

Verteilte Systeme

R. Kaiser, R. Kröger, O. Hahm

(HTTP: <http://www.cs.hs-rm.de/~kaiser>

E-Mail: robert.kaiser@hs-rm.de)

Kai Beckmann

Sebastian Flothow

Sommersemester 2022

6. Sicherheit



<https://blog.hackerrank.com/new-domain-tuesday-hackerrank-reveals-security-distributed-systems-domains/>

Inhalt



6. Sicherheit

- 6.1 Einführung
- 6.2 Verschlüsselungsverfahren
- 6.3 Kryptographische Hash-Funktionen
- 6.4 Authentifizierung
- 6.5 Digitale Signaturen
- 6.6 Schlüsselverwaltung
- 6.7 Protokolle und Anwendungen
- 6.8 Firewalls

Einführung



- Sicherheit

- ▶ Bedeutung hier nur im Sinne von Security
- ▶ nicht betrachtet: Sicherheit im Sinne von Safety

- Informationssicherheit (*)

- ▶ „Informationssicherheit hat zum Ziel, die Verarbeitung, Speicherung und Kommunikation von Informationen so zu gestalten, dass die *Vertraulichkeit*, *Verfügbarkeit* und *Integrität* der Informationen und Systeme in ausreichendem MaSS sichergestellt wird. Zur Zielerreichung müssen verschiedene Teilaspekte integriert betrachtet werden. Informationssicherheit bezeichnet in diesem Zusammenhang das Ziel, diese Systeme vor Gefahren bzw. *Bedrohungen* zu schützen, *Schaden* zu vermeiden und Risiken zu minimieren“.
- ▶ Trennung von Strategie (Policy, Politik) und Mechanismus
 - ★ Sicherheitsstrategie (Mengen von Regeln)
 - ★ Sicherheitsmechanismen (Mechanismen zur Durchsetzung)

(*)<http://de.wikipedia.org/wiki/Computersicherheit>

Sichere Systeme



- ... gibt es nicht.
- Die absolut sichere Firewall:



<http://www.brauwesen-historisch.de/seitenschneider.jpeg>

Schutzziele



- Allgemeine Schutzziele (CIA):

- ▶ *Vertraulichkeit (Confidentiality)*:
Information wird nur Berechtigten zugänglich
- ▶ *Integrität (Unversehrtheit, Integrity)*:
Daten dürfen nicht unbemerkt verändert werden
- ▶ *Verfügbarkeit (Availability)*:
Zugriff auf Daten ist mit vereinbarter Güte gewährleistet

- Weitere Schutzziele:

- ▶ *Authentizität (Authenticity)*:
Echtheit einer Person oder eines Dienstes ist überprüfbar
- ▶ *Verbindlichkeit (Nicht-Abstreitbarkeit, Non-Repudiation)*: Urheber von Daten muss erkennbar sein und kann dies nicht abstreiten
- ▶ *Zurechenbarkeit (Accountability)*:
Eine Aktion kann einem Benutzer zugeordnet werden
- ▶ *Privatsphäre (Privacy)*:
Personenmerkmale müssen vertraulich bleiben, und Anonymität muss möglichst gewahrt bleiben

Weitere Begriffe

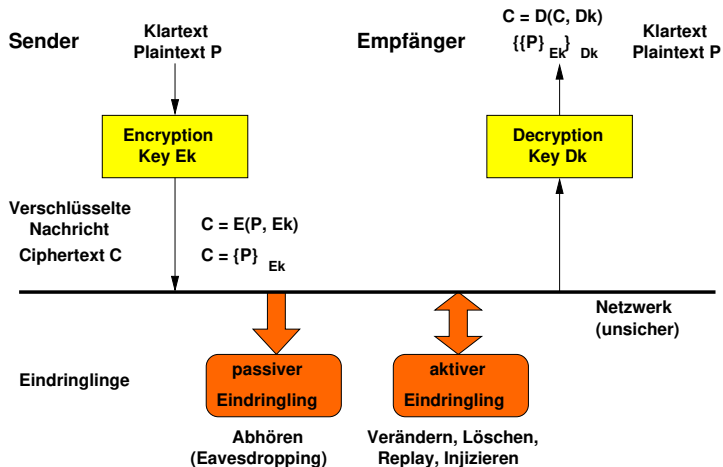


- Authentisierung, Authentifikation (authentication):
 - ▶ Verifikation einer Identität
 - ▶ beidseitige Authentifikation von Kommunikationspartnern
notwendig: z.B. Benutzer - Rechensystem und umgekehrt
- Autorisierung (authorisation):
 - ▶ Ermächtigung: Rechte haben und wahrnehmen können
 - ▶ Security-Modelle
 - ★ Discretionary Access Control
Zugriffsmatrix als abstraktes Modell
Verfahren: Capabilities, Access Control Lists (ACLs)
 - ★ Mandatory Access Control
z.B. militärische Klassifizierung, Restriktionen im Informationsfluss

Weitere Begriffe (2)



- Kryptographie: Lehre von der Übertragung geheimer Nachrichten
- Grundmodell:



Bedrohungen



• STRIDE Modell

- ▶ **S**poofing ⚡ Authenticity
- ▶ **T**ampering ⚡ Integrity
- ▶ **R**epudiation ⚡ Non-repudiability
- ▶ **I**nformation disclosure ⚡ Confidentiality
- ▶ **D**enial of Service ⚡ Availability
- ▶ **E**levation of Privilege ⚡ Authorization

Bedrohungen (Beispiele)



- Fehlerhafte Spezifikation von Sicherheitsstrategien
- Fehlerhaftes Design/Spezifikationen von Komponenten
- Fehlerhafte Konfiguration
- Fehlerhafter Code
- Schwache kryptographische Verfahren
- Ausnutzung von Insider-Wissen
- „Social Engineering“
- Lauschen / Abhören / Ausspähen (Eavesdropping)
- Denial-of-Service-Attacken
 - ▶ z.B. durch Erzeugung von Last
 - ▶ Verhinderung, ein vorhandenes Recht wahrnehmen zu können
- Diebstahl von Schlüsseln und Maskerade (Vorgeben einer anderen Identität)
- Aktives Verändern, Löschen, Wiederholen/Replay von Nachrichten
- Injizieren/Infiltrieren von Nachrichten, E-Mails, Viren, Würmern, Trojanischen Pferden, ...

Risikobewertung



<https://iso25000.com/images/figures/en/iso25010.png>

- Konflikt zu anderen Charakteristiken der Software-Qualität.
- Aufwand/Nutzen müssen abgewogen werden.
- Pro Bedrohung:
 - ▶ Potentieller Schaden (Leib und Leben, Sachschaden, Image)
 - ▶ Wahrscheinlichkeit des Auftretens
 - ▶ Wahrscheinlichkeit der Erkennung des Auftretens
- Je Höher das Risiko, desto wichtiger eine Berücksichtigung in der Sicherheitsstrategie.

Sicherheitsstrategie (Security Policy)



- Gesetzliche Vorgaben
 - ▶ Datenschutzgesetze
 - ▶ Anwendungsfeld-bezogene Gesetze
 - ★ Basel II (EU, Bankenbereich, Teil des Risiko-Managements)
 - ★ Sarbanes-Oxley Act (USA, Unternehmen, Verbesserung der Unternehmensberichterstattung, Forderung nach IT Governance)
 - ★ FDA Regulations (USA, Food and Drug Administration, starke Regulierung)
- Unternehmensorganisation, -prozesse
 - ▶ Identifikation der schützenswerten Vermögenswerte (Daten, Funktionen, Gegenstände)
 - ▶ Identifikation potenzieller Angreifergruppen
 - ▶ Identifikation der Angriffsvektoren
 - ▶ Ableiten von Schutzmaßnahmen (technische und nicht-technische)

Sicherheitsstrategie (Security Policy) (2)



Standards / Best Practices (Beispiele)

- ISO/IEC 27001:2005, „Information technology - Security techniques - Information security management systems Requirements“
 - ▶ spezifiziert Anforderungen für Herstellung, Einführung, Betrieb, Überwachung, Wartung, und Verbesserung eines dokumentierten Informationssicherheits-Managementsystems unter Berücksichtigung der Risiken innerhalb der gesamten Organisation
 - ▶ berücksichtigt sämtliche Arten von Organisationen (z.B. Handelsunternehmen, staatliche Organisationen, Non-Profit-Organisationen).
- ISO 17799:2005, „Information technology – Code of practice for information security management“
 - ▶ Kontrollmechanismen für die Informationssicherheit
 - ▶ 11 Überwachungsbereiche untergliedert in 39 Hauptkategorien, sogenannte Kontrollziele.
 - ▶ insgesamt 133 Sicherheitsmaßnahmen zur Unterstützung

Sicherheitsstrategie (Security Policy) (3)



Standards / Best Practices (Beispiele)

- IETF
 - ▶ Eigene Security Area mit ca. 20 Arbeitsgruppen
 - ▶ Spezifikationen zu TLS, SSH, IPSec ...
 - ▶ Alle RFCs enthalten Security Considerations
- Fast alle Standardisierungsgremien (IEEE, OMA, ...) betreiben Gremien zum Thema Security
- IT-Grundschutzhandbuch des Bundesamtes für Sicherheit in der Informationstechnik (BSI)
 - ▶ in Deutschland verbreitet
 - ▶ „Kochrezept“ für mittleres Schutzniveau
 - ▶ berücksichtigt neben Eintrittswahrscheinlichkeiten und potentieller Schadenshöhe auch Kosten der Umsetzung

Sicherheitsmechanismen



- Überwiegend mit kryptographischen Mechanismen:
 - ▶ Authentisierung
 - ★ von Systemen/Benutzern (entity authentication)
 - ★ von Datenpaketen (data origin authentication)
 - ▶ Integritätssicherung (integrity protection)
 - ★ häufig kombiniert mit Daten-Authentisierung
 - ▶ Verschlüsselung (encryption)
 - ▶ Schlüsselmanagement (key exchange)
 - ▶ ...
- Ohne kryptographische Mechanismen:
 - ▶ Zugriffskontrolle (access control)
 - ▶ Policy-Management
 - ▶ Einbruchserkennung (intrusion detection)
 - ▶ ...

Standardisierte Sicherheitskriterien



- Klassifizierung für sogenannte Sichere Systeme der Informationstechnik basierend auf Kriterienkatalogen.
- Kriterienkatologe finden häufig bei der Beschaffung „Sicherer IT-Systeme“ Anwendung.
- Ursprung US DoD „Orange Book“
 - ▶ TCSEC: Trusted Computer Systems Evaluation Criteria, 1983
- Deutschland:
 - ▶ Zuständigkeit heute: Bundesamt für Sicherheit in der Informationstechnik (BSI)
 - ▶ Deutsche IT-Sicherheitskriterien (Grünbuch), 1989
 - ▶ Konzept der Trennung von Funktionalität und Prüftiefe (Qualitätsstufe)

Sicherheitskriterien (2)



Europa:

- Information Technology Security Evaluation Criteria - ITSEC, 1990
- Vereinigung von TCSEC und ITSEC zu Common Criteria zur Bewertung (Gegenseitige Anerkennung von Prüfzertifikaten), 1996
- Common Criteria Version 2.1 als weltweiter Standard ISO/IEC 15408
 - ▶ Teil 1: Introduction and General Model
 - ▶ Teil 2: Security Functional Requirements
 - ▶ Teil 3: Security Assurance Requirements
- Unterscheidung Funktionalität und Vertrauenswürdigkeit
- Vordefinierte Beispielklassen (10 Funktionalitätsklassen)
- Vertrauenswürdigkeit unterscheidet zwischen Korrektheit und Wirksamkeit (Stärke niedrig, mittel und hoch)
- Bewertung des Vertrauens in die Korrektheit durch sechs hierarchische Evaluationsstufen E1 (niedrig) bis E6 (formal verifiziert) definiert.

Sicherheitskriterien (3)



Grobe Zuordnung:

ITSEC	TCSEC	Bedeutung
-, E0	D	kein Schutz; unwirksam
F-C1, E1	C1	Schutz gegen absichtliche Verstöße einfacher Angreifer; einfache Sicherheit informelle Spec., Funktionstest, gezielte Angriffe
F-C2, E2	C2	Schutz gegen absichtliche Verstöße einfacher Angreifer; login-Mech., getrennte User-Daten, logging, informelle Detail-Spec.
F-B1, E3	B1	guter Schutz; Sicherheitsmodell, regelbasierte Schutzstufen, Analyse des Quellcodes bzw. des Hardwarelayouts
F-B2, E4	B2	guter Schutz; formales Sicherheitsmodell, sicherer Datenfluss bei Authentisierung Formales Sicherheitsmodell, semiformale Detailspezifikation
F-B3, E5	B3	sehr guter Schutz; Referenzmonitor-Eigenschaften, Detailspezifikation nachvollziehbar auf Quellcode abbildbar
F-B3, E6	A1	zur Zeit nicht zu überwinden formale Spezifikation und Verifikation

Material



- Viele Quellen, z.B.
 - ▶ Claudia Eckert: IT-Sicherheit. Konzepte, Verfahren, Protokolle, Oldenbourg-Verlag, 2003
 - ▶ J. Plate, J.Holzmann: Sicherheit in Netzen, Skript FH München
 - ▶ RSA Laboratories: RSA Laboratories' Frequently Asked Questions About Today's Cryptography, Version 4.1, 2000, <http://www.rsasecurity.com/>

Verschlüsselungsverfahren



- Überblick

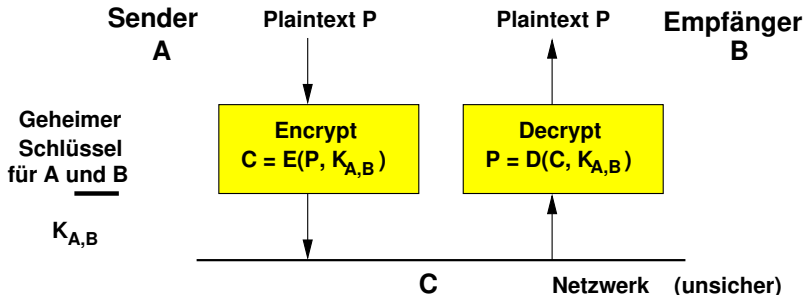
- ▶ Eingeschränkte Algorithmen
- ▶ Symmetrische Verfahren
- ▶ Asymmetrische Verfahren

- Eingeschränkte Algorithmen

- ▶ Geheimhaltung der Arbeitsweise
- ▶ heute als unbrauchbar angesehen (historisch)
- ▶ Beispiel: Caesar-Code: Verschieben um n Zeichen im Alphabet

Symmetrische Verfahren (private key encryption)

- **ein geheimer** Schlüssel für Ver- und Entschlüsselung
- sicherer Kanal zur Verteilung von Schlüsseln notwendig
- Vorteile:
 - ▶ kurze Schlüssel (ab mindestens 128 bit heute als brauchbar anzusehen)
 - ▶ geringer Rechenaufwand (schnell)
- Probleme:
 - ▶ Schlüsselaustausch und -verwaltung (Key Management)
 - ▶ keine Verbindlichkeit



Symmetrische Verfahren (2)



• Block-Algorithmen

- ▶ Verschlüsselung von Daten fester Länge, z.B. 64 Bit
- ▶ Alternativen:
 - ★ Electronic Code Book
 - alle Blöcke werden unabhängig voneinander verschlüsselt
 - ★ Cipher Block Chaining
 - Verschlüsselung hängt vom vorangehenden verschlüsselten Block ab (XOR)

• Strom-Algorithmen

- ▶ Bit/Byte-Strom-orientiert
- ▶ i.d.R. sehr schnell, kaum standardisiert

• Beispiele:

- ▶ DES Data Encryption Standard (US) historisch verbreitetster Vertreter
- ▶ Triple-DES, IDEA, AES
- ▶ RC4 (Strom-Algorithmus)

Asymmetrische Verfahren (public key encryption)

- Schlüssel besteht aus Paar (geheimer Schl., öffentlicher Schl.)
 - ▶ unterschiedliche Schlüssel für Ver- bzw. Entschlüsselung (daher der Name „asymmetrisch“)
 - ▶ Annahme: geheimer Schlüssel kann mit verfügbarem Rechenaufwand nicht aus öffentlichem Schlüssel und Verfahren rekonstruiert werden
- Vorteile:
 - ▶ kein sicherer Kanal zur Verteilung von Schlüsseln notwendig, geheimer Schlüssel wird nie übertragen
 - ▶ öffentliche Schlüssel leicht verteilbar (Verzeichnisdienst)
 - ▶ Verbindlichkeit erreichbar
- Probleme:
 - ▶ relativ lange Schlüssel notwendig (ab 2048 bit heute als gut angesehen)
 - ▶ hoher Rechenaufwand
 - ▶ Vertrauenswürdiges Schlüsselmanagement

Asymmetrische Verfahren (2)



Vertreter

- RSA-Algorithmus
 - ▶ Rivest, Shamir, Adelman: 1978
 - ▶ basiert auf Primfaktorzerlegung großer Zahlen als schwieriges Einwegproblem
- Diffie-Hellman
 - ▶ Aufbau von sicheren Verbindungen aus einem unsicheren Zustand (ohne Authentifizierung)
- Elliptische Kurven (Elliptic Curve Cryptography – ECC)
 - ▶ anderes/neues math. Verfahren als Basis
 - ▶ geringere Schlüssellänge für gleiche Sicherheit
 - ▶ besonders geeignet für ressourcenbeschränkte Systeme

Typische Verwendung



- Asymmetrische Verschlüsselung (Public Key) für
 - ▶ Authentifizierung
 - ▶ Digitale Signaturen
 - ▶ Schlüsselmanagement
 - Symmetrische Verschlüsselung (Private Key) für
 - ▶ schnelle Verschlüsselung großer Datenmengen
- ⇒ Nutzung von asymmetrischen Verfahren, um Schlüssel für anschließende symmetrische Verschlüsselung auszuhandeln (Session Key)

Beispiel 1: DES (Data Encryption Standard)



- symmetrisch
- US-Standard
- Blockverschlüsselung (64 Bit)
- Schlüssellänge 56 Bit (+ 8 Bit Parity)
- Vorgehensweise
 - ▶ aus 56 Bit-Schlüssel Ableiten von 16 Schlüsseln der Länge 48 Bit
 - ▶ Permutation und inverse Permutation am Anfang und am Ende
 - ▶ 16 Verschlüsselungsrunden (Shift, XOR, Mischfunktion)
- schnelle Implementierung in Hardware
 - ▶ spezielle DES-Chips vorhanden
- gilt heute mit 56 Bit Schlüssellänge nicht mehr als sicher
- Ersatz: Triple DES (3DES)
 - ▶ Mehrfachdurchläufe von DES
 - ▶ effekt. Schlüssellänge 112 Bit
 - ▶ Nachfolger: AES
 - ▶ in Software relativ langsam

Beispiel 2: IDEA (Int. Data Encryption Algorithm)

- symmetrisch
- Ursprung: ETH Zürich, Fa. ASCOM (Schweiz), 1992
- Patent in Europa bis 2011,
für nicht-kommerzielle Zwecke frei verfügbar
- Blockverschlüsselung (64 Bit)
- Schlüssellänge 128 Bit
- 8 iterative Runden, 6 Teilschlüssel je Runde
- in Software effizienter zu berechnen als DES
- gilt als sicherer als DES

Beispiel 3: AES (Advanced Encryption Standard)

- symmetrisch
- Name: Rijndael
- Ursprung: Joan Daemen, Vincent Rijmen (Belgien)
- Neuer US-Standard, 2001:
- Federal Information Processing Standard (FIPS) for the
- Advanced Encryption Standard, FIPS-197
- DES-Nachfolger
- Informationen: <http://csrc.nist.gov/encryption/aes/>
- Blockverschlüsselung
 - ▶ 3 unterstützte Blocklängen 128, 192 und 256 Bit
- relativ performant
 - ▶ in Software deutlich schneller als 3DES
 - ▶ Realisierung in Hardware möglich, AES-Koprozessoren
 - ▶ auch für Smartcards geeignet
- Verwendung in PGP v2.0 und OpenPGP

Beispiel 4: RC4



- symmetrisch
- Strom-Algorithmus (Byte-orientiert)
- Ursprung: Rivest (RSA Data Security)
- variable Key-Länge bis 2048 Bit
- schnell in Software
- wurde z.B. genutzt für Datei-Verschlüsselung und war in SSL/TLS wählbar
- seit 2015 für TLS verboten, da effiziente Angriffe bekannt geworden sind

Beispiel 5: RSA



- asymmetrisch
- Ursprung: Rivest, Shamir, Adleman, 1977
CACM Vol.21 No.2, Febr. 1978
- Blockverschlüsselung
- Schlüssellänge > 100 Dezimalstellen
z.B. RSA-129 (129 Dezimalstellen \Rightarrow 429 Bit)
- 1024 Bit = 105474-facher Aufwand
2048 Bit = $2.97 \cdot 10^{10}$ -facher Aufwand
- groSSe Schlüssellängen bieten noch Schutz
- sehr hoher Rechenaufwand (ca. 100-1000 mal Aufwand DES)

RSA (2)



Mathematische Basis

- Primfaktorzerlegung großer Zahlen
- Bitkette eines Blocks m des Klartextes wird interpretiert als ganze Zahl $\leq n$
- Verschlüsselung $c = E(m, K^+) = E(m, (e, n)) = m^e \bmod n$
- Entschlüsselung $m = D(c, K^-) = D(c, (d, n)) = c^d \bmod n$
- für e , d und n muss gelten: $m^{e \cdot d} = m \bmod n$ für alle $m < n$
- Bestimmung von e und d :
 - ▶ Bestimme zwei große Primzahlen p und q
 - ▶ $n := p \cdot q, z := (p - 1) \cdot (q - 1)$
 - ▶ Wähle d relativ prim zu z
 - ▶ Bestimme e mit $e \cdot d = 1 \bmod z$
 - ▶ öffentlicher Schlüssel: (e, n)
 - ▶ geheimer Schlüssel: (d, n)

Kryptographische Hash-Funktionen



- Bildung eines digitalen Fingerabdrucks
über Dokumenten/Nachrichten, genannt Message Digest
- Basis für digitale Signaturen
- Hash-Funktion H
 - ▶ $h = H(P)$
 - ▶ Nachricht P beliebiger Länge
 - ▶ h Bitkette fester Länge (z.B. 128 Bit)
 - ▶ vergleichbar zu CRC zur Fehlererkennung
- Annahmen
 - ▶ Berechnung von H ist einfach
 - ▶ Umkehrung, d.h. Ermittlung einer Ausgangsnachricht bei gegebenem Hashwert, ist schwierig (Einwegfunktion)
 - ▶ Veränderung am Dokument (P) führt zu anderem Hashwert (h)

Beispiel 1: MD5



● MD5 (Message Digest 5)

- ▶ Rivest, 1991
- ▶ 128-Bit Hashwert
- ▶ Vorgehensweise
 - ★ ausgelegt auf 32-Bit-Prozessoren
 - ★ basierend auf MD4 (gilt als nicht sicher)
 - ★ Zerlegung (bzw. Auffüllen) der Nachricht in Stücke von 448 Bit
 - ★ Anfügen der Gesamtlänge als 64 Bit-Zahl
(→ 512 Bit-Blöcke mit je 16 Unterblöcken der Länge 32 Bit)
 - ★ Ausgehend von einem konstanten Digest wird mit jedem neuen Block ein neuer Digest bestimmt (Phase)
 - ★ Jede Phase besteht aus 64 Iterationen mit 32-Bit-Funktionen über (AND, OR, NOT) über den Unterblöcken
- ▶ bekannt, früher von pgp verwendet
- ▶ erste Kollisionen bekannt
- ▶ gilt mittlerweile als unsicher

Beispiel 2: SHA-0, SHA-1, SHA-2, SHA-3



- SHA-1 (Secure Hash Algorithm)
 - ▶ Weiterentwicklung vom SHA-0 (Konstruktionsfehler, 1994)
 - ▶ 160-Bit Hashwert
 - ▶ Standard ANSI X9.30
 - ▶ Berechnung etwas aufwendiger als MD-5
 - ▶ Rechenintensive Angriffe bekannt
- SHA-2
 - ▶ Weiterentwicklung vom SHA-1 (ab 2001) SHA-224, SHA-256, SHA-384 und SHA-512 (Hashwertlänge in Bit)
 - ▶ 512/1024-Bit blockweise Verarbeitung, Dokumente bis 2128 Bit
 - ▶ Gilt noch als sicher (NIST)
- SHA-3 (Keccak) (Bertoni, Daemen, Peeters, Van Assche / Italien, Belgien)
 - ▶ Gewinner NIST-Ausschreibung (2012)
 - ▶ Alternative zu SHA-2
 - ▶ Modifikationen vom NIST vorgeschlagen (kritisch)

Authentifizierung

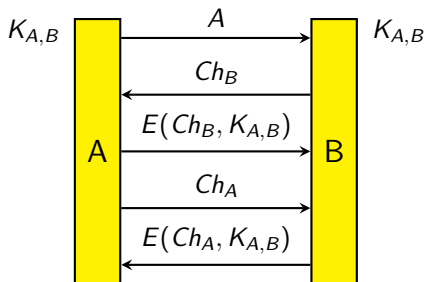


- Authentifizierung und Nachrichtenintegrität sind nicht voneinander trennbar
 - ▶ Was nützt Authentizität, wenn Nachricht verändert sein kann?
 - ▶ Was nützt Integrität einer Nachricht, wenn sie von jemand anderem kommen kann?
- Vorgehensweise
 - ① zuerst sicheren Kanal einrichten mit gegenseitiger Authentifizierung
 - ② dann geheimen Sitzungsschlüssel verwenden, um Integrität und Vertraulichkeit sicherzustellen

Authentifizierung bei geheimem Schlüssel



- Prinzip eines Challenge-Response-Protokolls



$K_{A,B}$: gemeinsamer geheimer Schlüssel

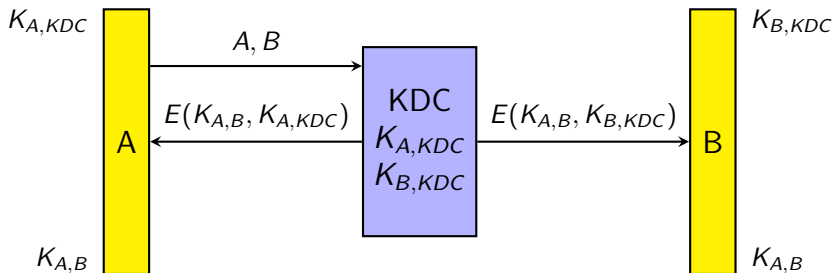
- Problem: Verwaltung vieler geheimer Schlüssel

- Kommunikationswunsch A, enthält Identität A
- Challenge Ch_B (z.B. Zufallszahl) gestellt von B
- B kann überprüfen, ob Ch_B in Antwort enthalten (\rightarrow nur A kann Partner sein)
- analog für andere Richtung (\rightarrow nur B kann Partner sein)

Authentifizierung bei geheimem Schlüssel (2)



- Problem (s.o.): Verwaltung vieler geheimer Schlüssel
- Lösung: Schlüsselverteildienst (Key Distribution Center, KDC)
- Prinzip
 - ▶ Jeder Teilnehmer hat geheimen Schlüssel mit KDC
 - ▶ KDC generiert geheimen Schlüssel $K_{A,B}$ für gewünschten Kanal und verteilt diesen Schlüssel an beide Partner

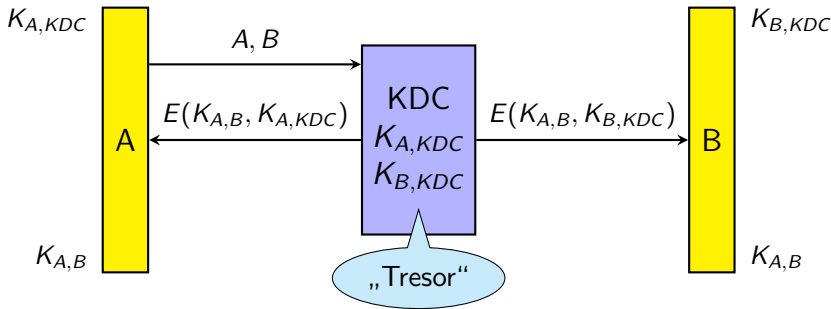


- Problem: Synchronisation zwischen A und B

Authentifizierung bei geheimem Schlüssel (2)



- Problem (s.o.): Verwaltung vieler geheimer Schlüssel
- Lösung: Schlüsselverteildienst (Key Distribution Center, KDC)
- Prinzip
 - ▶ Jeder Teilnehmer hat geheimen Schlüssel mit KDC
 - ▶ KDC generiert geheimen Schlüssel $K_{A,B}$ für gewünschten Kanal und verteilt diesen Schlüssel an beide Partner



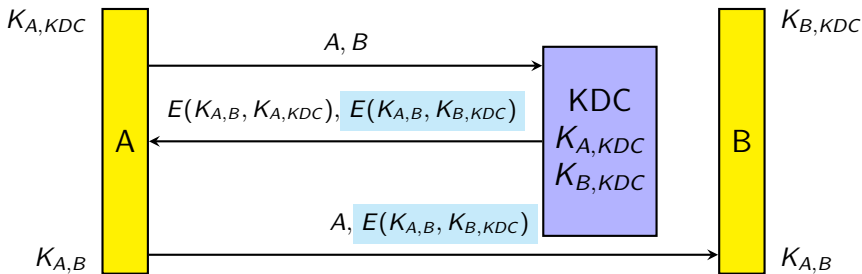
- Problem: Synchronisation zwischen A und B

Authentifizierung bei geheimem Schlüssel (3)



- Lösung:

- ▶ Schlüsselverteildienst (Key Distribution Center, KDC)
- ▶ Einführung von Tickets

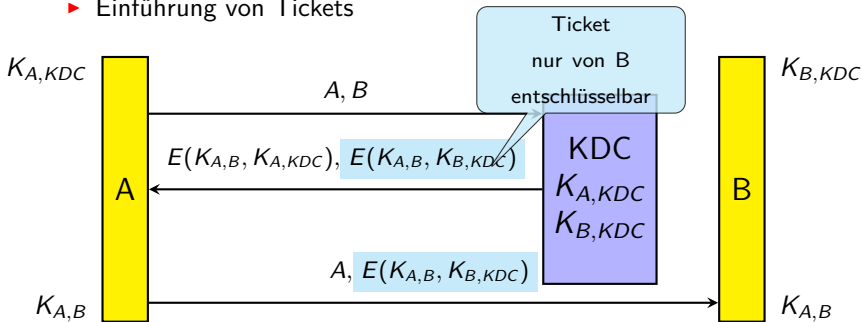


Authentifizierung bei geheimem Schlüssel (3)



• Lösung:

- ▶ Schlüsselverteildienst (Key Distribution Center, KDC)
- ▶ Einführung von Tickets

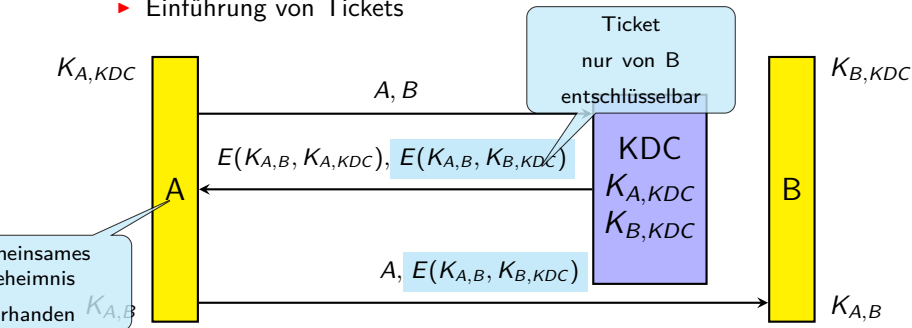


Authentifizierung bei geheimem Schlüssel (3)



• Lösung:

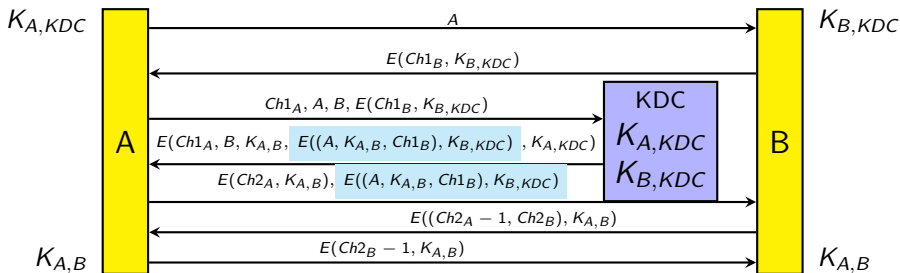
- ▶ Schlüsselverteildienst (Key Distribution Center, KDC)
- ▶ Einführung von Tickets



Authentifizierung bei geheimem Schlüssel (4)



- Weiterentwicklung: Needham-Schroeder-Protokoll
 - ▶ Needham, Schröder, 1978
 - ▶ Sicherungen gegen wiederholtes Einspielen von Nachrichten
 - ▶ Variation dieses Protokolls in Kerberos verwendet (vgl. 6.7)

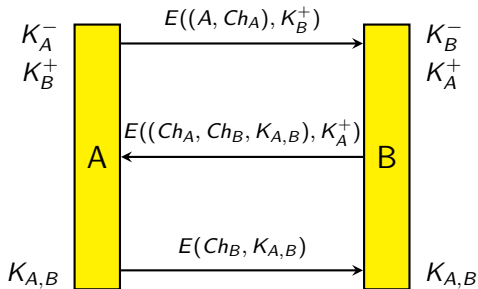


Authentifizierung mit öffentl. Schlüssel



• Prinzip

- ▶ kein KDC erforderlich
- ▶ Zuordnung der öffentlichen Schlüssel zu den wahren Personen muss gewährleistet sein



- K_A^- geheimer Schlüssel von A
- K_A^+ öffentl. Schlüssel von A
- $K_{A,B}$ Sitzungsschlüssel, von B erzeugt, kurzlebig

Digitale Signaturen

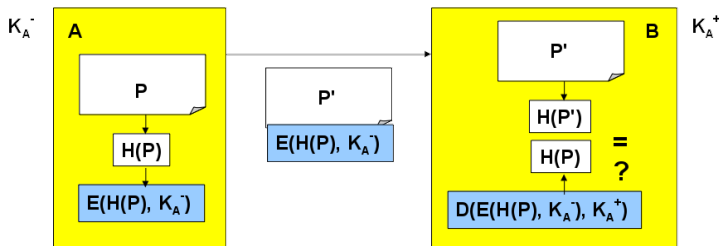


- Bedeutung wie Unterschrift
 - ▶ nicht vom unterschriebenen Dokument zu trennen
 - ▶ nicht (leicht) fälschbar
- Signatur bietet zuverlässige Feststellung von
 - ▶ Urheberschaft
 - ▶ Nichtabstreitbarkeit
 - ▶ Integrität
 - ▶ Authentizität
- schützt nicht Vertraulichkeit des Inhalts
 - ▶ dazu ist zusätzlich Verschlüsselung notwendig (s.u.)
- Kombination aus
 - ▶ Hash-Algorithmus
 - ▶ Public Key-Infrastruktur
- Europa besitzt bzgl. digitalen Signaturen relativ starke Stellung

Vorgehensweise



- Signieren durch Verschlüsselung des Hash-Werts einer Nachricht mit privatem Schlüssel
- Öffentlicher Schlüssel dient auf Empfänger-Seite zur Überprüfung der Signatur



Verfahren



Ablauf

- ➊ Teilnehmer A (Sender) bildet über Klartext P mit Hash-Alg. H einen Hashwert $V_A = H(P)$
- ➋ Teilnehmer A chiffriert Hashwert V_A mit seinem privaten Schlüssel K_A^- (Vorteil Zeitersparnis)

$$VC_A = E(V_A, K_A^-) (= \text{Signatur})$$

- ➌ Chiffrierter Hashwert wird der (unverschlüsselten) Nachricht angehängt und mit übertragen
- ➍ Teilnehmer B (Empfänger) dechiffriert VC_A mit dem öffentlichen Schlüssel K_A^+ von A

$$V = D(VC_A, K_A^+)$$

- ➎ Neuermittlung des Hashwerts der Nachricht P :

$$V_B = H(P)$$

- ➏ $V = V_B$?
falls ja: Signatur echt und Nachricht unverändert

Schlüsselverwaltung



- Ziel
 - ▶ Sicheres und effizientes Life Cycle Management von Schlüsseln
 - ★ Erzeugen/Einrichten
 - ★ Verteilen
 - ★ Ungültig erklären (Revocation)
 - ▶ Vertrauen in Schlüsselverwaltung notwendig!
- Verschiedene Vorgehensweisen
 - ▶ bei Umgang mit geheimen Schlüsseln (Key Distribution Center, KDC)
 - ▶ bei Umgang mit öffentlichen Schlüsseln (Public Key Infrastructure, PKI)
 - ▶ alles andere als trivial!

Beispiel für KDC-Ansatz: Kerberos



- Grundlage: Umgang mit geheimen Schlüsseln
- Ursprung: MIT
- basiert auf Needham-Schroeder-Authentifizierung
- erweitert um Zeitstempel
 - ▶ Ticket nur in bestimmtem Zeitintervall gültig
- KDC aufgespalten in
 - ▶ Authentication Server (AS)
 - ▶ Ticket Granting Server (TGS)
- Einsatz
 - ▶ Andrew File System
 - ▶ OSF DCE
 - ▶ Microsoft Active Directory

Kerberos (2)



Annahmen:

- Netzwerk unzuverlässig
- Security Server (Rechner für AS und TGS) ist sicher
 - ▶ in sicherem Raum (durch Kerberos bewacht)
 - ▶ kein Eindringling kann Manipulationen vornehmen
 - ⇒ darf geheime Schlüssel aller Benutzer speichern
- Uhren im verteilten System sind „einigermaSSen“ synchron
- Benutzer vergessen Passwörter nicht

Kerberos (3)



Ziele:

- gegenseitige Authentifizierung
- zeitlich befristete Gültigkeit von Schlüsseln, um Schaden zu begrenzen, falls Schlüssel bekannt werden sollte
- Passwörter nie im Klartext auf dem Netzwerk oder auf normalen Servern
- Passwörter auf Client-Rechnern im Klartext nicht länger als einige Mikrosekunden (ständen sonst im Core Dump)

PKI-Systeme



- PKI: Public Key Infrastructure
- Grundlage: Umgang mit öffentlichen Schlüsseln
- Wesentliches Problem
 - ▶ Sichere Verteilung der öffentlichen Schlüssel
 - ▶ Man-in-the-Middle-Attacke beim Schlüsselaustausch möglich
- Basis
 - ▶ Zertifikate
 - ★ Echtheit (Authentizität) öffentlicher Schlüssel
 - ▶ Verzeichnisdienste
 - ★ Auffinden öffentlicher Schlüssel
 - ★ z.B. LDAP

Zertifikate



- Zertifikate
 - ▶ dienen der Bestätigung der Echtheit eines öffentlichen Schlüssels
 - ▶ d.h. der Zugehörigkeit zu einer bestimmten Entität, wie Person, Dienst, ...
- Zertifizierungsstelle (Certification Authority, CA)
 - ▶ bezeichnet ausstellende Instanz
 - ▶ Garant der Zuordnung Schlüssel-Person
 - ▶ Vertrauenswürdigkeit vorausgesetzt oder öffentliche Schlüssel der Zertifizierungsstellen selbst zertifiziert durch übergeordnete CA
 - ▶ von zentraler Instanz (Root CA) überwacht, die die öffentlichen Schlüssel der Zertifizierungsstellen zertifiziert (Vertrauenskette)
- Sperrliste (Certification Revocation List, CRL)
 - ▶ identifiziert Seriennummern ungültig gewordener Zertifikate

X.509-Standard für Zertifikate



- Versionen: v1-v3, umfassend X.509v3
- Wesentliche Informationen in einem Zertifikat:
 - ▶ Version
 - ▶ öffentlicher Schlüssel des Zertifikatinhabers
 - ▶ Name des Inhabers (Distinguished Name)
 - ★ Common Name, CN, (Name der Person)
 - ★ Organization, O, (Firma oder Organisation)
 - ★ Organizational Unit, OU, (Abteilung oder Firmenteil)
 - ★ Locality, L, (Stadt, Sitz der Organisation)
 - ★ State, ST, (Staat, Provinz, Gegend)
 - ★ Country, C, (ISO Ländercode)
 - ▶ Name und Land der ausstellenden CA (Distinguished Name)
 - ▶ Gültigkeitszeitraum
 - ▶ verwendete Algorithmen
 - ▶ Extensions

Beispiel: CAs nach Deutschem Signaturgesetz (SigG)

- SigG seit 1997, akt. Fassung 2001, zuletzt geändert Juli 2009
- Ziel: fortgeschrittene elektronische Signatur basierend auf qualifizierten Zertifikaten (mit geforderten Informationen)
- SigG bestimmt die Regulierungsbehörde für Telekommunikation und Post (Reg TP) als Wurzelinstanz aller CAs (Reg TP umbenannt in Bundesnetzagentur seit 2005).
- Seit 2007 Verwendung längerer Schlüssel RSA 2048, SHA-512

Beispiel: CAs nach Deutschem Signaturgesetz (~~SigG~~)

Hochschule RheinMain

● Zertifizierungsdiensteanbieter (ZDA) Deutschland (Mai 2020):

▶ Derzeit 13 (vgl. <http://www.bundesnetzagentur.de/>)

- ★ 1&1 De-Mail GmbH
- ★ Atos Information Technology GmbH
- ★ Bank-Verlag GmbH
- ★ Bundesagentur fuer Arbeit
- ★ Bundesnetzagentur
- ★ Bundesnotarkammer
- ★ D-Trust (Tochter der Bundesdruckerei)
- ★ DGN Deutsches Gesundheitsnetz Service GmbH
- ★ Deutsche Telekom AG
- ★ T-Systems International GmbH
(1994, erstes Trustcenter)
- ★ exceet Secure Solutions GmbH
- ★ Deutsche Post AG
- ★ medisign GmbH

▶ zahlreiche nicht mehr tätige oder untersagte ZDAs

Protokolle und Anwendungen



- hier nicht betrachtet:
 - ▶ Link-Layer Security
 - ▶ sicherheitsbezogene Protokolle auf der Netzwerkebene, wie IPsec
 - ▶ darauf aufbauende Architekturen wie Virtual Private Networks (VPN)
sichere Verbindung von Teilnetzen über unsichere Netze

Sicherheit in Anwendungsprotokollen



Beispiele

- S/MIME (Secure / Multipurpose Internet Mail Extensions)
 - ▶ RFC 2633
 - ▶ Beispiel für Verschlüsselung auf Anwendungsebene
 - ▶ Standard zur Verschlüsselung und Signatur
 - ▶ Kann mit TLS kombiniert werden
 - ▶ basiert auf X.509-Zertifikaten
 - ▶ Benötigt Unterstützung im Mail-Client
- SSH, SSH2
 - ▶ sicherer entfernter Zugriff zu SSH-Server als Ersatz für telnet, rlogin
 - ▶ verschiedene Formen der Authentifizierung und Verschlüsselung möglich (RSA; DES, 3DES, Blowfish)
 - ▶ Tunneln anderer Anwendungen (z.B ftp, Basis für sftp)
 - ▶ hohe praktische Bedeutung

Transport Layer Security (TLS)



- früher: Secure Socket Layer (SSL)
 - ▶ SSL 3.1 = TLS 1.0
 - ▶ TLS 1.3: RFC 8446 (2018)
- Sicherheit auf **Transportebene**
 - ▶ TLS-Protokoll liegt zwischen der Transportschicht und der Anwendungsschicht
 - ▶ Transparenz für alle höheren Anwendungsprotokolle gegeben (HTTP, SMTP, IIOP, ...)
 - ▶ hohe praktische Bedeutung
- Ursprung
 - ▶ Netscape, Einsatz in Browser als Ersatz für unsichere 40-Bit-Verschlüsselung
- Funktionalität
 - ▶ Authentifizierung und Verschlüsselung
 - ▶ Basis:
 - ★ Public Key-Zertifikate nach X.509
 - ★ symmetrische Verschlüsselung mit geheimen Session-Schlüsseln

TLS (2)



Teilprotokolle

- Handshake-Protokoll:
 - ▶ Server-Authentication
 - ★ Server antwortet auf Client-Request mit Zertifikat und Präferenzen für Verschlüsselungsverfahren (RC4, IDEA, DES, 3DES, ...)
 - ★ Client erzeugt Master Key, verschlüsselt diesen mit öffentl. Schlüssel des Servers aus Zertifikat, sendet verschl. Master Key und gewähltes Verfahren
 - ★ Server ermittelt Master Key und authentifiziert sich gegenüber Client mit Master Key aus verschlüsselter Nachricht
 - ★ anschließend werden Keys benutzt, die aus Master Key abgeleitet sind
 - ▶ Optionale Client-Authentication
 - ★ Server schickt Challenge-Anfrage an Client
 - ★ Client antwortet mit signierter Anfrage und Client-Zertifikat
- Change Cipher Spec Protocol
- Alert Protocol: Fehler-Behandlung
- Application Data Protocol (für Anwendungsdaten)
- Record-Protokoll: Codierung und Transfer (untere Schicht, unmittelbar auf TCP, symm. Verschlüsselung u.a. DES, TripleDES, AES)

Verbreitete Implementierungen: OpenSSL, GnuTLS, LibreSSL, ...

TLS (3)



- TLS benötigt einen zuverlässigen Transportdienst (üblicher TCP)
- für UDP-Kommunikation steht analog Datagram Transport Layer Security (DTLS) zur Verfügung
besitzt hohe Relevanz für IoT-Anwendungen
- **Beispiele**
 - ▶ HTTPS
 - ★ HTTP over TLS, `https://` ...
 - ★ Standard in Browsern
 - ★ Aufbau einer SSL-gesicherten Transportverbindung
 - ★ HTTP nutzt diese Verbindung zur geschützten Übertragung von vertraulichen Daten
 - ★ Port 443 statt 80
 - ▶ SMTPS
 - ▶ IMAPS, POP3S
 - ▶ FTPS

Firewalls



- Ziele
 - ▶ Monitoring aller eingehenden (und ausgehenden) Nachrichten
 - ▶ Eindringlinge verhindern
 - ▶ autorisierten Zugriff erlauben
 - ▶ möglichst geringe Performance-EinbuSSen
- Annahme
 - ▶ Firewall ist selbst sicher und nicht angreifbar
- Klassifikation
 - ▶ nach Ebene, auf der Kontrollen stattfinden:
 - ▶ Router-basiertes Filtern (Packet Filter, Screening)
 - ▶ Application Level Gateways
 - ▶ häufig kombiniert

Packet Filter

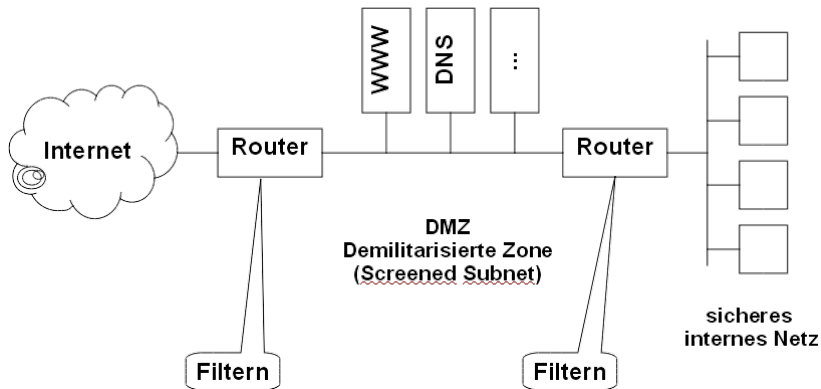


- Netzwerk-bezogen
 - ▶ Betrachtung auf Paketebene
- typischerweise in Routern realisiert
- Regeln für das Nichtweiterleiten
 - ▶ Sperren von Subnetzen
 - ▶ Sperren von Rechnern
 - ▶ Sperren von Diensten
 - ▶ basierend auf IP-Adressen u. Portnummern
- Vorteil
 - ▶ geringer Overhead (=hohe Performance)
- Nachteil
 - ▶ komplexe, nichtmodulare Regeln bei groSSen Netzen
 - ▶ i.d.R. kein Logging
- Beispiel: iptables

Packet Filter (2)



Architekturebene



Application Level Gateway

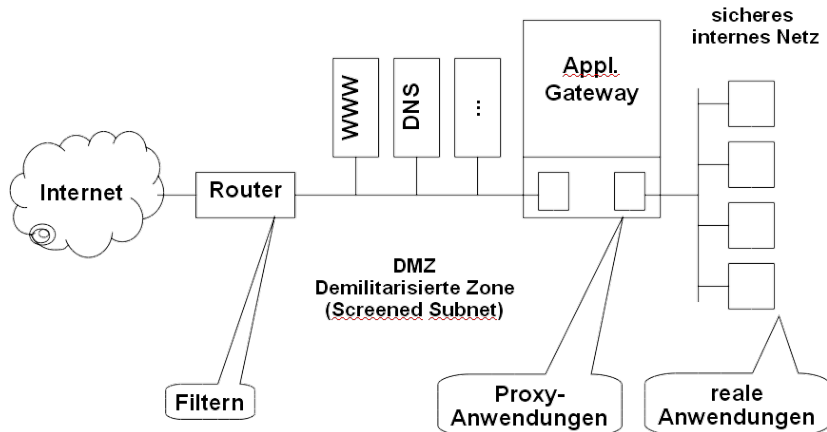


- Sammlung spezieller Proxy-Programme als Ersatz der üblichen Anwendungen
- Üblich für HTTP, FTP, SMTP, X-Protokoll, ...
- Proxy-Anwendungen haben i.d.R.
 - ▶ Zugangsüberprüfung
 - ▶ Logging
- Vorteil: hoher Sicherheitsgrad
- Nachteile:
 - ▶ Notwendigkeit von Proxies
 - ▶ Nachziehen bei neuen Anwendungen
 - ▶ Hoher Performance-Overhead

Application Level Gateway (2)



Architekturebene



Zusammenfassung



- Vollständige Sicherheit (Security) gibt es nicht, es ist immer ein Kompromiss einzugehen.
- Sicherheitsmaßnahmen werden oft durch kryptographische Verfahren umgesetzt.
- Verschlüsselung und Authentifizierung sind zentrale Bestandteile eines jeden Sicherheitskonzeptes.