Verteilte Systeme

17. Schutz

Motivation

Schutz (Protection)

- Zugang zu den Ressourcen eines Rechners tatsächlich kontrollieren
  - Wer darf welche Ressourcen wie nutzen?
  - Dateien
  - Seiten im Adreßraum
  - Ein- und Ausgabegeräte
  - Semaphore
  - ...
- Authentisierung
- Schutz umsetzen
- Form der Beschreibung

Sicherheit (Security)

- Schutz vor unautorisiertem Zugriff auf Ressourcen
  - Systemeigenschaft
  - Wie weit kann ein Benutzer dem System vertrauen?
- Benutzer authentifizieren
- Große Bedeutung durch die Vernetzung von Rechnern
  - Eingehende Nachrichten können von jedem kommen

17.1 Schutz

Schutz

- Umfang
  - Anwendungen stehen bestimmte Ressourcen zur Verfügung
  - Jede Ressource besitzt Namen und Typ
  - Abhängig vom Typ sind bestimmte Operationen erlaubt
- Schutz vor unzulässigem Zugriff
- Erhöhte Zuverlässigkeit
  - Schutz vor unbewußter falscher Nutzung von Ressourcen
  - Aufdecken versteckter Fehler durch Zugriffsverletzung
- Need-to-know-Prinzip: Nur die Ressourcen verwenden, die zur Erfüllung der Aufgabe unabdingbar sind
- Schutzpolitik (Policy)
  - Administration
- Schutzmechanismen (Umsetzung)

Schutzdomänen

Domäne 1

<O1, {Read, Write}>
<O2, {Execute}>
<O3, {Print}>

Domäne 2

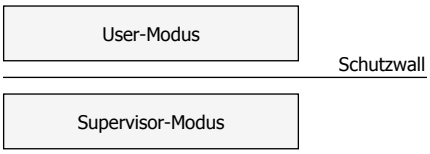
<O4, {Read}>
<O5, {Write}>
<O3, {Execute,Write}>

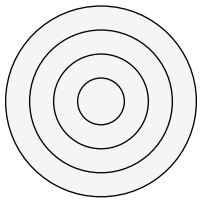
- Grundstruktur
  - Zu schützende Objekte
    - Name
    - Typ
    - Definierte Operationen
  - Kontrollflüsse (Subjekte), deren Objektzugriffe überwacht werden müssen
- Kontrollflüsse arbeiten innerhalb bestimmter Schutzdomäne
- Schutzdomäne definiert Zugriffsrechte eines Kontrollflusses auf die Objekte
- Schutzdomäne = Menge von Zugriffsrechten <Objekt, Rechte>
- Zuordnung Domäne-Kontrollfluß
  - Statisch (Inhalte ändern)
  - Dynamisch (Domäne wechseln)

Verteilte Systeme, Sommersemester 1999 Folie 17.5

Schutzmechanismen

- Supervisor/User-Modus
  - Unterscheidung zweier Modi
  - Wohldefinierte Übergänge
    - Interrupts
    - Traps
  - Problem: Verschiedene Prozesse auch untereinander schützen
- Schutzringe
  - Verallgemeinerung
  - z.B. Intel-Prozessor





Verteilte Systeme, Sommersemester 1999 Folie 17.6

Schutzdomänen sind auch Objekte

- Beschreibung dynamischer Aspekte
  - Wechsel der Schutzdomäne (Switch)
  - Übertragung, Hinzufügen und Änderung von Rechten

	Objekt 1	Objekt 2	Objekt 3	Domäne 1	Domäne 2	Domäne 3	Domäne 4
Domäne 1	read				switch		
Domäne 2		read					
Domäne 3			execute			switch	switch
Domäne 4	read write						

Verteilte Systeme, Sommersemester 1999 Folie 17.7

Administration

- Zusätzliche Rechte
  - Copy: (Stern hinter dem jeweiligen Recht)
    Prozesse einer Domäne dürfen das Recht in der Spalte vergeben
  - Owner: Hinzufügen und Löschen von Rechten in der Spalte
  - Control (auf Domäne): Hinzufügen und Löschen in der Zeile

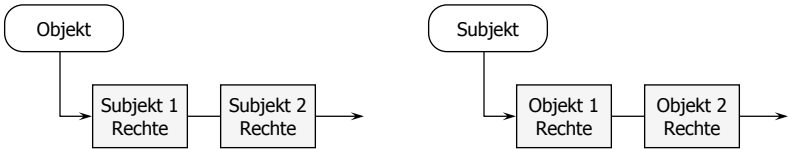
	Objekt 1	Objekt 2	Objekt 3	Domäne 1	Domäne 2	Domäne 3	Domäne 4
Domäne 1	read owner				switch		
Domäne 2		read*					
Domäne 3			execute			switch control	switch
Domäne 4	read write						

Control

Copy

Verteilte Systeme, Sommersemester 1999 Folie 17.8

Realisierungsvarianten



Zugriffskontrolllisten

- Speicherung bei Objekten
  - Welche Subjekte dürfen welche Operationen ausführen
- Beispiele
  - Windows NT
  - UNIX (Eingeschränkt auf 3 Domänen)

Capabilities

- Speicherung beim Subjekt
  - Welche Operationen dürfen auf welchen Objekten ausgeführt werden?
- Subjekte erhalten nur Verweise auf Capabilities
  - Änderung nur durch Betriebssystem

17.2 Sicherheit

"Arsenal" der Angreifer

- Wer greift an?
  - Primär reguläre Benutzer
  - "Einbrecher" (Paßwort ermittelt)
- Bedrohungen
  - Unerlaubtes Mithören
  - Fälschen von Nachrichten und Programmen
  - Unerlaubte Nutzung von Ressourcen
  - Vandalismus
- Angriffsformen
  - Nachrichtenverkehr abhören (Eavesdropping)
  - Vorspiegelung falscher Tatsachen (Masquerading)
  - Nachrichten im Verlauf der Übertragung fälschen (Message Tampering)
    - Store-and-Forward-Netze
    - Man in the Middle
  - Nachrichten speichern und später wieder einspielen (Replay)
    - Verschlüsseln allein hilft nicht

... und der Verteidiger

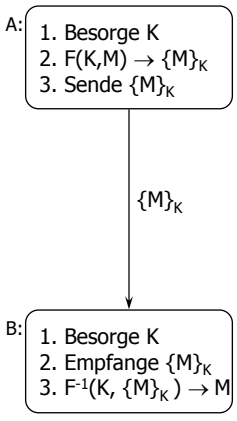
- Vertrauenswürdige Basis (Trusted Base)
- Sicherheitsanforderungen an Client/Server-Systeme
  - Kommunikationskanäle sichern
    - auch Speicherbereiche auf einem Rechner
    - verhindert Abhören
  - Client und Server müssen sich wechselseitig mißtrauen
    - Authentifizierung
    - Server: Ist Client rechtmäßiger Vertreter seines Benutzers?
    - Client: Ist Server authentisch?
  - Nachrichten mit integriertem Verfallsdatum
    - verhindert u.a. Replay

Konzepte und Techniken

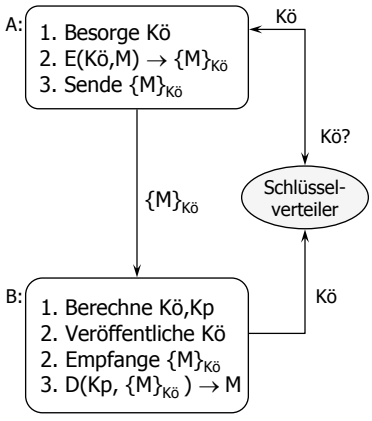
- Verschlüsselungstechniken (Kryptographie)
  - Verschlüsseln von Nachrichten
  - Grundlage für Authentifizierung
    - Wer eine Nachricht mit einem bestimmten Schlüssel erfolgreich entschlüsselt ist authentisch
  - Digitale Signaturen (authentische Unterschriften)
- Echtheit beglaubigen (Authentifizierung)
  - Ein Geheimnis, daß nur Einer kennen kann, identifiziert ihn
  - Authentifizierungsdienst
- Zugriff kontrollieren (Schutz)
  - Objekt-basiert
  - Subjekt-basiert

Kryptographie

- Geheime Schlüssel

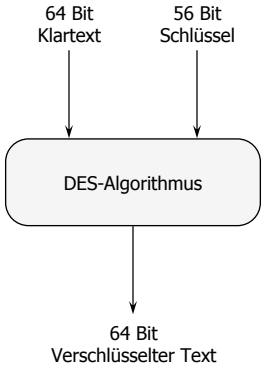


- Öffentliche Schlüssel



Geheime Schlüsselverfahren

- Data Encryption Standard (DES)
- $F=F^{-1}$
- 16 schlüsselabhängige Runden (Rounds)
  - Bit-Rotationen
- 3 schlüsselunabhängige Transpositionen
- spezielle DES-Chips
  - Faktor 1000 schneller als öffentliche Schlüsselverfahren
- siehe z.B. Tanenbaum, *Computer Networks*, 2. Auflage, 1996



Verteilte Systeme, Sommersemester 1999 Folie 17.15

Öffentliche Schlüsselverfahren

P = 948475836214346273847273464838944892837478392834792004823847923728271726262362722348728348234723842834982347298388 3
Q = 345837459387524029420372347298743984573095809853875298742987420348 2-48304573984573952929649276428764954938573983331 1

- RSA von Rivest, Shamir und Adelman (1978)
- Ansatz: Faktorisierung sehr großer Zahlen aufwendig
- Falltürfunktion f(x)=y
  - Umkehrfunktion schwierig zu berechnen
- Berechnung von Kö und Kp:
  - Wähle Primzahlen P und Q (beide größer als 10^{100})
  - N := P*Q
    Z := (P-1)*(Q-1)
  - d := Zahl relativ prim zu Z
  - e := e*d = 1 mod Z
  - Kö := <e,N> und Kp := <d,N>
- Beispiel
  - P = 13
    Q = 17
  - N := 13*17 = 221
    Z := 12*16 = 192
  - d := 5
  - e*5 = 1 mod 192

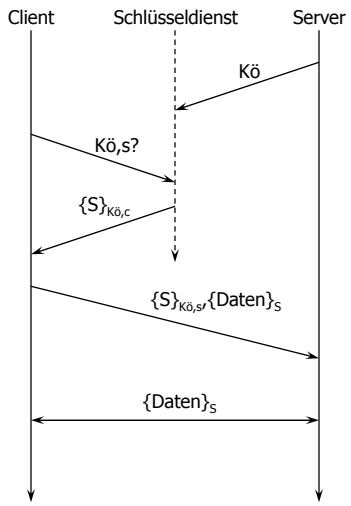
    1 mod 192 = 193, 385, 577, ...

    e := 385/5 = 77
  - Kodiert werden k Bits ($2^k < N$)
  - $E(e,N,M) = M^e \bmod N$
    $D(d,N,C) = C^d \bmod N$

Verteilte Systeme, Sommersemester 1999 Folie 17.16

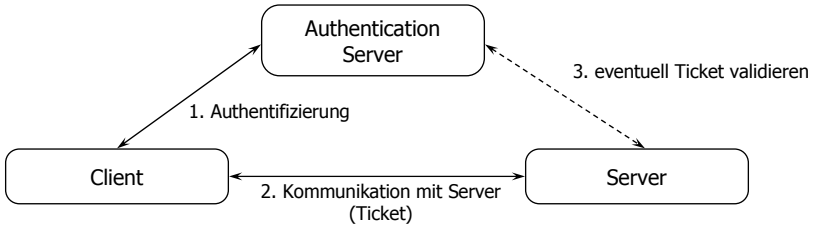
Kombinierte Techniken

- Geheime Schlüssel
  - Austausch des gemeinsamen Schlüssels aufwendig
  - Verschlüsselung großer Datenmengen effizient möglich (Hardware-Unterstützung)
- Öffentliche Schlüssel
  - Bekanntgabe und Weitergabe des öffentlichen Schlüssels unkritisch
  - Verschlüsselung zeitaufwendig
- Kombination
  - Mit Hilfe öffentlicher Schlüssel wird ein geheimer Sitzungsschlüssel (Session Key) ausgetauscht
  - Verschlüsselung der Daten mit Session Key



Verteilte Systeme, Sommersemester 1999 Folie 17.17

Authentifizierung



- Authentifizierungsdienst besonders schützen
  - Keine Benutzerprozesse
  - Firewall
  - Besonderer physischer Schutz
- Needham und Schröder, 1978
  - Authentifizierung mit geheimen Schlüsseln
  - Authentifizierung mit öffentlichen Schlüsseln

Verteilte Systeme, Sommersemester 1999 Folie 17.18

Authentifizierung mit geheimen Schlüsseln

- $C \rightarrow AS$: C, S, N_C
  - Client C möchte mit Server S kommunizieren
  - N_C = Frischedatum
- $AS \rightarrow C$: $\{N_C, S, KSession, \{KSession, C\}_{KS}\}_{KC}$
  - Authentifizierungsdienst
    - Sitzungsschlüssel KSession
    - Ticket $\{KSession, C\}_{KS}$
- $C \rightarrow S$: $\{KSession, C\}_{KS}$
- $S \rightarrow C$: $\{N_S\}_{Ksession}$
  - Ist Client authentisch oder wiederholt er nur eine Nachricht
- $C \rightarrow S$: $\{N_S + 1\}_{Ksession}$

Schwachstelle

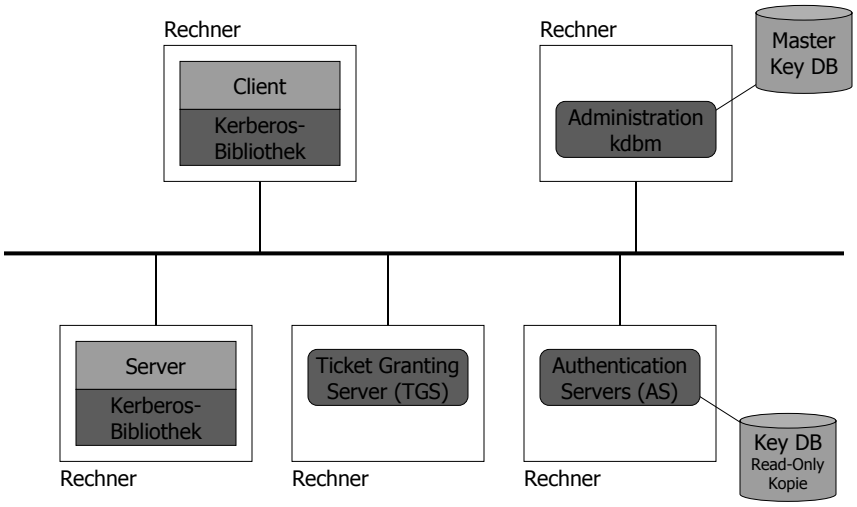
- $C \rightarrow S$: $\{KSession, C\}_{KS}$
  - Woher weiß der Server, daß empfangenes Ticket noch frisch ist?
- Man will vermeiden, daß Tickets eine beliebig lange Lebensdauer haben
  - Server steht Client nur zeitlich begrenzt zur Verfügung
  - Client wird das Nutzungsrecht zu einem späteren Zeitpunkt entzogen
- Lösung
  - Zeitstempel oder Verfallsdatum in Ticket integrieren: $\{Ksession, C, Verfallsdatum\}_{KS}$

Kerberos (Version 4)

- Authentifizierungsprotokoll (Steiner et al., 1988)
- basiert auf dem Verfahren von Needham und Schröder
  - Geheime Schlüssel
  - Erweiterung um Zeitstempel
- Integriert u.a. in
  - UNIX (MIT)
  - AFS
  - OSF/DCE
- Kerberos-Ticket: $\{Ticket(C,S)\}_{PK(S)} := \{C,S,t1,t2,Ksession\}_{PK(S)}$
  - Nur gültig im Zeitintervall [t1,t2]
- Authentication Server (AS)
- Ticket-Granting Server (TGS)

Verteilte Systeme, Sommersemester 1999 Folie 17.21

Komponenten von Kerberos

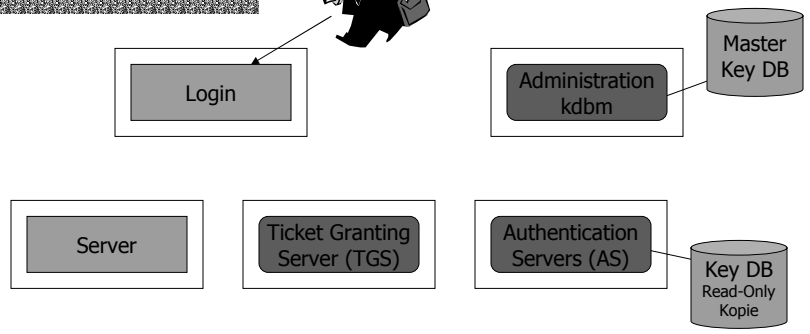


Verteilte Systeme, Sommersemester 1999 Folie 17.22

Kerberos-Namen

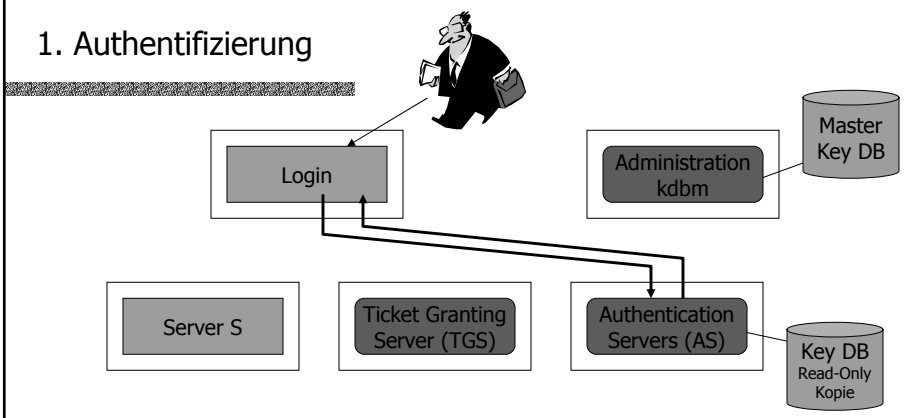
- 3 Komponenten
  - Benutzername (name)
  - Instanz (instance)
    - Verschiedene Sicherheitsstufen für einen Benutzer
    - Administratorrechte: root oder admin
    - Default: NULL
  - Realm
    - Schutz- und Authentifizierungsdomäne
- Beispiel
  - sturm@informatik.uni-trier.de
  - grutter.admin@informatik.uni-trier.de

1. Authentifizierung



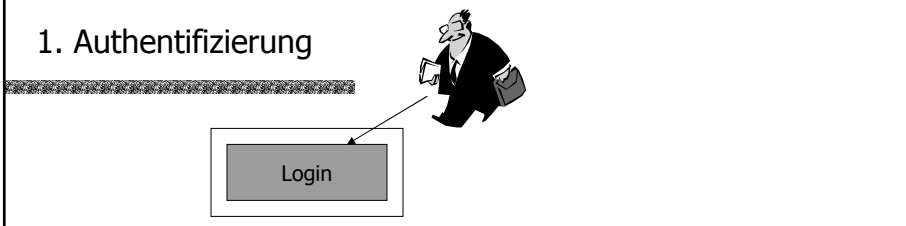
- Anmelden
  - Eingabe des Benutzernamens

1. Authentifizierung



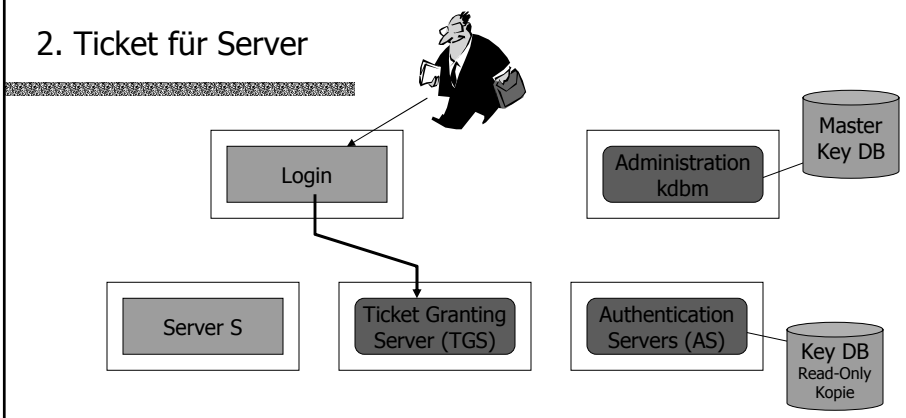
- Anfordern eines Tickets für den TGS
  - Client, TGS
- Antwort von AS
  - $\{ Session_{Client,TGS}, \{T_{TGS}\}K_{TGS} \}K_{client}$
  - T_{TGS} = (TGS, Client, Client-Addresse, Timestamp, TTL, $Session_{Client,TGS}$)

1. Authentifizierung



- Client empfängt
  - $\{ Session_{Client,TGS}, \{T_{TGS}\}K_{TGS} \}K_{client}$
- Eingabe des Passwords
- Nur der wirkliche „User" kann entschlüssen
  - Session-Key zwischen Client und TGS
  - Ticket für TGS
- Beachte: Password wurde nicht ausgetauscht
- Ticket längere Zeit nutzbar (Zeitstempel)

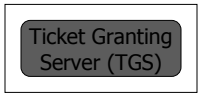
2. Ticket für Server



- Anfordern eines Tickets für den Server S
  - S, $\{T_{TGS}\}K_{TGS}$, $\{A_{Client}\}Session_{Client,TGS}$
- Authentizitätsnachweis A_{client}
  - A_{Client} = (Client, Client-Adresse, Zeitstempel)

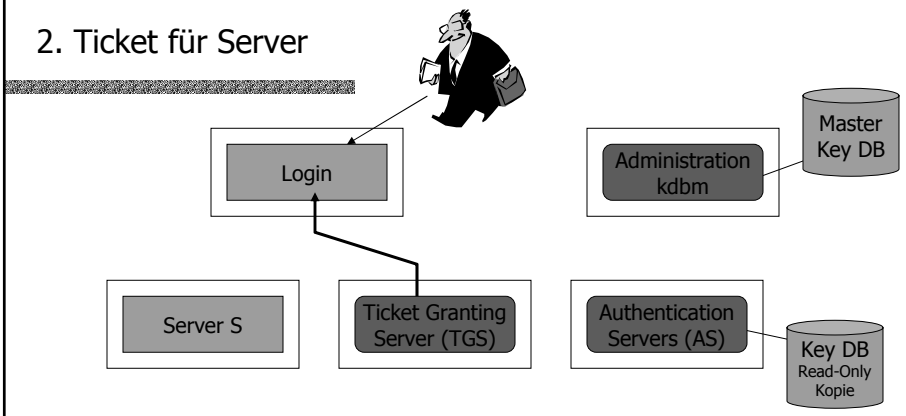
2. Ticket für Server

- TGS empfängt
  - S, $\{T_{TGS}\}K_{TGS}$, $\{A_{Client}\}Session_{Client,TGS}$



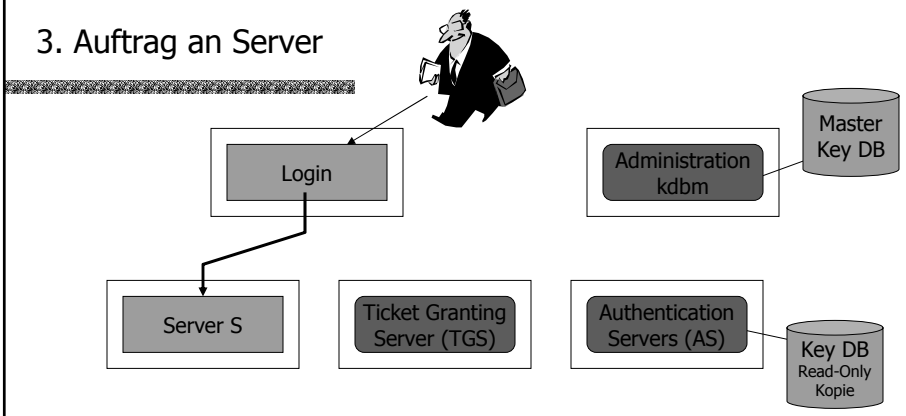
- Ticket öffnen
  - T_{TGS} = (TGS, Client, Client-Addresse, Timestamp, TTL, $Session_{Client,TGS}$)
- Authentizität überprüfen
  - A_{Client} = (Client, Client-Adresse, Zeitstempel)
- Zeitstempel darf nur wenige Minuten alt sein
  - Synchronisierte Uhren

2. Ticket für Server



- Annahme: Client darf Server benutzen
- TGS sendet Ticket für Server S an Client
  - $\{ \{T_{Client,S}\}K_S, Session_{Client,S} \}Session_{Client,TGS}$

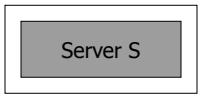
3. Auftrag an Server



- Client wendet sich an Server S
  - $\{ A_{Client} \}Session_{Client,S}, \{ T_{Client,S} \}K_S$
- Erneut Übertragung eines Authentizitätsnachweises
  - A_{Client} = (Client, Client-Adresse, Zeitstempel)

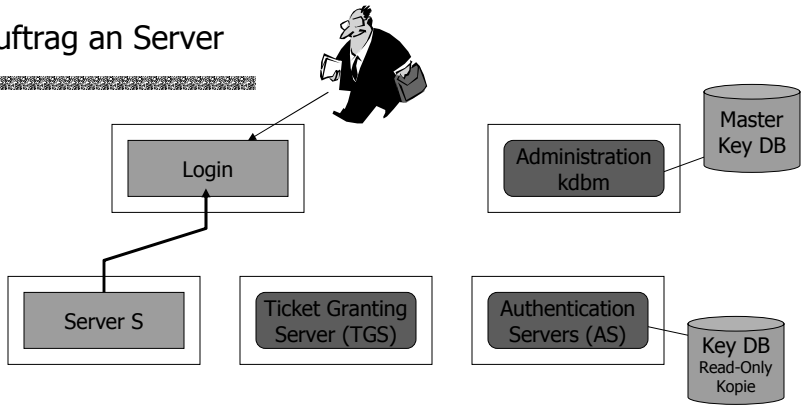
3. Auftrag an Server

- S empfängt
  - $\{ A_{Client} \}Session_{Client,S}$, $\{ T_{Client,S} \}K_S$



- Ticket öffnen
  - $T_{client,S}$ = (S, Client, Client-Addresse, Timestamp, TTL, $Session_{Client,S}$)
- Authentizität überprüfen
  - A_{Client} = (Client, Client-Adresse, Zeitstempel)
- Zeitstempel überprüfen
- Services ggf. ausführen

3. Auftrag an Server



- Client will Authentizität des Servers
  - Challenge/Response
- Server sendet an Client
  - $\{ Zeitstempel+1 \}Session_{Client,S}$

Ändern eines Passwords

- Client wendet sich AS
  - Eingabe des alten Passwords
  - AS sendet ein Ticket für kdbm an Client
  - Client sendet neues Password an kdbm
    - Verschlüsselt?
- Kdbm
  - Instanz = NULL
    - Nur eigene Passwords änderbar
  - Instanz != NULL
    - Durchsuchen einer Berechtigungsliste
    - Ggf. Password ändern
- Logging aller Vorgänge
- Analog Eintrag neuer Clients und Löschungen

Verteilte Systeme, Sommersemester 1999 Folie 17.33

Digitale Signaturen

- Wie kann man sicher sein, daß eine Nachricht von X tatsächlich von X kommt?
- Eine Lösung
  - Öffentliche Schlüssel
  - Einschalten eines vertrauenswürdigen Notars N:

    1. $A \rightarrow B$: M, A, $\{M\}_{SK(A)}$
    2. $B \rightarrow N$: A
    3. $N \rightarrow B$: A, PK(A)

  - Braucht nicht gesamte Nachricht nochmal verschlüsseln (MD5, ...)
- Anwendungen
  - Email, z.B. Privacy Enhanced Mail (PEM)
  - WWW
  - Allgemeine digitale Signaturen, z.B. Pretty Good Privacy (PGP)
  - Sichere Capabilities

Verteilte Systeme, Sommersemester 1999 Folie 17.34

Literatur

- R.M. Needham, M.D. Schröder (1978)
  Using encryption for authentication in large networks of computers
  CACM, Vol. 21, No. 12, pp. 993-999
- R.L. Rivest, A. Shamir, L. Adelman (1978)
  A method of obtaining digital signatures and public key cryptosystems
  CACM, Vol. 21, No. 2, pp. 120-126
- B. Schneier (1995)
  Applied Cryptography: Protocols, Algorithms, and Source Code
  Wiley, (2. Auflage)
- J. Steiner, C. Neuman, J. Israel (1988)
  Kerberos: an authentication service for open network systems
  Proc. Usenix Winter Conference, Berkeley