Verteilte Systeme

Oktober - November 2023

10. Vorlesung – 16.11.2023

Kurs: TINF21AI1

Dozent: Tobias Schmitt, M.Eng.

Kontakt: d228143@student.dhbw-mannheim.de

Wiederholungsfragen

- Was bedeutet zuverlässiges Multicasting? Nennen Sie einen möglichen Ansatz?
- Was ist atomares Multicasting?
- Wie funktioniert der 2-Phasen-Commit?
- Welche Ansätze zur Wiederherstellung kennen Sie?
- Nennen Sie Ansätze zur Erstellung von Kontrollpunkten in verteilten Systemen?
  Mit welchen Problemen, Hindernissen bzw. Schwierigkeiten sehen Sie sich konfrontiert?

Wiederholungsfragen

- Welche Schutzziele können / sollten Sie in einem verteilten System anstreben?

- Welchen Sicherheitsbedrohungen sind Sie in verteilten Systemen ausgesetzt?

- Welche Aspekte hinsichtlich Sicherheitsmechanismen kennen Sie?

- Welche Aspekte sollten Sie bei einem Entwurf eines sicheren Systems bedenken?

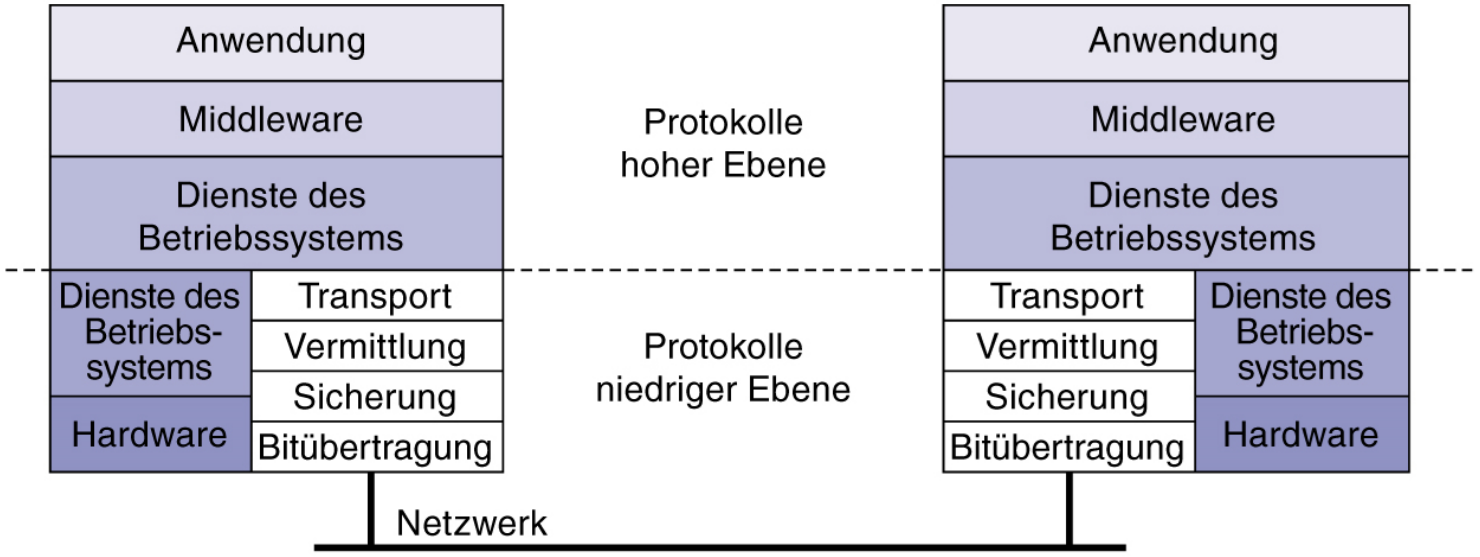
Themenüberblick

- **Sicherheit**
  - **Grundlagen und Entwurfsfragen (Teil 2)**
  - Kryptografie
  - Sichere Kanäle
  - Zugriffssteuerung und Sicherheitsverwaltung
- Wiederholung

Sicherheit - Entwurfsfragen

.Schichten der Sicherheitsmechanismen

- Frage: Auf welcher Ebene müssen Sicherheitsmechanismen platziert werden?
- Hinweis bzgl. verteilter Systeme: Sicherheitsmechanismen meist auf der Schicht der Middleware
- Problematik des Vertrauens in die Sicherheit zugrundeliegender Schichten



Sicherheit - Entwurfsfragen

.Verteilung von Sicherheitsmechanismen

- Bestehen von Abhängigkeiten hinsichtlich des Vertrauens → Trusted Computing Base (TCB)

- TCB = Menge aller Sicherheitsmechanismen in einem (verteilten) Computersystem, die zur Durchsetzung von Sicherheitsrichtlinien benötigt werden … und denen deshalb vertraut werden muss.

- Verteiltes System → Sicherheit beruht auf dem zugrunde liegenden lokalen Betriebssystem.
  - Verwendung von VMs → Vertrauen beruht auf dem verwendeten Hypervisor (vgl. https://www.theregister.com/2020/03/17/virtual_machines_patch/)

- Konsequenz: Trennung von Sicherheitssystemen von anderen Diensten durch Nutzung verschiedener Computer mit entsprechendem Grad an Sicherheit
  - z.B. Dateiserver auf Rechner mit vertrauenswürdigem Betriebssystem und Ausführung der Clients auf „unsicheren“ Computern

Sicherheit - Entwurfsfragen

.Einfachheit

- „Einfachheit trägt zum Vertrauen bei, dass Endbenutzer in die Anwendung setzen, und, was wichtiger ist, zu der Überzeugung der Entwickler, dass das System keine Sicherheitslücken hat.“
- Prinzip: „keep it simple“ → Zusätzliche Features können mehr Angriffspunkte liefern.
  - z.B. E-Mails in Plain-Text sicherer
  - z.B. E-Mail-Programm mit automatischen Öffnen der Anhänge unsicherer
- Verteiltes System → mitunter hohe Komplexität
  - Anwendung von relativ einfachen und leicht verständlichen Mechanismen
- Beispiel: Arbeit mit Mircoservices
  - Unabhängigen Prozessen kommunizieren über API / Programmierschnittstelle
  - Microservice sollte von jedem Teammitglied überschaubar sein und in vertretbarem Zeitaufwand erstellt werden
  - Siehe https://de.wikipedia.org/wiki/Microservices

Themenüberblick

- **Sicherheit**
  - Grundlagen und Entwurfsfragen (Teil 2)
  - **Kryptografie**
  - Sichere Kanäle
  - Zugriffssteuerung und Sicherheitsverwaltung
- Wiederholung

Kryptographie - Einstiegsfragen

- Kinder überlegen sich für jeden Buchstaben ein neues Zeichen. Ist die Verschlüsselung sicher?

- Daten werden in einem binären Format in einer Datei abgelegt. Vorausgesetzt, dass das Datenformat nicht veröffentlicht wurde, sind dann die Daten sicher?

- Eine Softwareschmiede bietet ein High-End-Verschlüsselungsprodukt an. Aus Gründen des Marktvorteils will die Softwareschmiede den Algorithmus nicht veröffentlichen. Wie sicher sind dann Daten, die mit der Software verschlüsselt wurden?

- Ab wann gilt ein Verschlüsselungsalgorithmus als gebrochen?

- Worin liegt die Effizienz eines Kyptographie-Verfahrens?

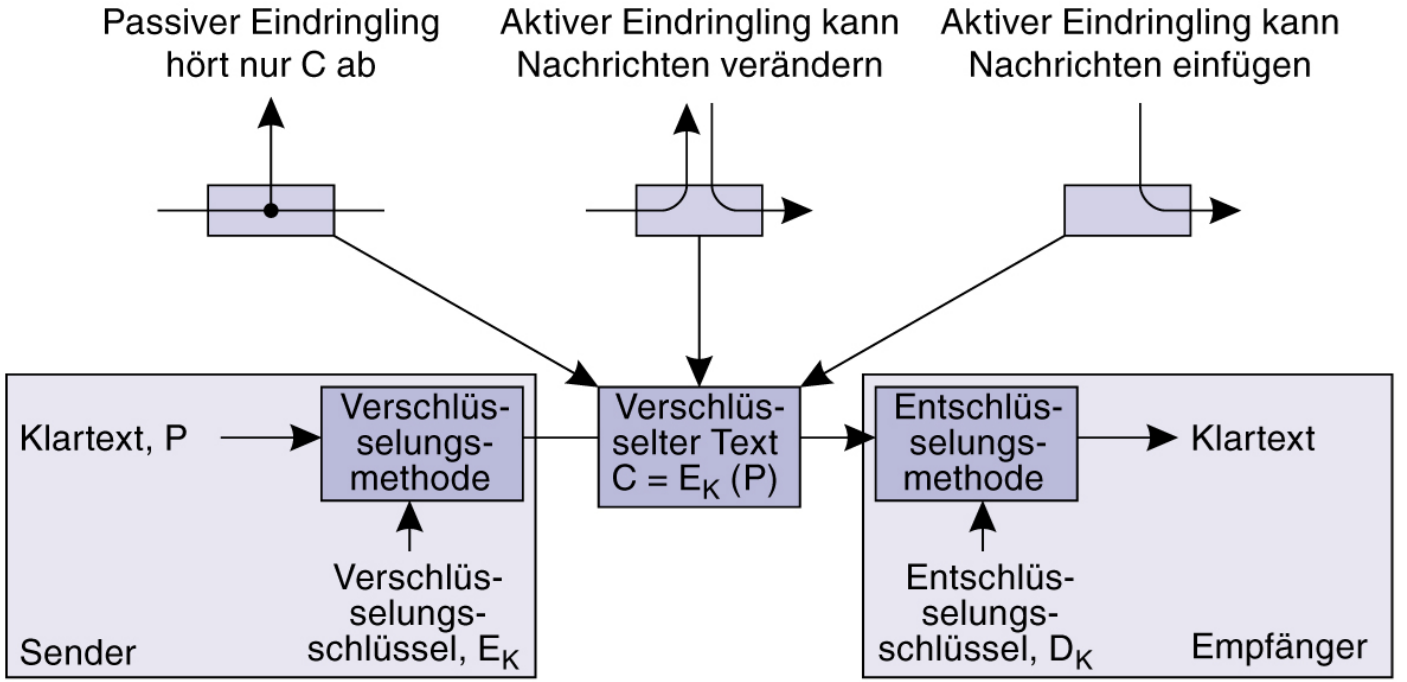
Kryptographie - Grundlagen

- Grundprinzip der Verschlüsselung
und Arten von Sicherheitsangriffen

- Klartext P

→ Verschlüsselter Text $C = E_K(P)$

→ Entschlüsselter Text $D_K(C)=D_K(E_K(P)) = P$



Kryptographie - Grundlagen

.Symmetrische Verschlüsselung

- Gleicher Schlüssel für die Ver- und Entschlüsselung (Sender und Empfänger haben gleichen Schlüssel)
- Begrifflichkeit: Secret-Key-Systeme, Shared-Key-Systeme
- Effizienz abhängig von Schlüssellänge – Bsp.: AES (Advanced Encryption Standard)

.Asymmetrische Verschlüsselung

- Schlüsselpaar – öffentlicher Schlüssel und privater Schlüssel
- Abhängig von Verwendung, welcher Schlüssel zu Ver- oder Entschlüsselung verwendet wird
  - Bsp.: Senden von Nachrichten: Verschlüsselung mit dem öffentlichen Schlüssel des Empfängers
  - Bsp.: Signaturen – Hashwert der Nachricht wird mit eigenem privaten Schlüssel verschlüsselt

Schreibweise	Erklärung
$K_{A,B}$	Geheimer Schlüssel, den A und B gemeinsam nutzen
K_A^+	Öffentlicher Schlüssel von A
K_A^-	Privater Schlüssel von A

Kryptographie - Grundlagen

.Hash-Funktionen

- Hash-Funktion *H* erzeugt aus einer Nachricht *m* von beliebiger Länge eine Bit-Folge / Hash-Wert *h* mit fester Länge: *h = H(m)*
- Eigenschaften
  - Einwegefunktionen → Bestimmung von *m* aus *h* durch Berechnung nicht möglich
  - Schwache Kollisionsresistenz → Keine Berechnung *m‘* aus *m* möglich, so dass *H(m‘) = H(m)*
  - Starke Kollisionsresistenz → Aus Kenntnis von *H* ist es durch Berechnung nicht möglich zwei unterschiedliche Eingaben *m* und *m‘* zu finden, so dass *H(m‘) = H(m)*
- Beispiel: md5
  - „Das ist ein Test.“ → e2f8af8ce3fa850a1591100a57f12222
  - „Das ist ein Dest.“ → 58d5dd4a294adef74cff290e4096da12
  - Achtung: Hash-Kollisions bekannt
  - Unsicher aufgrund Existenz einer „Rainbow Table“
    (z.B. 1-7 Zeichen und Erfolgschange von 99,9% → Tabelle umfasst 52 GB)

Interludium: RSA-Verschlüsselung

- Benannt nach 3 Mathematikern: Rivest, Shamir, Adleman
- Vorausberechnung

1) Wähle zufällig und stochastisch unabhängig 2 Primzahlen

$$p \neq q$$

2) Berechne RSA-Modul:

$$N = p * q$$

3) Berechne Eulersche Funktion:
(Anzahl der zu N teilerfremden ganzen Zahlen kleiner N)

$$\varphi(N) = (p - 1)(q - 1)$$

4) Wähle teilerfremde Zahl e zu $\varphi(N)$ mit

$$1 < e < \varphi(N)$$

5) Berechne Entschlüsselungsexponent d mit

$$(e*d) \bmod \varphi(N) = 1$$

Interludium: RSA-Verschlüsselung

- Verschlüsselung mit (e,N)
  - Unverschlüsselte Nachricht m
  - Verschlüsselung durch: $c = m^e \ mod \ N$

- Entschlüsselung mit (d,N)
  - Verschlüsselte Nachricht c
  - Entschlüsselung durch: $m = c^d \ mod \ N$

Interludium: RSA-Verschlüsselung

- Beispiel:
  - Vorausberechnungen
    1) $p = 7, q = 11$
    2) $N = p * q \rightarrow N = 77$
    3) $\varphi(N) = (p - 1)(q - 1) \rightarrow \varphi(N) = 60$
    4) $1 < e < \varphi(N)$ … $e = 13$ (ausgewählt)
    5) $(e*d) \bmod \varphi(N) = 1 \rightarrow d = 37$,
       da $(13*37) \bmod 60 = 481 \bmod 60 = 1$
  - Verschlüsselung: $m=2 \rightarrow c = 2^{13} \bmod 77 = 30$
  - Entschlüsselung: $c = 30 \rightarrow m = 30^{37} \bmod 77 = 2$
  - Hinweis: $(x^a)^b \bmod N = (x^a \bmod N)^b \bmod N$
  - z.B. $30^3 \bmod 77 = ((30*30) \bmod 77) * 30 \bmod 77 = (53 * 30) \bmod 77 = 50$
    $30^4 \bmod 77 = (30^3 \bmod 77) * 30 \bmod 77 = (50 * 30) \bmod 77 = 37$
    ….

Themenüberblick

- **Sicherheit**
  - Grundlagen und Entwurfsfragen (Teil 2)
  - Kryptografie
  - **Sichere Kanäle**
  - Zugriffssteuerung und Sicherheitsverwaltung
- Wiederholung

Sichere Kanäle

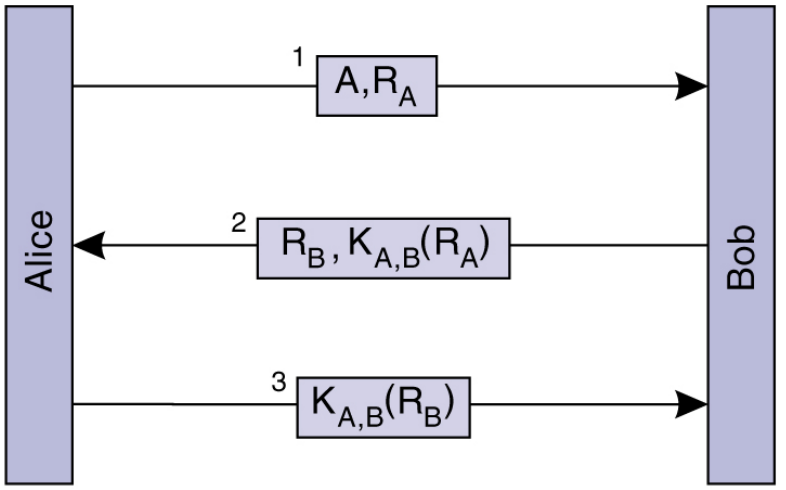
- Sicherer Kanal
  - Schutz vor Abfangen, Änderung und Fälschung der Nachrichten

- Betrachtungsaspekte
  - Authentifizierung
    - Identifikation der beteiligten Parteien
  - Nachrichtenintegrität
    - Sicherstellung, dass keine Nachrichten verändert werden

- Frage zur Authentifizierung
  - Alice und Bob besitzen geheimen Schlüssel $K_{A,B}$
  - Wie können sich beide damit gegenseitig authentifizieren?

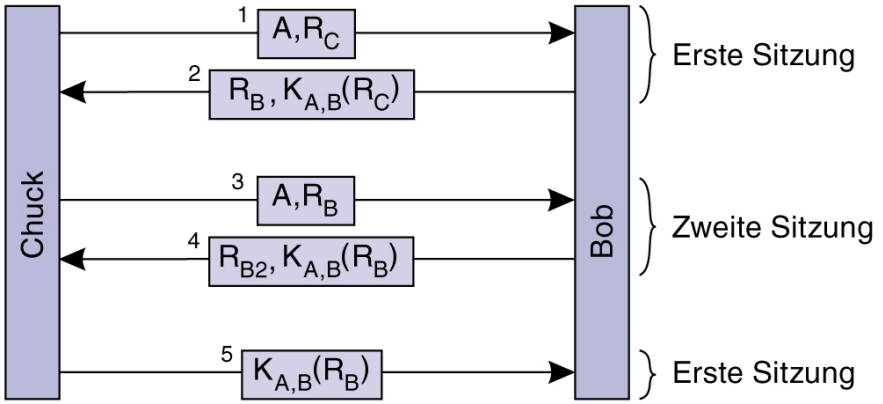
Authentifizierung

- Authentifizierung auf Grundlage eines geheimen Schlüssels
  - Ansatz:
    - Alice und Bob besitzen den geheimen Schlüssel $K_{A,B}$
    - Alice sendet Anfrage A an Bob und bitten R_A zu verschlüsseln, um den Besitz des Schlüssels zu bestätigen
    - Bob verschlüsselt R_A und sendet selber eine Aufgabe R_B an Alice
    - Alice soll nun ihrerseits R_B verschlüsseln und an Bob senden, damit Sie auch den Besitz des geheimen Schlüssels bestätigt
    - Ablaufplan:



Authentifizierung

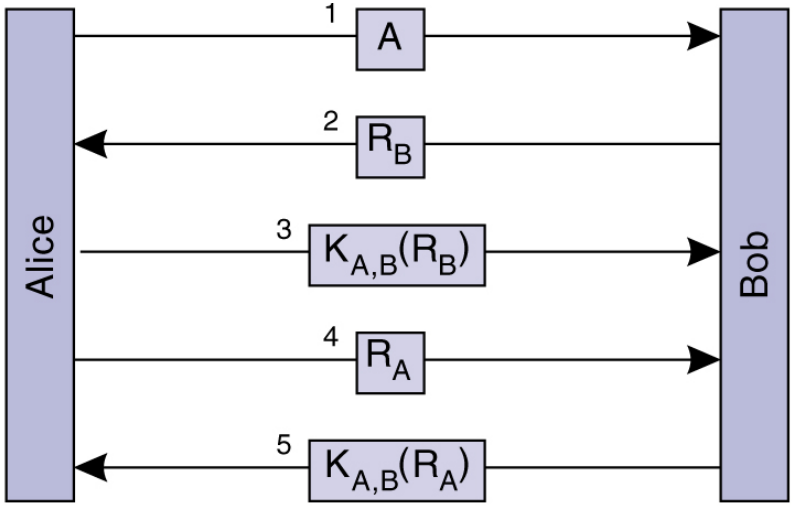
- Authentifizierung auf Grundlage eines geheimen Schlüssels
  - Problem des Ansatzes – **Reflektionsangriff** ist möglich!
    - Angreifer nimmt Anfrage R_B und bittet durch eine erneute Verbindungsanfrage die Verschlüsselung von R_B vorzunehmen



  - Verbesserung
    - Jede Aufgabe darf nur einmal gestellt werden.
    - Aber Möglichkeit für Man-in-the-Middle-Angriff

Authentifizierung

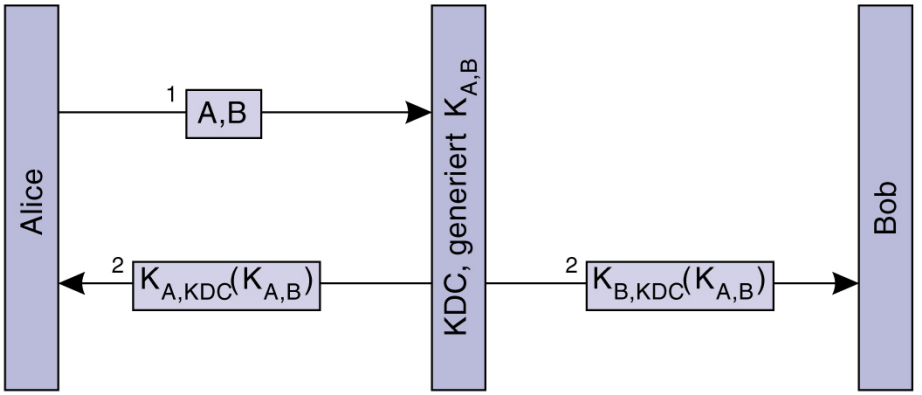
- Authentifizierung auf Grundlage eines geheimen Schlüssels
  - Fazit aus dem ersten Ansatz: Der Anfragende darf den Anzufragenden nicht als erstes fragen sich zu authentifizieren.
  - Tatsächliche Realisierung:
    - Alice sendet Anfrage an Bob
    - Bob sendet Aufgabe R_B an Alice zur Verschlüsselung
    - ...



Authentifizierung

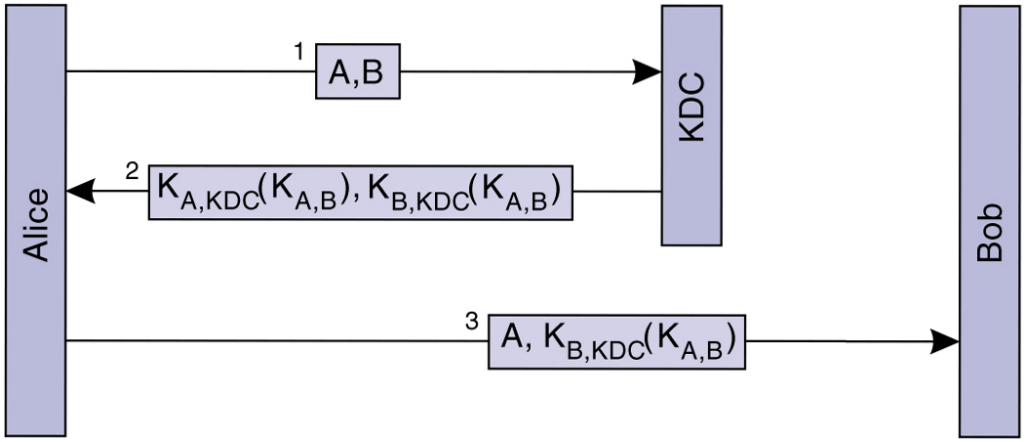
- Problem bzgl. Skalierbarkeit
  - N Hosts → Pflege von (N – 1)-Schlüsseln für Kommunikation mit anderen Hosts
  - Insgesamt N (N – 1) / 2 Schlüssel im System

- Authentifizierung über ein KDC (Key Distribution Center)
  - Verwahrung von N Schlüsseln (zu jedem Host)
  - Ansatz:
    - Alice stellt Anfrage an KDC für Kommunikation mit Bob
    - KDC sendet Schlüssel an Alice und Bob
    - Frage: Was ist, wenn Alice dann doch keinen Kanal zu Alice öffnet? Dann wurde Bob umsonst geweckt.



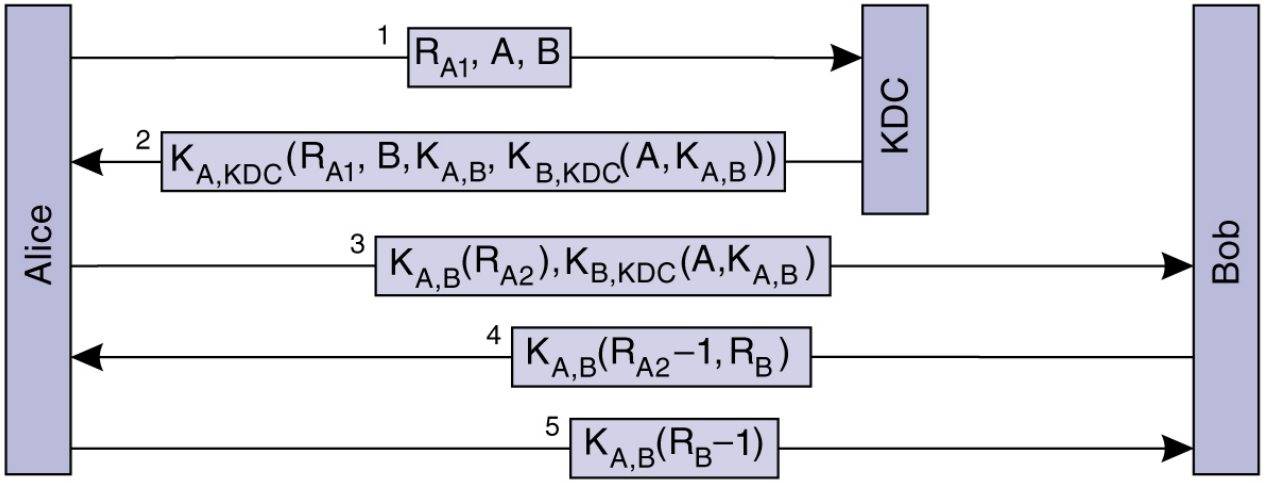
Authentifizierung

- Authentifizierung über ein KDC
  - Abänderung:
    - Alice fragt beim KDC an
    - Alice erhält 2 Infos
      - Selbst entschlüsselbaren Schlüssel $K_{A,B}$
      - Und Ticket: nur von Bob entschlüsselbaren Schlüssel $K_{A,B}$

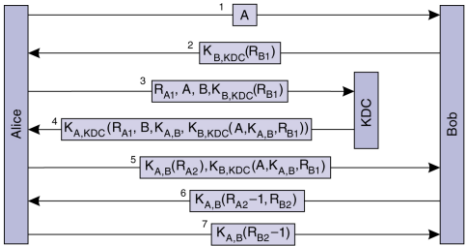


Authentifizierung

- Authentifizierung über ein KDC
  - Needham-Schroeder-Authentifizierungsprotokoll
    - Alice sendet Anfrage an KDC für Kommunikation mit Bob und eine **Nonce**
    - Nonce
      - Zufallszahl, die nur einmal verwendet wird
      - Zweck: 2 Nachrichten in Beziehung zu setzen
    - Schwäche: Angreifer kann Nachricht 3 erneut einspielen

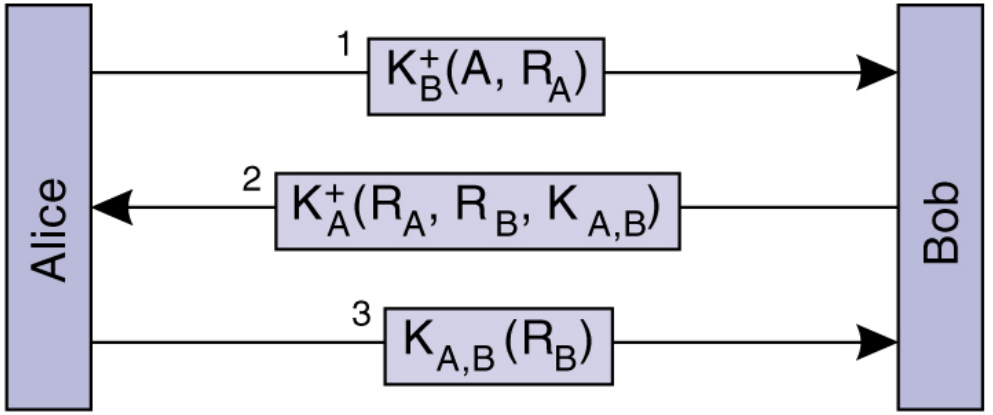


Behebung der Schwäche:



Authentifizierung

- Authentifizierung mit öffentlichen Schlüsseln
  - Alice nutzt öffentlichen Schlüssel von Bob für
    - Anfrage und Nonce
  - Bob nutzt öffentlichen Schlüssel von Alice für
    - Schlüsse, alte Nonce, neue Nonce
  - Rückmeldung von Alice mit gemeinsamen Schlüssel verschlüsselter Nonce



Nachrichtenintegrität und Vertraulichkeit

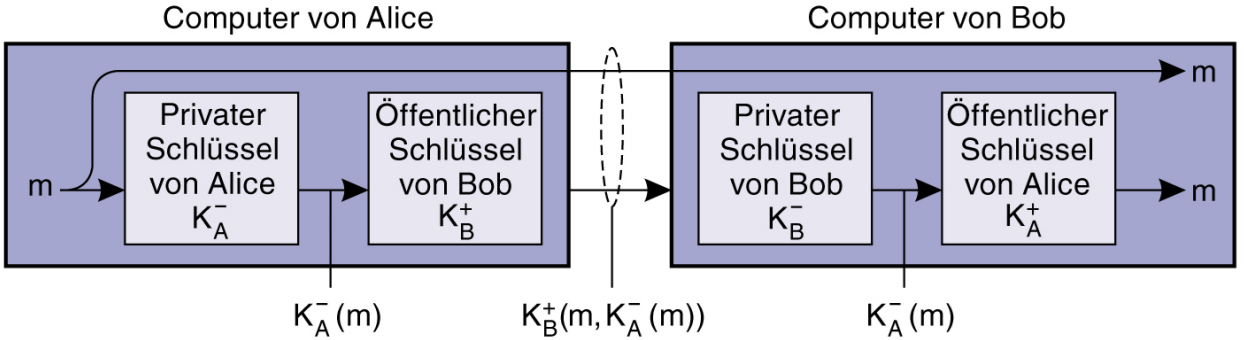
- Vertraulichkeit
  - Schutz vor Abfangen und Lesen
  - Realisierung durch Verschlüsselung

- Nachrichtenintegrität
  - Schutz vor (betrügerischen) Veränderungen
  - Verwendung von
    - Digitalen Signaturen
    - Sitzungsschlüsseln

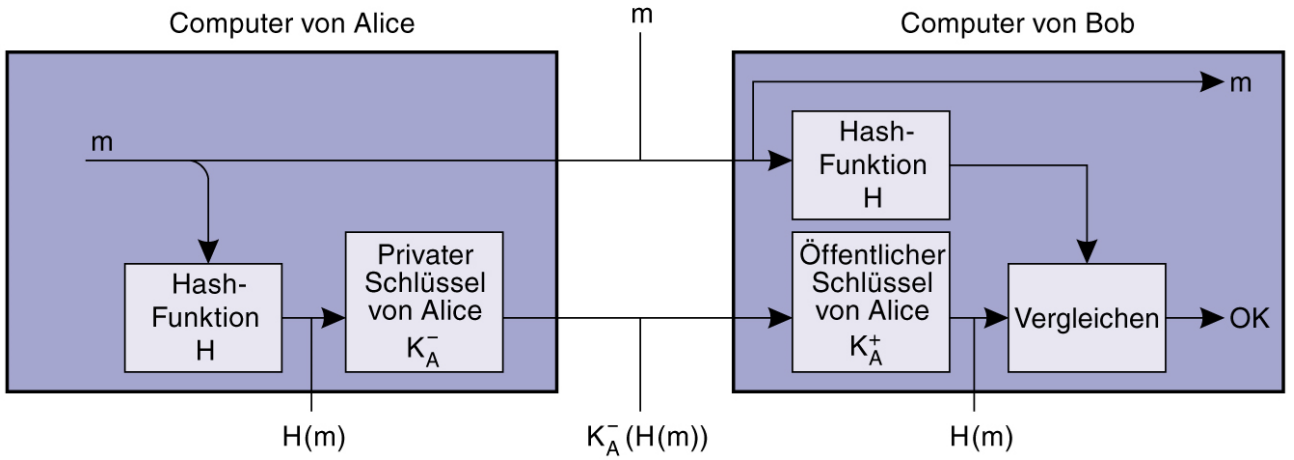
- Sitzungsschlüssel
  - Anmerkung: Aufdeckung eines Schlüssels leichter, wenn er häufig verwendet wird
  - Schutz vor Wiedereinspielungen alter Nachrichten
    (normalerweise Nutzung von Zeitstempeln oder Nummerierung)

Nachrichtenintegrität

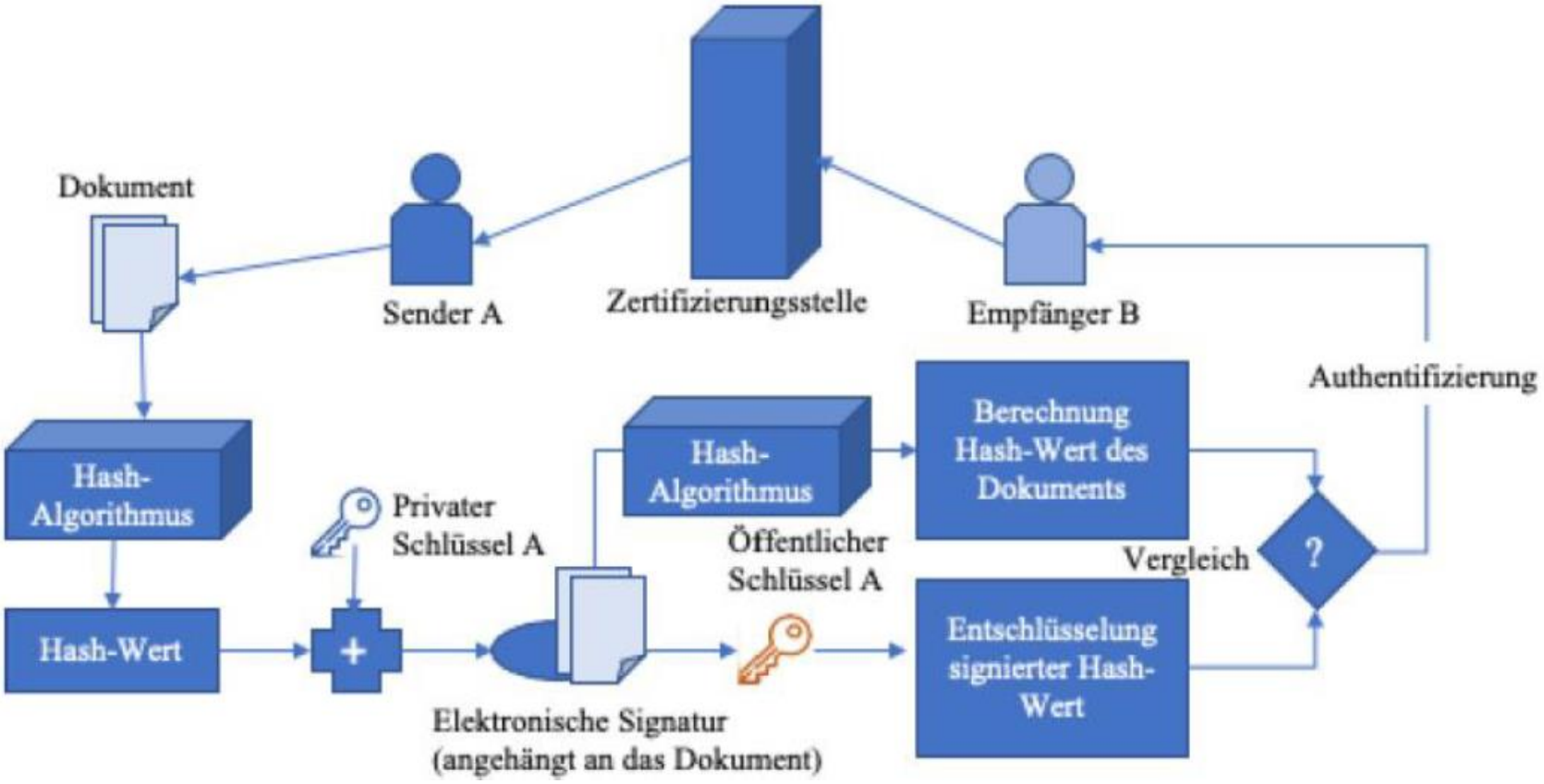
- Digitale Signaturen
  - Variante 1



  - Variante 2



Nachrichtenintegrität



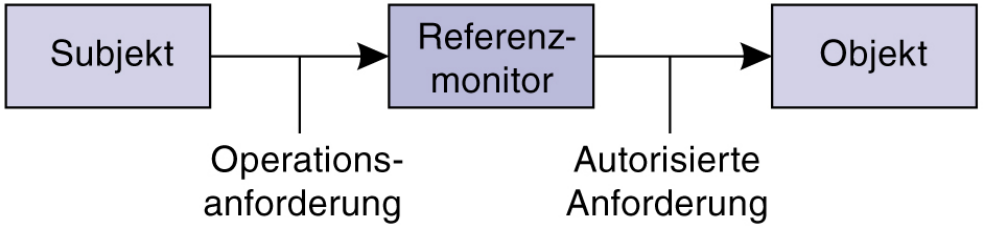
Themenüberblick

- **Sicherheit**
  - Grundlagen und Entwurfsfragen (Teil 2)
  - Kryptografie
  - Sichere Kanäle
  - **Zugriffssteuerung und Sicherheitsverwaltung**
- Wiederholung

Zugriffssteuerung

- Überprüfen der Zugriffsrechte = Zugriffssteuerung
- Autorisierung → Gewährung von Zugriffsrechten

- Referenzmonitor
  - Allgemeines Modell der Kontrolle des Zugriffs auf ein Objekt
    - Was darf welches Subjekt tun?
    - Entscheidung, was erlaubt ist.
    - Aufruf, wenn Zugriff auf Objekt.
    - Achtung: Sicherung des Referenzmonitors gegen Eingriffe



Zugriffssteuerung

- Zugriffssteuerungsmatrix (Access Control Matrix)
  - Jedes Subjekt eine Zeile, jedes Objekt eine Spalte
  - Ergebnis: Sparse Matrix → Ineffizient
  - Alternativen
    - Zugriffssteuerungsliste → Objekt erhält Liste von Subjekten
    - Liste der Fähigkeiten → Subjekt erhält Liste von Objekten
- Schutzdomänen (Protection Domains)
  - Schutzdomäne = Menge von Paaren aus Objekt + Zugriffsrechten
  - Bildung von Benutzergruppen, ggf. hierarchische Gruppen
  - Realisierung auch durch Rollen
  - Alternative zur Arbeit des Referenzmonitors: Benutzer erhält Zertifikat mit Gruppenzugehörigkeiten

Zugriffssteuerung

- Firewalls
  - Paketfilterungs-Gateway
  - Gateway auf Anwendungsschicht (z.B. Spamfilter)

Sicherheitsverwaltung

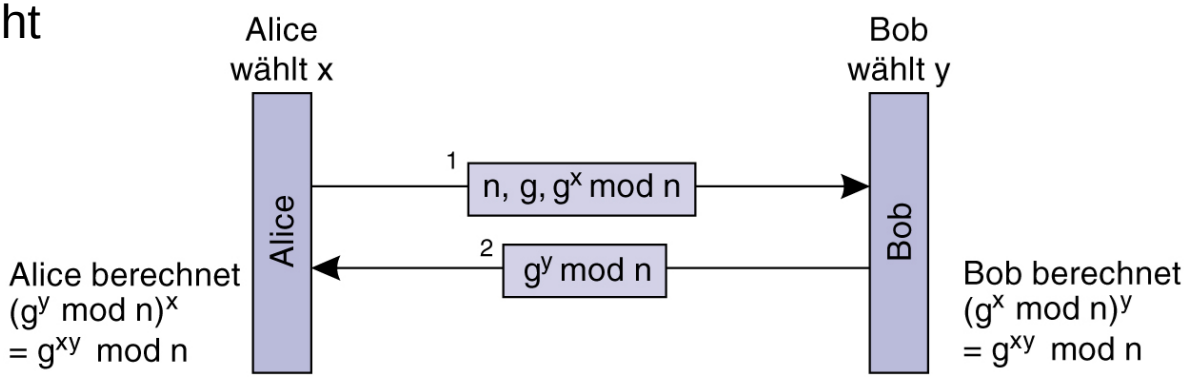
- Aspekte der Schlüsselverwaltung
  - Schlüsseleinrichtung
    - Verwendung des Diffie-Hellman-Schlüsselaustausches
  - Schlüsselverteilung
    - Verteilung von öffentlichen Schlüsseln mittels Zertifikat
    - Zertifikat
      - Bezeichnung der Entität zu dem öffentlichen Schlüssel
      - Signatur einer Zertifizierungsstelle (Certification Authority)
      - Signierung mittels privaten Schlüssel der Zertifizierungsstelle

Sicherheitsverwaltung

- Aspekte der Schlüsselverwaltung
  - Gültigkeitsdauer von Zertifikaten
    - Problematik hinsichtlich Kompromittierung eines privaten Schlüssels
    - Bei unbegrenzter Gültigkeitsdauer → Einrichtung von Zertifikatssperrlisten
    - Alternative: Begrenzung der Gültigkeitsdauer → vgl. Leases
    - … aber Notwendigkeit des regelmäßigen Prüfens der Sperrlisten

Sicherheitsverwaltung

- Diffie-Hellman-Schlüsselaustausch
  - Festlegung einer Zahl g und einer großen Primzahl n
  - Alice wählt zufällige Zahl x und berechnet $g^x \bmod n = a$
  - Bob wählt zufällige Zahl y und berechnet $g^y \bmod n = b$
  - Austausch der Berechnungen und:
    - Alice: $(g^y \bmod n)^x \bmod n = g^{xy} \bmod n$
    - Bob: $(g^x \bmod n)^y \bmod n = g^{xy} \bmod n$
  - Alice und Bob haben den gleichen Schlüssel, etwaige Lauscher aber nicht



(Vgl. auch https://www.youtube.com/watch?v=iHDNpH_Xw4Y)

Themenüberblick

- Sicherheit
  - Grundlagen und Entwurfsfragen (Teil 2)
  - Kryptografie
  - Sichere Kanäle
  - Zugriffssteuerung und Sicherheitsverwaltung
- **Wiederholung**

Wiederholung

Themen der Vorlesung

- Architekturen
- Prozesse
- Kommunikation
- Benennung und Namenssysteme
- Synchronisierung
- Konsistenz und Replikation
- Fehlertoleranz
- Sicherheit

Wiederholung - Grundlagen

- Definition eines verteilten Systems
  - Ein verteiltes System ist eine Ansammlung *unabhängiger Computer*, die den Benutzern wie *ein einzelnes kohärentes System* erscheinen.
- Beispiele für verteilte Systeme
- Gründe für den Einsatz von verteilten Systemen
- Ziele eines verteilten Systems
  - Ressoucenzugriff
  - Verteilungstransparenz
  - Offenheit
  - Skalierbarkeit

Transparenz	Beschreibung
Zugriff	Verbirgt Unterschiede in der Datendarstellung und die Art und Weise, wie auf eine Ressource zugegriffen wird
Ort[1]	Verbirgt, wo sich eine Ressource befindet
Migration	Verbirgt, dass eine Ressource an einen anderen Ort verschoben werden kann
Relokation	Verbirgt, dass eine Ressource an einen anderen Ort verschoben werden kann, während sie genutzt wird
Replikation	Verbirgt, dass eine Ressource repliziert ist
Nebenläufigkeit	Verbirgt, dass eine Ressource von mehreren konkurrierenden Benutzern gleichzeitig genutzt werden kann
Fehler	Verbirgt den Ausfall und die Wiederherstellung einer Ressource

Wiederholung - Architektur

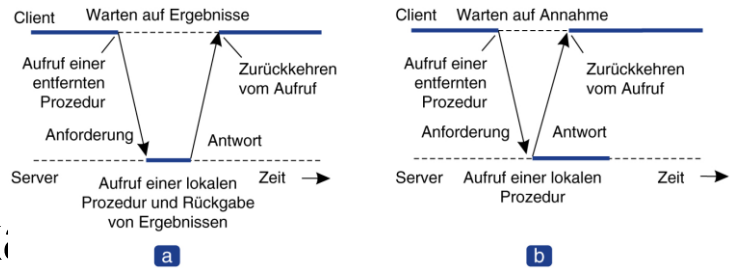
- Zentralisierte Strukturen
  - Client-Server
    - Zustandslose und zustandsbehaftete Server
  - Zwei-Tier-Architektur
    - Verteilung der 3 Ebenen (Benutzerschnittstelle, Verarbeitungsebene, Datenebene)
    - Thin-Clients, Thick-Clients
  - Drei-Tier-Architektur
- Dezentralisierte Strukturen
  - Peer-to-Peer-Systeme (strukturiert, unstrukturiert)

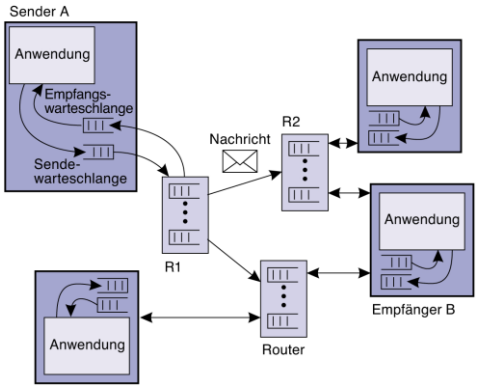
Wiederholung - Prozesse

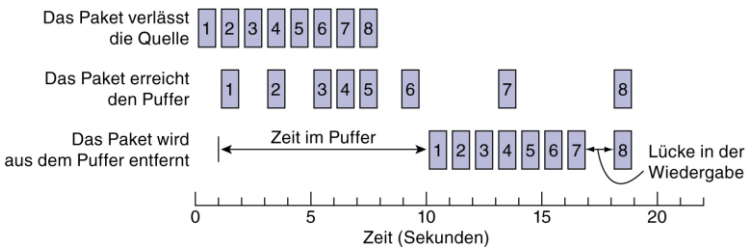
- Threading
  - Multithread-Clients, Multithread-Server
- Virtualisierung
  - Bedeutung, Ansätze, ...
- Codemirgration / Prozessmigration
  - „schwache“ und „starke“ Mobilität
  - Sender- und empfängerinititiierte Migration
  - Umgang mit Ressourcen

Wiederholung - Kommunikation

- OSI-Modell und Middleware-Schicht
- Betrachtungsaspekte allgemein
  - Persistente / Flüchtige Kommunika
  - Synchrone / Asynchrone Kommunikation
  - Diskrete / Fließende Kommunikation
- Betrachtungsaspekte speziell
  - Entfernte Prozeduraufrufe (RPC)
    - Konzept, Arbeitsweisen
    - Vgl. Synchrone / Asynchrone Kommunikation







Wiederholung - Kommunikation

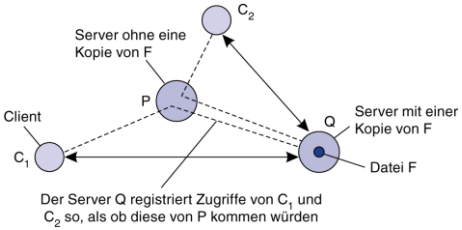
- Betrachtungsaspekte speziell
  - Nachrichtenorientierte Kommunikation
    - Vgl. Persistente / Flüchtige Kommunikation
    - Sockets, Warteschlangen
  - Streamorientierte Kommunikation
    - Vgl. Diskrete / Fließende Kommunikation
    - Dienstgüte, Synchronisierung
  - Multicast-Kommunikation
    - Overlay-Konstruktion, Multicast-Baum, Gossip

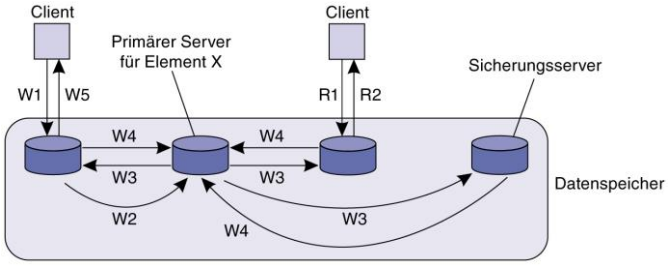
Wiederholung

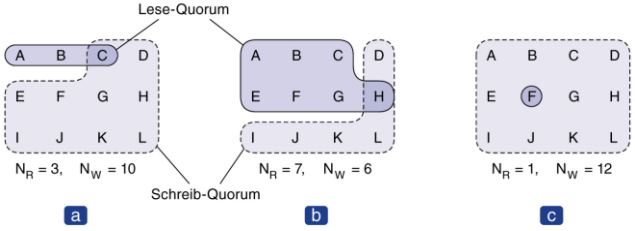
- Benennung und Namenssysteme
  - Lineare Benennung, Hierarchische Benennung
  - DNS-Nameserver
    - Iterative und rekursive Namensauflösung
- Synchronisierung
  - Network Time Protocol (NTP), Berkley-Algorithmus
  - Logische Uhren (z.B. Logische Uhr von Lamport)
    - Vollständig geordnetes Multicasting
  - Gegenseitiger Ausschluss
    - Zugriff auf Ressource durch nur einen Prozess → diverse Algorithmen
  - Wahlalgorithmen
    - Ziel ist Bestimmung eines neuen Koordinators

Wiederholung
Konsistenz und Replikation

- Replikationsverwaltung
  - Permanente Replikation, Serverinitiierte Replikati
    Clientinitiierte Replikation
  - Verteilen von Inhalten
    - Push-basierter Ansatz, Pull-basierter Ansatz, Leases
- Konsistenzprotokolle
  - Urbildbasierte Protokolle, aktive Replikation, Protokoll für replizierte Schreibvorgänge

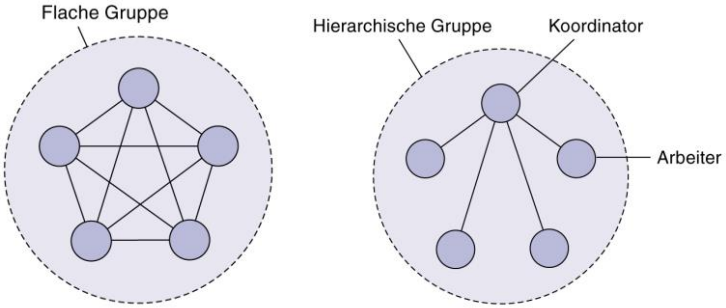


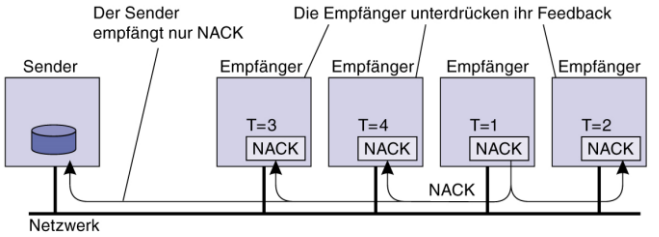


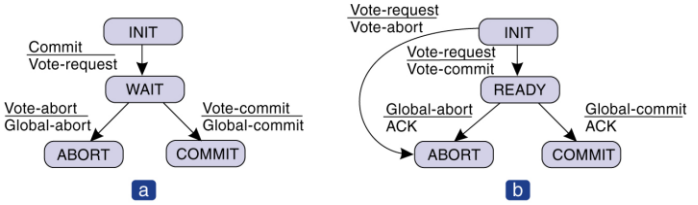


Wiederholung - Fehlertoleranz

- Grundbegriffe
  - Ausfallarten
- Prozess-Resilienz
  - Lineare Gruppe, hierarchische Gruppe
  - Gruppenverwaltung, Replikation, Einigungsalgorithmen
- Client-Server-Kommunikation
  - Fehler bei RPC-Semantik
- Gruppenkommunikation
  - Zuverlässiges Multicasting, atomares Multicasting
- Verteilter Commit
  - Zwei-Phasen-Commit, Drei-Phasen-Commit
- Wiederherstellung
  - Rückwärtswiederherstellung, Kontrollpunkte







Wiederholung - Sicherheit

- Sicherheitsbedrohungen
- Sicherheitsmechanismen
  - Verschlüsselung, Authentifizierung, Autorisierung, Accounting (Kontrolle)
  - Beispiele für spezifische Sicherheitsmechanismen
- Entwurfsaspekte
- Sichere Kanäle
  - Authentifizierung aufgrund geheimer oder öffentlicher Schlüssel
  - Nachrichtenintegrität (Digitale Signaturen)
- Zugriffssteuerung und Sicherheitsverwaltung
  - Referenzmonitor
  - Schlüsseleinrichtung, Schlüsselverteilung, Zertifikate