

## Netzicherheit – Architekturen und Protokolle Privilege Management Infrastructure



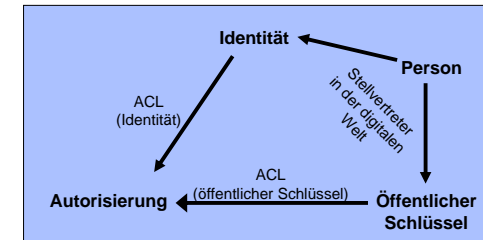
1. Privilege Management Infrastructure
2. X.509-Attributzertifikate



## PMI – Motivation

Autorisierung über **Zugangskontrolllisten** anhand von

- Identität (via Passwort/Ticket/etc., z.B. pop, imap, smtp-auth)
- Besitz eines privaten Schlüssels (z.B. SSH: authorized\_keys)



**Problem:** Zuordnung von Privilegien zu Personen lokal oder auf Privilegien-Server zentral

**Eine PMI ermöglicht eine unabhängige, transportierbare Bindung von Autorisationsdaten**

1



## Access Control Methoden

Im Folgenden betrachtete Access Control Methoden

- **Discretionary Access Control**
  - individuelle Rechte pro Benutzer
- **Mandatory Access Control**
  - jede Ressource wird klassifiziert
  - Benutzer bekommen Zugangsrechte zu Klassen
- **Role-based Access Control**
  - Rechte abhängig von der Rolle des Benutzers
  - Rollen ist eine Menge von Zugriffsrechten zugeordnet
- **Hierarchical Role-based Access Control**
  - Hierarchische Organisation der Rollen



2



## Beispiel

- Discretionary Access Control

Datei A

Datei B

Datei C

Alice:

Datei A: Lesen  
Datei B: Lesen, Schreiben

Bob:

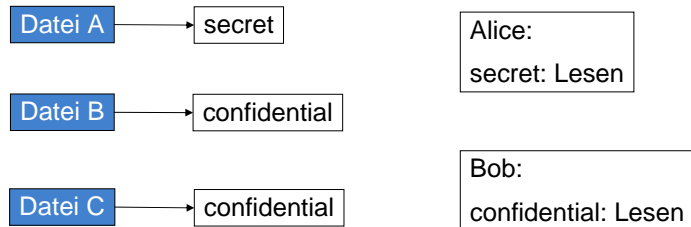
Datei B: Lesen  
Datei C: Lesen

**Ressourcen**

**Zuordnung von Privilegien**

3

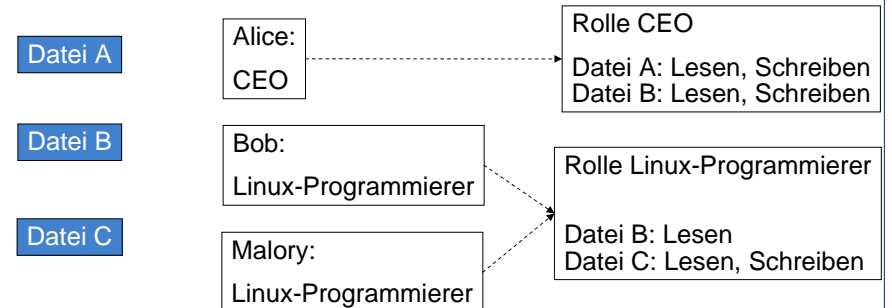
### • Mandatory Access Control



**Ressourcen**    **Klassifizierung**    **Zuordnung von Privilegien zum Zugriff auf Klassen**

4

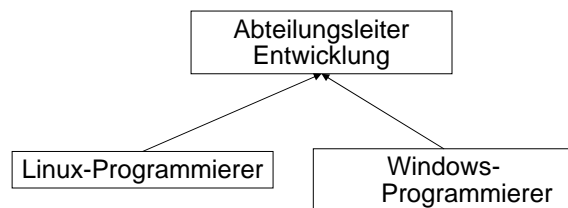
### • Role-based Access Control



**Ressourcen**    **Zuordnung von Rollen**    **Vergabe von Privilegien an Rollen**

5

### • Hierarchical Role-based Access Control



- Abteilungsleiter erhält auch die Privilegien von Linux-Programmierer und Windows-Programmierer

6

- **Authentifikation** → realisiert über ID-Zertifikate
- Wie kann **Autorisierung** realisiert werden?
  - welcher Benutzer hat in Bezug auf eine Ressource welche Privilegien?
  - wer darf Privilegien vergeben?
  - wer darf Privilegien weitergeben?

7

### Nachweis der Autorisierung über **Attributzertifikate**

- attestieren Identität bestimmtes Privileg (=Attribut)
- zeitlich einschränkbar
- basieren auf PKI und deren ID-Zertifikaten
- können widerrufen werden  
(Attribute Certificate Revocation List)

Analog zur PKI ist eine Infrastruktur zum Management dieser Zertifikate notwendig

### Privilege Management Infrastructure

8

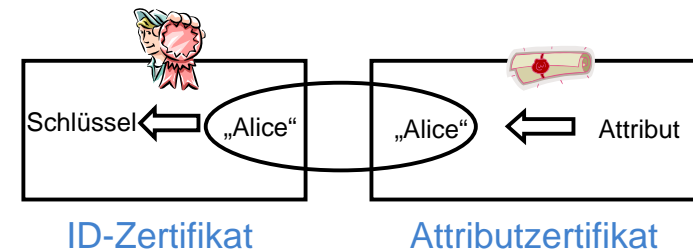
### Aufbau einer PMI ähnlich Aufbau einer PKI

- **CA der PMI: Attribute Authority (AA)**
  - vergibt Zugriffsrechte
  - zertifiziert diese Rechte in Form von Attributzertifikaten
- **Root CA der PMI: Source of Authority (SOA)**
  - Verifizierung eines Privilegs beginnt bei der SOA
  - oberste AA für dieses Privileg
  - Attributzertifikat der SOA
    - ▶ selbstzertifiziert
    - ▶ signiert mit dem zum ID-Zertifikat der SOA gehörenden privaten Schlüssel

9

| Konzept                | PKI                                  | PMI                       |
|------------------------|--------------------------------------|---------------------------|
| Zertifikatsbezeichnung | Public Key- bzw. ID-Zertifikat       | Attributzertifikat        |
| Zertifizierender       | Certificate Authority (CA)           | Attribute Authority (AA)  |
| Zertifizierter         | Subject                              | Holder                    |
| Zertifizierter Inhalt  | ID an Public Key                     | Attributwerte an ID       |
| Widerruf               | CRLs:<br>EPRLs/CARLs                 | ACRLs:<br>EARLs/AARLs     |
| Vertrauensanker        | Root Certificate Authority (Root-CA) | Source of Authority (SOA) |

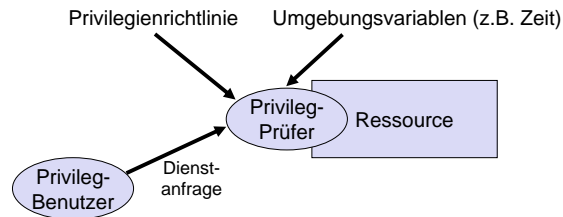
10



11

### Allgemeines Modell

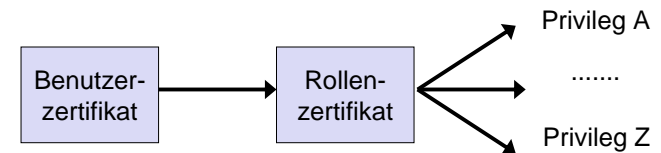
- Privileg-Prüfer (*privilege verifier*)
- Privileg-Benutzer (*privilege assenter*)
- Ressource, auf die zugegriffen wird
- Privilegien-Richtlinie (*privilege policy*)
- Umgebungsvariablen (z.B. aktuelle Zeit)



12

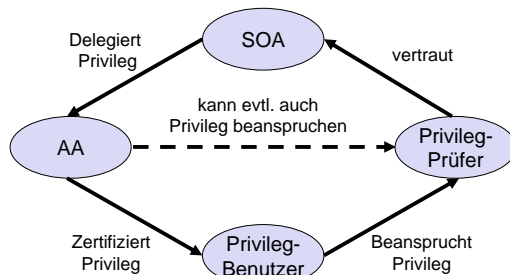
### Rollen

- Rolle steht stellvertretend für Menge von Privilegien
- Privilegien der Rolle werden ebenfalls in einem Attributzertifikat spezifiziert, können somit jeder Zeit geändert werden



13

- „Single-CA“ mit Delegation
  - SOA vergibt die Privilegien an AAs
  - AAs können Privilegien weiter delegieren
  - nur eigene Privilegien oder Untermenge delegierbar
  - ermöglicht natürliche Verteilung der Privileg-Zuweisung
  - zur Verifizierung muss Pfad von AAs aufgebaut werden



14

## Netzicherheit – Architekturen und Protokolle Privilege Management Infrastructure



1. Privilege Management Infrastructure
2. X.509-Attributzertifikate





- Seit der letzten Edition spezifiziert X.509 ein detailliertes Framework zur Autorisierung authentifizierbarer Schlüsselbesitzer
- X.509 bietet zwei Wege zur Autorisierung an
  - Privilegien in ID-Zertifikaten
  - Privilegien in Attribut-Zertifikaten

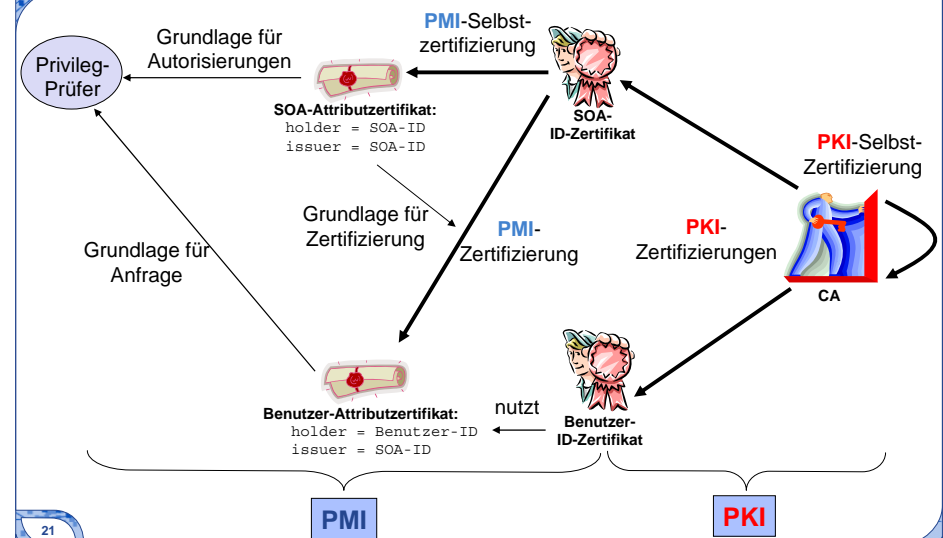
- Welche Vor- und Nachteile haben Privilegien in ID-Zertifikaten?
- Welche Vor- und Nachteile haben Privilegien in Attribut-Zertifikaten?

- X.509 Erweiterung **subjectDirectoryAttributes** enthält Privilegien
- CA muss auch AA für die Ressource sein
  - in Praxis eher nicht der Fall
- Dauer der Privilegien- und ID-Zertifizierung muss gleich sein
  - Widerruf einzelner Privilegien nicht möglich
  - Zertifikat nur als Ganzes widerrufbar
- Nur gesamte Menge an Privilegien delegierbar (keine Untermenge)

- Attestieren einer Identität bestimmte Privilegien
  - Privilegien können getrennt von ID-Zertifizierung vergeben werden
    - ▶ CA muss nicht gleichzeitig AA sein
  - mehrere unabhängige Privilegien in getrennten Zertifikaten zuweisbar
    - ▶ jedes Zertifikat kann einzeln widerrufen werden
  - Gültigkeitsdauer der Privilegien beliebig
    - ▶ insbesondere auch kürzer als zugeh. ID-Zertifikat möglich
  - Delegation einer Untermenge von Privilegien möglich

Ein Attributzertifikat besteht aus einer signierten Ausgabe folgender Daten

| Feld                  | Beschreibung  |
|-----------------------|---|
| version               | Versionsnummer des Formates   |
| holder                | Verweis auf ID-Zertifikat des Besitzers <ul style="list-style-type: none"> <li>ID der CA+Seriennummer des ID-Zertifikats</li> </ul> |
| issuer                | Verweis auf ID-Zertifikat der AA  |
| signature             | für die Signatur genutzter Algorithmus  |
| serialNumber          | Seriennummer  |
| attCertValidityPeriod | Gültigkeitsdauer <ul style="list-style-type: none"> <li>notBeforeTime</li> <li>notAfterTime</li> </ul>                              |
| attributes            | Zertifizierte Attribute   |
| issuerUniqueID        | ID der ausstellenden AA   |
| extensions            | Erweiterungen   |



Verwendung des **Erweiterungsmechanismus** um komplexere Szenarien abzubilden

### • Basiserweiterungen

#### • timeSpecification

- ▶ zeitliche Einschränkung der Nutzung (nicht Delegierung!) eines Privilegs
- ▶ z.B. während Bürozeiten, Mo - Fr von 08:00 – 18:00 Uhr

#### • acceptablePrivilegePolicies

- ▶ spezifiziert akzeptable Privilegienrichtlinien
- ▶ ist vom Dienstgeber zu prüfen
- ▶ Flag critical ist für diese Erweiterung immer gesetzt

### • Widerrufserweiterungen

#### • CRLdistributionPoint

- ▶ gibt an, wo die Attribute Certificate Revocation List zu finden ist (wie bei ID-Zertifikaten)

#### • noRevAvail

- ▶ gibt an, dass dieses Zertifikat nicht widerrufen werden kann
- ▶ oft bei kurzlebigen Zertifikaten

Optionale Autorisierung eines ID-Zertifikates, SOA sein zu dürfen, über folgende Erweiterung

- **sOAIdentifier**
  - Identifiziert den Inhaber eines ID-Zertifikates als SOA
  - ermächtigt Inhaber
    - ▶ Attribute zu definieren, Privilegien zuweisen
    - ▶ Zertifikate zu erstellen, die diese beschreiben (Attributbeschreibungszertifikat)
    - ▶ Privilegien anderen Nutzern zuzuteilen

Beschreibung der Bestandteile eines Attributes über ein *Attributbeschreibungszertifikat* das die Erweiterung **attributeDescriptor** enthält

- selbstsigniert
- beschreibt ein Attribut
  - ID, Name
  - Syntax, Beschreibung
  - Definition der Enthaltensein-Relation (*attribute domination*)

Ermöglicht die Verwendung von Rollen

- Attributzertifikat des Benutzers enthält bei Zuweisung einer Rolle folgende Erweiterung
  - **roleSpecCertIdentifier**
    - ▶ Name der Rolle
    - ▶ Aussteller des Rollen-Zertifikats
    - ▶ Seriennummer des Rollen-Zertifikats
    - ▶ Ort wo das Zertifikat zu finden ist
- Rollenzertifikat an sich
  - ist gekennzeichnet durch das Attribut **role**
    - ▶ **roleName** gibt den Namen der Rolle an
    - ▶ **roleAuthority** nennt ID der Instanz, die Rolle definiert hat
  - enthält die tatsächlichen Attributswerte

Erweiterungen zur Kontrolle der Delegation

- **basicAttConstraints**
  - gibt an, ob Zertifikatsinhaber das Privileg weiterdelegieren darf, d.h. AA ist
  - maximale Zertifikat-Pfadlänge spezifizierbar
- **delegatedNameConstraints**
  - schränkt den Namensraum der Zertifikatsbesitzer ein, denen Zertifikate ausgestellt werden dürfen
  - **permittedSubtrees, excludedSubtrees**

### Erweiterungen zur Kontrolle der Delegation

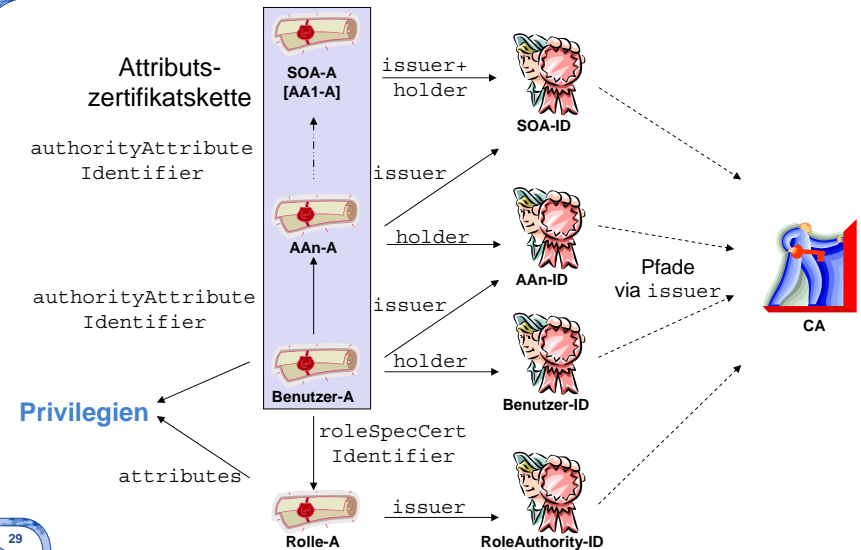
- **acceptableCertPolicies**

- Einschränkung der Zertifikatsbesitzer, die Privilegien bekommen
  - ▶ nur Besitzer gewisser ID-Zertifikate werden akzeptiert
  - ▶ Anhand von Zertifizierungsrichtlinien der CA

- **authorityAttributeIdentifier**

- verweist auf das Attribut-Zertifikat, das dem Issuer den Status als AA verleiht
- wird benutzt, um Zertifikatskette zu erstellen

28



29

1. Aufbau der Attributskette (Benutzer bis SOA)
2. Verifizieren der Signaturen der Attributskette
3. Validieren der ID-Zertifikate (siehe PKI), die in der Kette referenziert werden
4. Prüfen der Gültigkeit der Attributs- und ID-Zertifikate (Dauer, Zeitpunkt, Richtlinien, Einschränkungen, Widerruf, etc.)
5. Prüfen, ob alle Attributskette zwischen SOA und Nutzer
  1. autorisierte AAs sind
  2. gültig sind in Hinsicht auf Pfad- und Namensbeschränkungen
  3. durch ID-Zertifikate authentifiziert sind, die unter einer gegebenen Richtlinie gültig sind
  4. die Dominanzregel korrekt angewandt haben (nicht mehr Rechte weitergegeben wurden, als man selbst besitzt)

30

### Realisierung der Modelle mit X.509

- **Discretionary Access Control**
  - individuelle Privilegien in Attributskette der Benutzer
- **Mandatory Access Control**
  - Benutzer werden Klassen und erlaubte Aktionen in Attributskette zugewiesen
- **Role-based Access Control**
  - mittels **roleSpecCertIdentifier** Verweis auf Rollen-Zertifikat
  - Rollen-Zertifikat ist Attributskette mit Attribut **role**
- **Hierarchical Role-based Access Control**
  - innerhalb von Rollen-Zertifikat Verweis auf weitere Rollen möglich

31



- Vorteile der X.509-PMI
  - ☺ Aufgabentrennung möglich
    - ▶ PKI für Authentifizierung zuständig
    - ▶ PMI mit AAs für Autorisierung zuständig
  - sehr flexible Aufgaben- und Rechteverteilung möglich
  - ☺ Abbildung auf Unternehmensstrukturen einfach durch Delegation
- Nachteile der X.509-PMI
  - ☹ Autorisierung mit Attributzertifikaten geschieht in zwei Schritten
    - ▶ öffentlicher Schlüssel – Identität
    - ▶ Identität – Privileg
 und ist somit an zwei Stellen angreifbar
  - ☹ Validierung aufwendig
  - ☹ Autorisierung basiert auf Namen, die jedoch selten relevant sind (Alternative: Autorisierung basierend auf öffentlichem Schlüssel)

- ITU-T Recommendation X.509 (03/00)  
<http://www.itu.int/rec/T-REC-X.509-200003-I/en>
- David Chadwick, „The X.509 Privilege Management Infrastructure“,  
<https://www.cs.kent.ac.uk/pubs/2004/2278/content.pdf>
- PKI, PMI und X.509 Zusammenfassung  
<http://www.cryptoshop.com/de/knowledgebase/pki/index.php>