschlüssels verwendet wird.

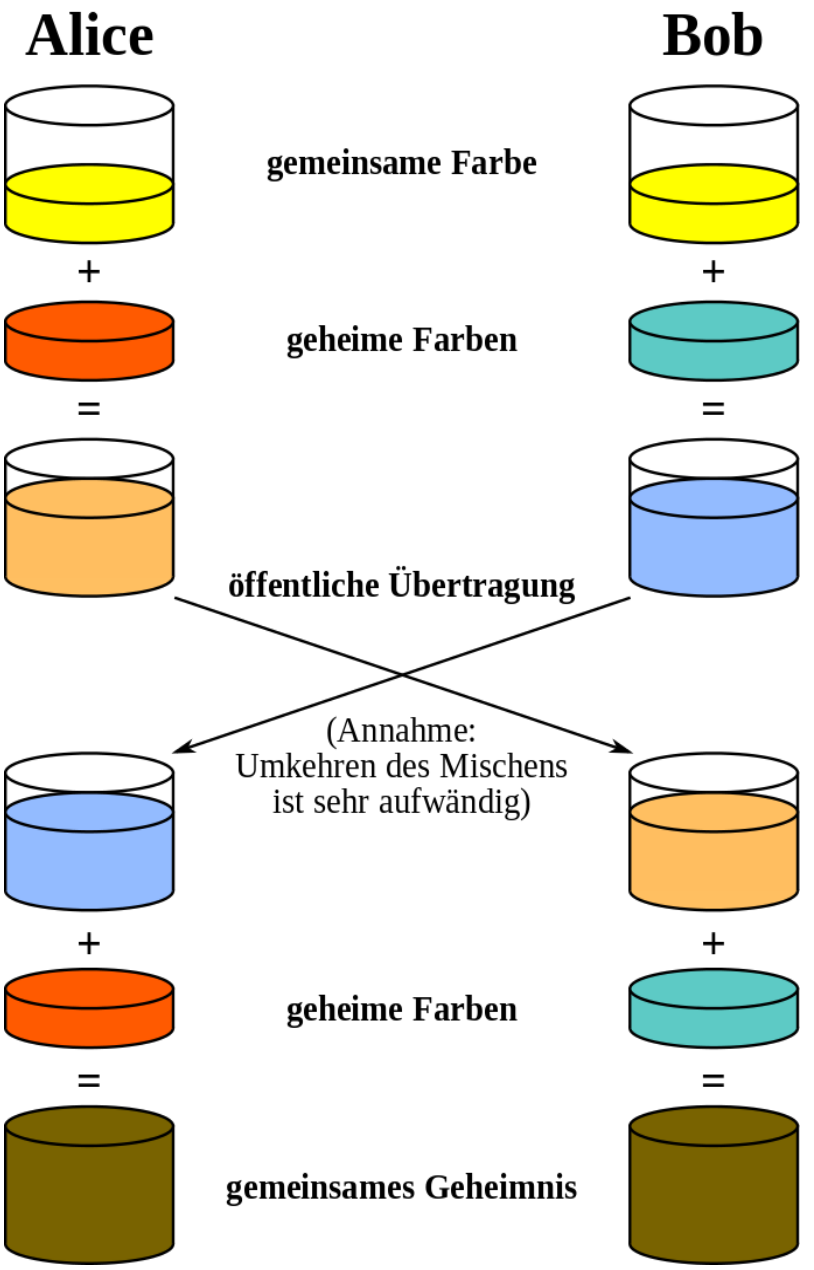
h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Idee hinter Forward Secrecy (2)

- **Design Prinzip: Kein Schlüssel sollte mehr als ein Teil der Kommunikation gefährden.**
- Also, keine lang-lebendige Schlüssel, die andere Schlüssel verschlüsseln bzw. woraus kurzzeitige Schlüssel berechnet (abgeleitet) werden.
- Idee: **Diffie-Hellman** (s.u.), mit signierte kurzzeitige Schlüsseln.

Diffie & Hellman haben dieses Verfahren in 1976 veröffentlicht, vor RSA; einer der erste asymmetrischen Algorithmen

Vorbereitung: Zeigt nur das Prinzip (es wird natürlich nicht mit Farben gearbeitet).



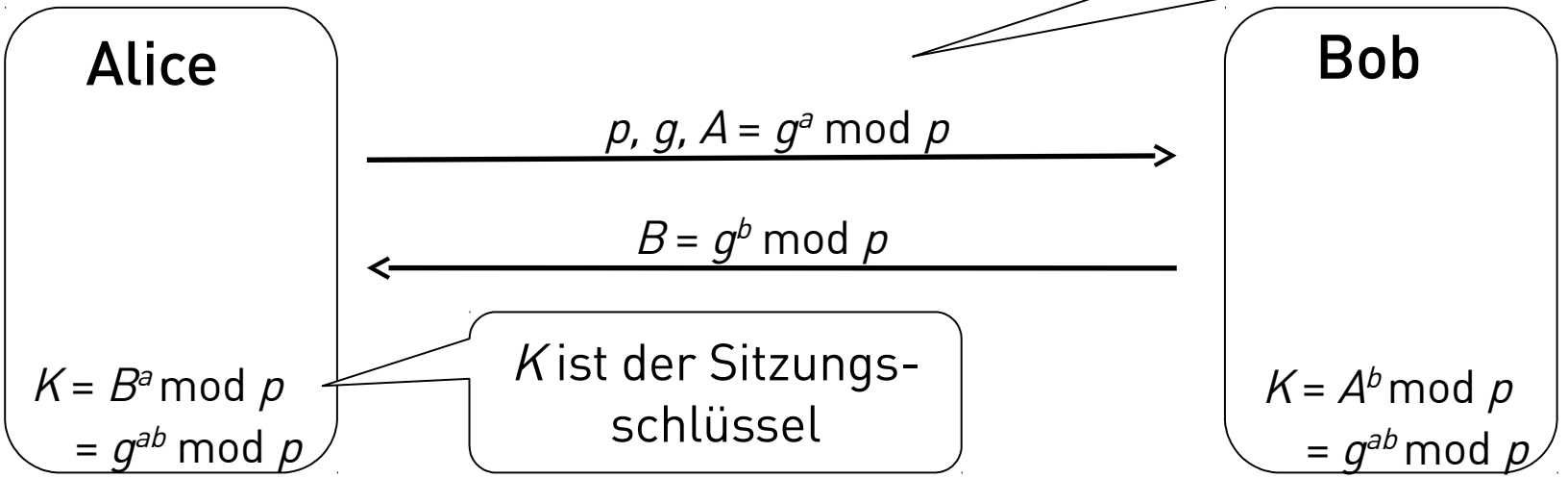
Quelle: https://commons.wikimedia.org/wiki/File:Diffie-Hellman_Key_Exchange_%28de%29.svg

Diffie-Hellman, Teil 1

- Austausch eines Sitzungsschlüssels zwischen Alice und Bob
- Öffentlich
  - Alice und Bob wählen p prim und g, sodass g ein „Primitivwurzel modulo p“ ist.
- Geheim
  - *Alice* wählt $a < p$ und Bob wählt $b < p$
- Ablauf

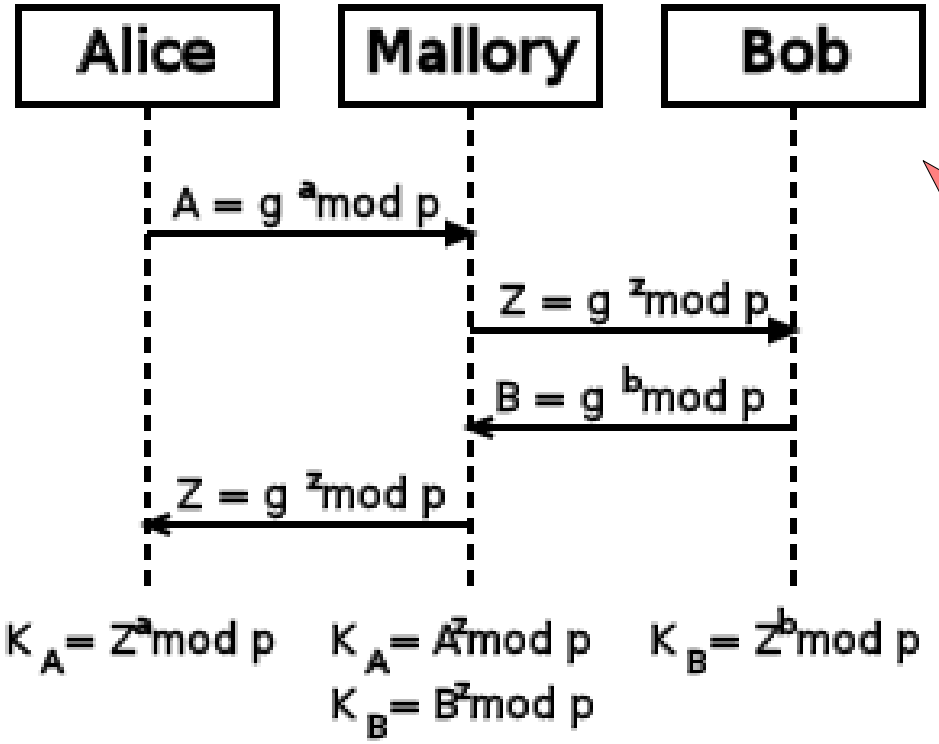
Für jede m & n, $0 \leq m,n < p$,
$m \neq n \iff g^m \bmod p \neq g^n \bmod p$,
d.h. für jede m, gibt es eine n mit
$m = g^n \bmod p$

Wichtig: diese Nachrichten werden jeweils mit dem öffentlichen Schlüssel signiert!



Diffie-Hellman und der „Man in the Middle“

- Frage: Was passiert, wenn Alice und Bob ihre Nachrichten nicht signieren?
- Antwort: *man-in-the middle attack*
- Beispiel



Diffie & Hellman's Algorithm von 1976 hatte diese Schwäche; verschiedene **Authentifizierungen** wurden später erfunden.

Mallory muss aufpassen, um sich nicht zu verraten. Warum? Was muss er unbedingt tun, um nicht zu verraten, dass er da gewesen ist?

Quelle: http://de.wikipedia.org/wiki/Diffie-Hellman-Schl%C3%BCsselaustausch

Diffie-Hellman Realisiert

- Sicherheit basiert darauf, dass
  1. *a* und *b* aus g^{ab} nicht effizient rekonstruierbar sind (diskreter Logarithmus)
  2. weder *a* noch *b* übertragen werden (auch nicht verschlüsselt)
  3. *a* und *b* werden nach der Sitzung weggeworfen (nicht persistent gespeichert)
- Problem: Berechnung der Potenzen aufwändig, kostet viel Rechenzeit
- Lösung: elliptische Kurven (hier nicht näher erläutert)
- Beispiel Webbrowser: Aktivierung der entsprechenden Ciphersuites (muss auch vom Webserver unterstützt werden)

  Beispiel: ecdh_ecdsa_aes_128_sha

DH + elliptische Kurven, AES als symm. Verfahren

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

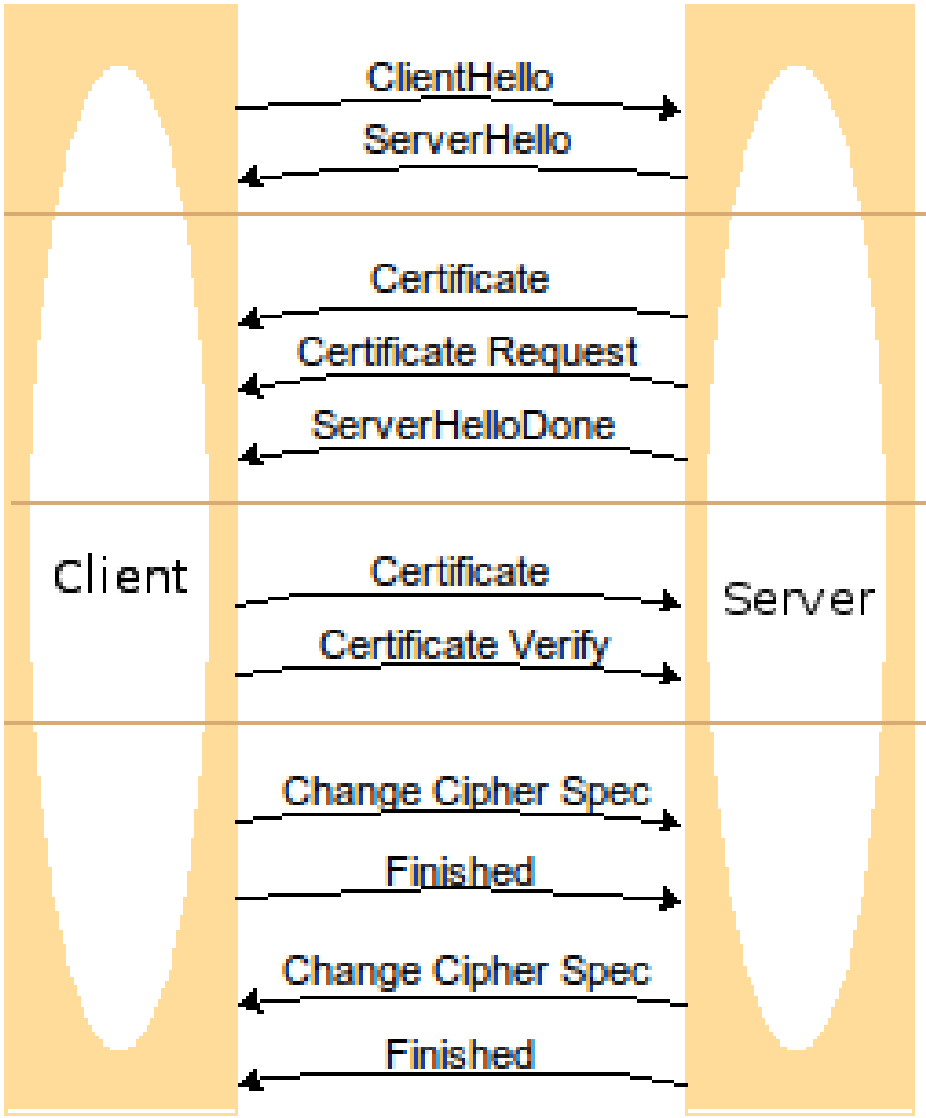
Kapitel VI – Sicherheit in verteilten Systemen

TLS: Transport Layer Security (1)

- 1996: SSL – Secure Socets Layer – wird von Netscape vorgeschlagen als Standard
- 1999: Erweiterte Version – TLS – wird als Standard vom ISO angenommen. RFC 2246.
- Inzwischen von (fast) alle Browser verwendet.
- Features / Ziele:
  1) **Negotiable algorithms.**
     Die Algorithmen für sowohl Authentifizierung als auch Verschlüsselung sind nicht festgelegt, werden am Anfang ausgehandelt.
  2) **Bootstrapped secure communication.**
     Kein 3. Teilnehmer notwendig, kein KDC notwendig:
     - Zuerst wird unverschlüsselt kommuniziert,
     - danach mit *Public-Key* Verschlüsselung,
     - danach mit *Secret-Key* Verschlüsselung.

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

TLS: Transport Layer Security (2)



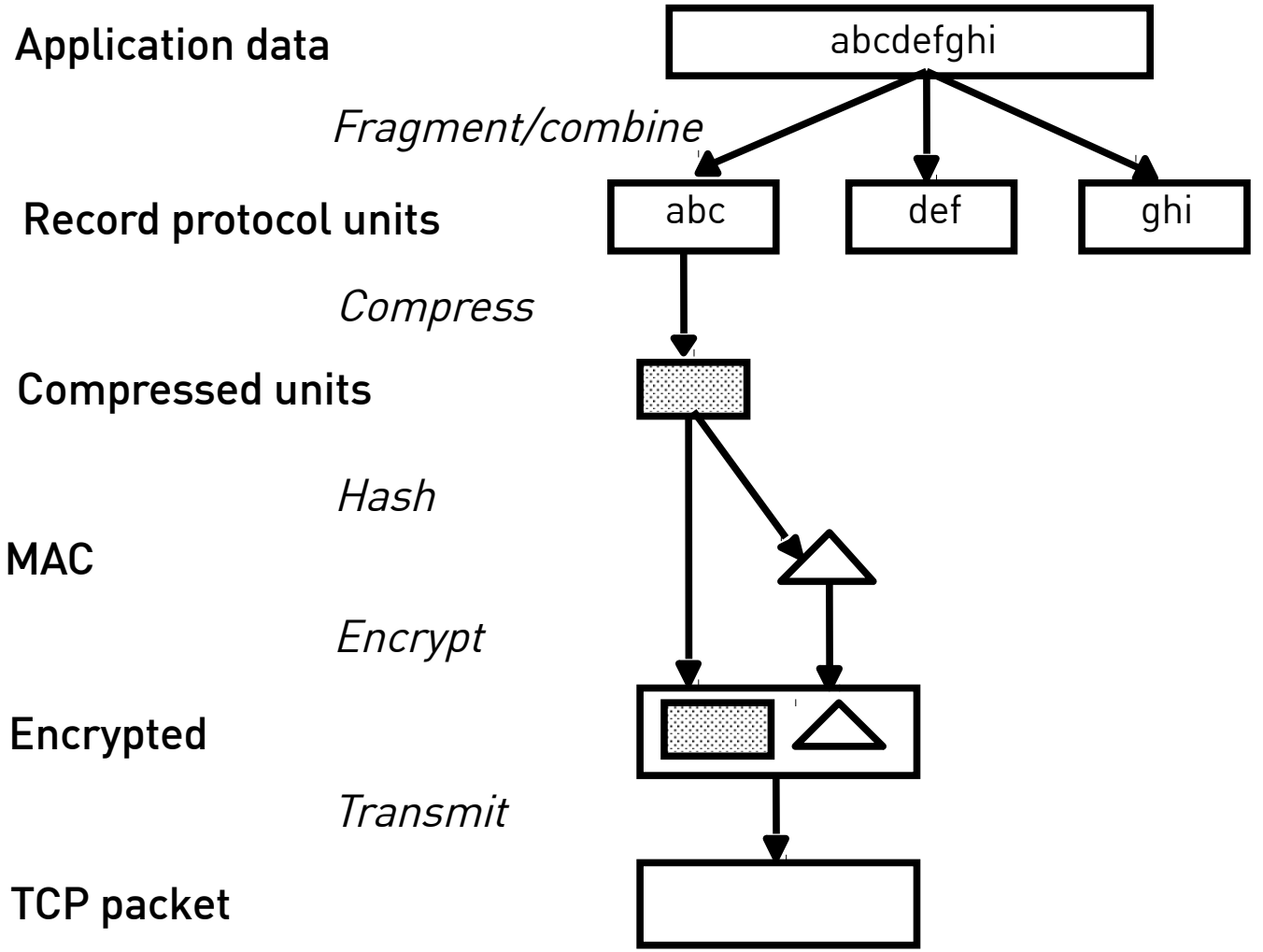
X.509 Zertifikate werden hier verwendet – können von einem Server stammen, oder einfach *on the fly* erzeugt (und danach weggeworfen) werden.

TLS: Transport Layer Security (3)

Verhandlung: *Handshake Configuration Options*

Component	*Description*	*Example*
Key exchange method	the method to be used for exchange of a session key	RSA with public-key certificates
Cipher for data transfer	the block or stream cipher to be used for data	IDEA
Message digest function	for creating message authentication codes (MACs)	SHA

TLS: Transport Layer Security (4)



Kapitel VI – Sicherheit in verteilten Systemen

WEP: Ein Fallstudie (1)

- 1999: WEP – *Wired Equivalent Privacy* – wird mit IEEE 802.11 veröffentlicht. Soll WLAN so sicher wie Ethernet-Kabel machen.
- 2003: WEP wird „deprecated“ und mit WPA – *Wi-Fi Protected Access* – ersetzt.

  Was ist schief gegangen?

- WEP hat zwei Ziele / Features:
  1) *Access Control* – Challenge/Response Protokol (vgl. Kerberos – wird verwendet, zu sehen, ob ein Neuer den *Shared Secret* kennt.
  2) *Privacy and Integity* – dasselbe Geheimnis kann für Stromverschlüsselung (Stream cipher encryption, s.o.) verwendet werden. Es gibt auch Checksums für Fehler-Erkennung

Es gibt *nur ein* Geheimnis; alle Teilnehmer müssen es kennen.

Es kann 40, 64 oder 128 bits lang sein.

WEP: Ein Fallstudie (2)

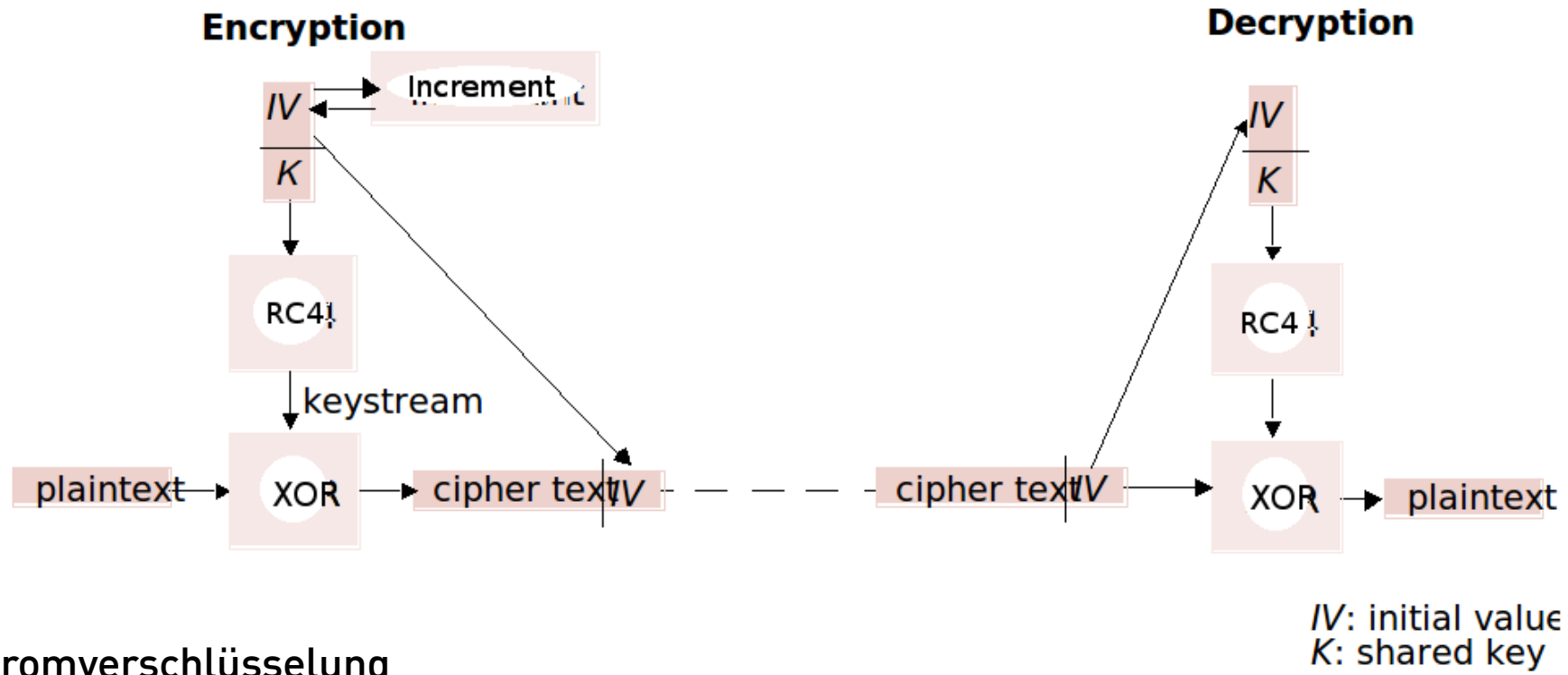
Schwachstellen von WEP

- Alle Benutzer teilen einen Schlüssel (ein Geheimnis)
  - Also wird er irgendwann verraten.
  - Nicht leicht zu erneuern / ersetzen.
  - **Lösung**: Public-Keys für individuele Schlüsseln, wie bei TLS.
- *Base Stations* werden nicht authentifiziert.
  - Möglichkeit für *Masquerading.*
  - **Lösung**: Base-Stations sollten sich ausweisen (mit einem Zertifikat).

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

WEP: Ein Fallstudie (3)

Schwachstellen von WEP - fortgesetzt



- **Stromverschlüsselung fehlerhaft**
  - Anfangswert (Initial Value, IV) muss beidseitig bekannt sein. Nur 24 Bits lang(!).
  - Also wird nach *spätestens* $2^{24} \approx 10^7$ Paketen wiederholt.
  - Zusammen mit bekannten Pakete („Cribs"), kann der Schlüssel berechnet werden
  - **Lösung**: Verhandlung von einem neuen Schlüssel viel öfter.

WEP: Ein Fallstudie (4)

Schwachstellen von WEP - fortgesetzt

- Schlüssel von 40 bzw. 64 Bits einfach zu kurz
  - Ursprünglich von US-Export-Gesetze gezwungen.
  - **Lösung:** Mindestens 128 Bits verwenden.
- RC4 Stromverschlüsselung hat schwerwiegend Fehler
  - Auch ohne Wiederholung, kann sie in wenige Minuten geknackt werden.
  - **Lösung:** Die Algorithmen sollten nicht festgelegt werden, sondern verhandelt werden (vgl. TLS). Leider war RC4 die einzige erlaube Verschlüsselung in WEP.

    Der Nachfolger heißt WPA bzw. WPA2 bzw. 802.11i
- Benutzer verwenden Verschlüsselung nicht!
  - Verschlüsselung wird als schwierig und langsam empfunden.
  - **Lösung?!?**

Update:
Spätestens nach Edward Snowden gibt es eine steigende Akzeptanz für Kryptographie, jedoch gibt es viel Platz noch für Verbesserungen – z.B. beim Email.

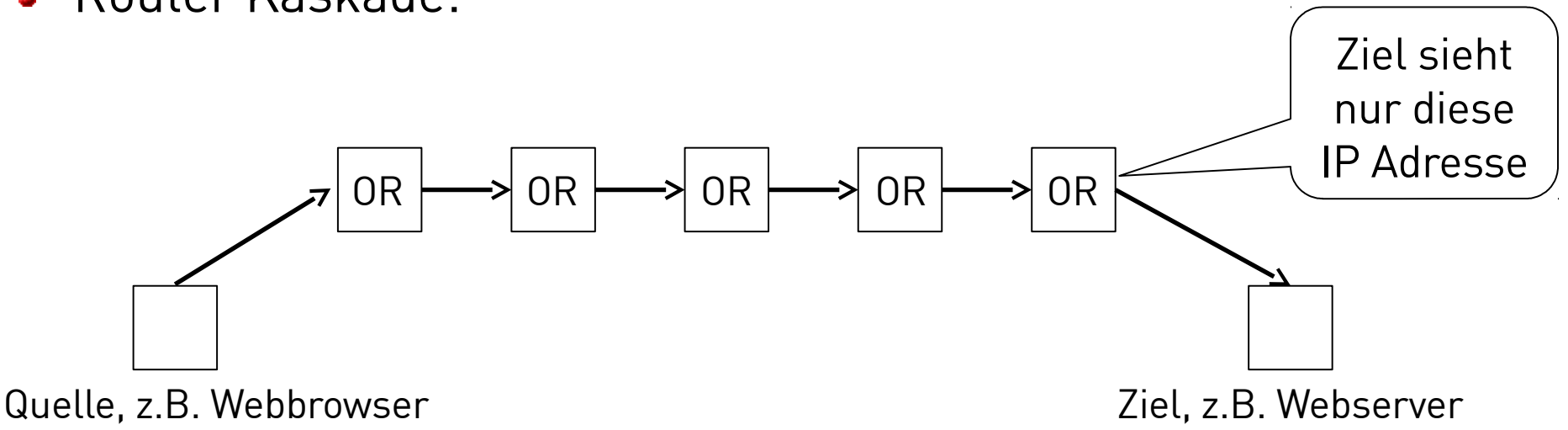
h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Kapitel VI – Sicherheit in verteilten Systemen

Anonymisierung mit Tor & Co

Was ist, wenn Alice *anonym* Bob's Website besuchen will?

- Dingledine, Matthewson, Syverson: "Tor: The Second-Generation Onion Router"
  <https://svn.torproject.org/svn/projects/design-paper/tor-design.pdf >
- Mitentwickelt von US Naval Research Labs
- Bietet Anonymität für TCP Verbindungen
- TOR besteht aus vielen einzelnen *Onion* Routern
- Router Kaskade:



Tor Kommunikation

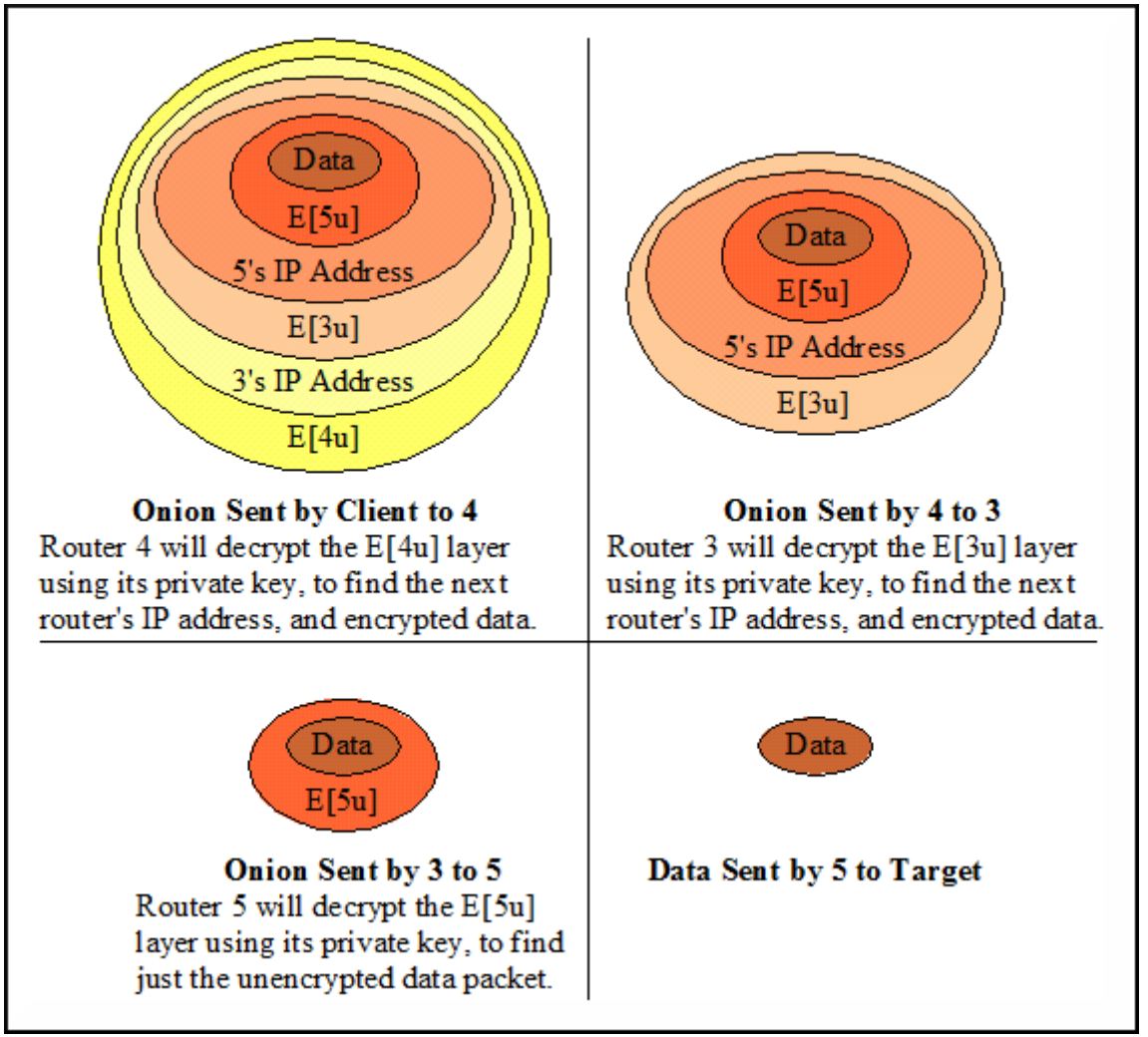
Grundlage: *Onion Routing* (D. Chaum, 1981):

- Quelle erstellt Pakete mit mehreren Schichten (wie eine Zwiebel)
- Header der Pakete enthalten Adresse des nächsten Routers der Kaskade, Checksummen, etc.

- Jeder Router kennt nur den vorigen und nächsten der Kaskade
- Quelle vereinbart einen zeitlich begrenzt gültigen Schlüssel mit jedem Router der Kaskade
- **TLS (mit *perfect forward secrecy*, s.o.) zwischen einzelnen Routern.**

Die Tor Zwiebel

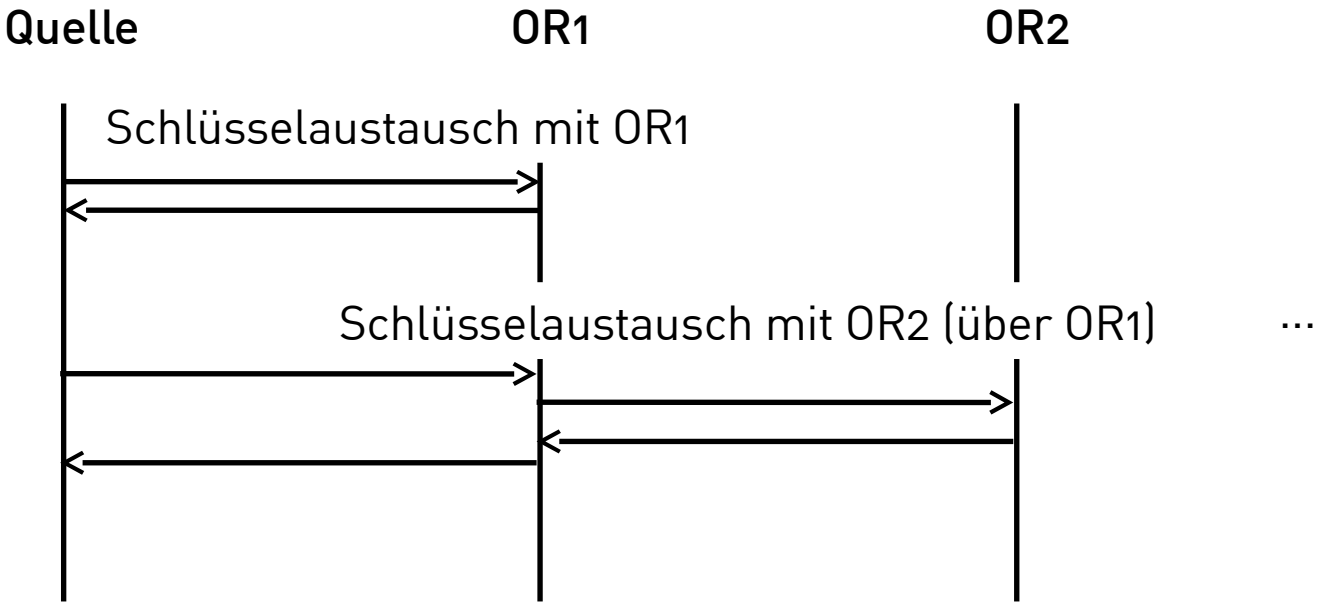
Onion Routing Beispiel vom Klient zu Server 4 zu Server 3 zu Server 5 zum Ziel



Quelle: http://ntrg.cs.tcd.ie/undergrad/4ba2.05/group10/

Tor: Erstellen der Kaskade

- Quelle wählt Menge von OR
- Quelle vereinbart einen zeitlich begrenzt gültigen Schlüssel mit dem ersten Router der Kaskade (Diffie-Hellman)
- Iteration: nutze bisherige Kaskade zum Schlüsselaustausch mit weiteren Routern



Tor: Finden der Onion-Router

- Frage: Wie findet ein Client benötigte Onion Router?

- Antwort: Namensdienst
  - bekannte TOR Knoten halten Liste von Onion Routern
  - Neu gestartete Router melden sich beim Namensdienst an

- Frage: Welche Gefahr birgt dieses System?

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Tor: Angriffsmöglichkeiten (3 Beispiele)

(1) Einschleusen von *hostile* OR

- Frage: Was bringt das?
- Antwort: Sinnvoll nur wenn *hostile OR* sowohl Ein- und Ausgangsknoten sind

(2) Timing-Analyse vor Ein- und Ausgangsknoten

- Frage: Was bringt das?
- Antwort: Korrelation zwischen Quell- und Zieldatenstrom.
  Aber: relativ schwer zu realisieren.
  Und, je mehr Leute Tor verwenden, desto schwieriger.

(3) Covert-Channel-Angriffe (verwenden Leute TOR *und andere Kommunikationskanäle?* Wie z.B. Videos mit AJAX, Torrents...)

Fazit: "TOR stinks" (sagt die NSA)

- http://www.theguardian.com/world/interactive/2013/oct/04/tor-stinks-nsa-presentation-document

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Tor: Offene Fragen

- Tor vs. VPN: Vergleich? Ähnlichkeiten, Unterschiede?
- Tor vs. TLS (https)? Ähnlichkeiten, Unterschiede?
- Tor verwendet nur 3 bzw. 4 ORs per Session.
  Wären nicht mehr besser?
- Tor ist nur für TCP gut. Was ist mit UDP?
- Wenn ich Tor verwende, bin ich anonym? Reicht Tor aus?
- Kann Tor von Kriminellen, Terroristen (et. al.)
  verwendet werden? Sollte es *Backdoors* geben?
- Ist Anonymität immer gut? Gegenbeispiele?

Vgl. https://www.torproject.org/docs/faq.html

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Kapitel VI – Sicherheit in verteilten Systemen

Zusammenfassung (verteilter IT-Sicherheit)

- Definitionen :
  Sicherheit == CIA ==Confidentiality, Integrity & Availability
- Firewalls, Proxies & VPNs
  - Sicherheit durch Mauern, Brücken bzw. Kryptographie
- Grundlagen der Kryptographie:
  Verschiedene Wege, etwas zu verschlüsseln – oder zu signieren!
- Kerberos
  Zentralisierte Authentifizierung *und* Session-Schlüssel-Vergabe
- **Perfect Forward Secrecy** durch signierte (!) Diffie-Hellman.
- TLS
  – Dezentralisierte, erweiterbare Verschlüsselung
- 802.11 Wireless LAN
  – gescheiterte Verschlüsselung
- Anonymisierung mit TOR & Co
  – Zufällige Routing zwischen Onion-Router

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK