

---

# Das Kerberos 5 Protokoll

## Ein Tutorium

**Dr. Dieter Mack**

*Mack@uni-hohenheim.de*

<http://www.uni-hohenheim.de/rz/mack>

**Rechenzentrum der Universität Hohenheim**

**AFS Workshop 2003 - 7. bis 10. Okt. 2003**

**Deutsches Elektronen-Synchrotron (DESY) Zeuthen**



# Das Problem

---

**Sicherheit in einer vernetzten Umgebung:**

***Authentication of Unknown Entities  
on an Insecure Network  
of Untrusted Workstations***

B. Clifford Neumann, Jennifer G. Steiner

Authentisierung unbekannter Einheiten  
in einem unsicheren Netzwerk  
von nicht vertrauenswürdigen Rechnern



# Authentisierung

---

- **der Nachweis, daß ein Agens wirklich dasjenige ist, welches es zu sein vorgibt**
  - das Agens (Kerberos: Principal) kann sein:
    - ein menschlicher Benutzer
    - ein Rechner, Server oder Service
- **wechselseitige Authentisierung zwischen Klienten und Servern ist wichtig**
  - die Bank will wissen, wer vor dem Automaten steht
  - der Kunde sollte aber auch wissen wollen, ob der Automat **wirklich** von seiner Bank aufgestellt worden ist
- **sie hat durch einen zentralen Security Service auf einem physikalisch sicheren System in fälschungssicherer Weise zu erfolgen**



# Autorisierung

---

- **die Feststellung, ob ein Agens im Netzwerk zu einem spezifischen Zugriff auf eine bestimmte Ressource oder ein bestimmtes Objekt berechtigt ist**
- **dies setzt natürlich vorherige Authentisierung voraus**
- **prinzipiell kann diese zentral oder dezentral erfolgen**
- **die Arten des Zugriffs sind i.a. von der Art der Objekte und der sie verwaltenden Anwendung abhängig**
- **dies legt nahe, die Autorisierung dezentral als Teil der jeweiligen Anwendung anzulegen**
- **DCE verfolgt diese Strategie, bietet dabei aber einen Rahmen zur einheitlichen Gestaltung**



# Kerberos

---

- **Kerberos ist ein Authentisierungs-System**
  - Kerberos liefert einen fälschungssicheren befristeten Nachweis der Identität eines Principals: das Ticket
  - die DCE Security liefert zusätzlich die Gruppen, zu denen ein Principal gehört, um hiermit die Autorisierung über Gruppen zu ermöglichen
- **Kerberos ist ein 'Trusted Third Party' System**
- **Fälschungssicherheit und Identitätsnachweise werden durch Kryptographie gewährleistet**
- **die Identität eines Principals wird durch einen Schlüssel als gemeinsames Geheimnis ausgewiesen, das nur dem Principal und dem Kerberos-Server bekannt ist**
  - für menschliche Principals wird dieser aus einem Passwort ermittelt
  - dieses geht nie im Klartext übers Netz



# Kerberos (2)

---

- **die Fähigkeit, eine Nachricht zu entschlüsseln, dient als Identitätsnachweis**
- **hierzu erzeugt und verteilt Kerberos auch temporäre Schlüssel, die Session Keys**
- **hiermit wird erreicht, daß Klient und Server nach erfolgreicher Authentisierung auch ein gemeinsames Geheimnis haben**
- **deshalb wird der Kerberos-Server auch als KDC, als Key Distribution Center, bezeichnet**



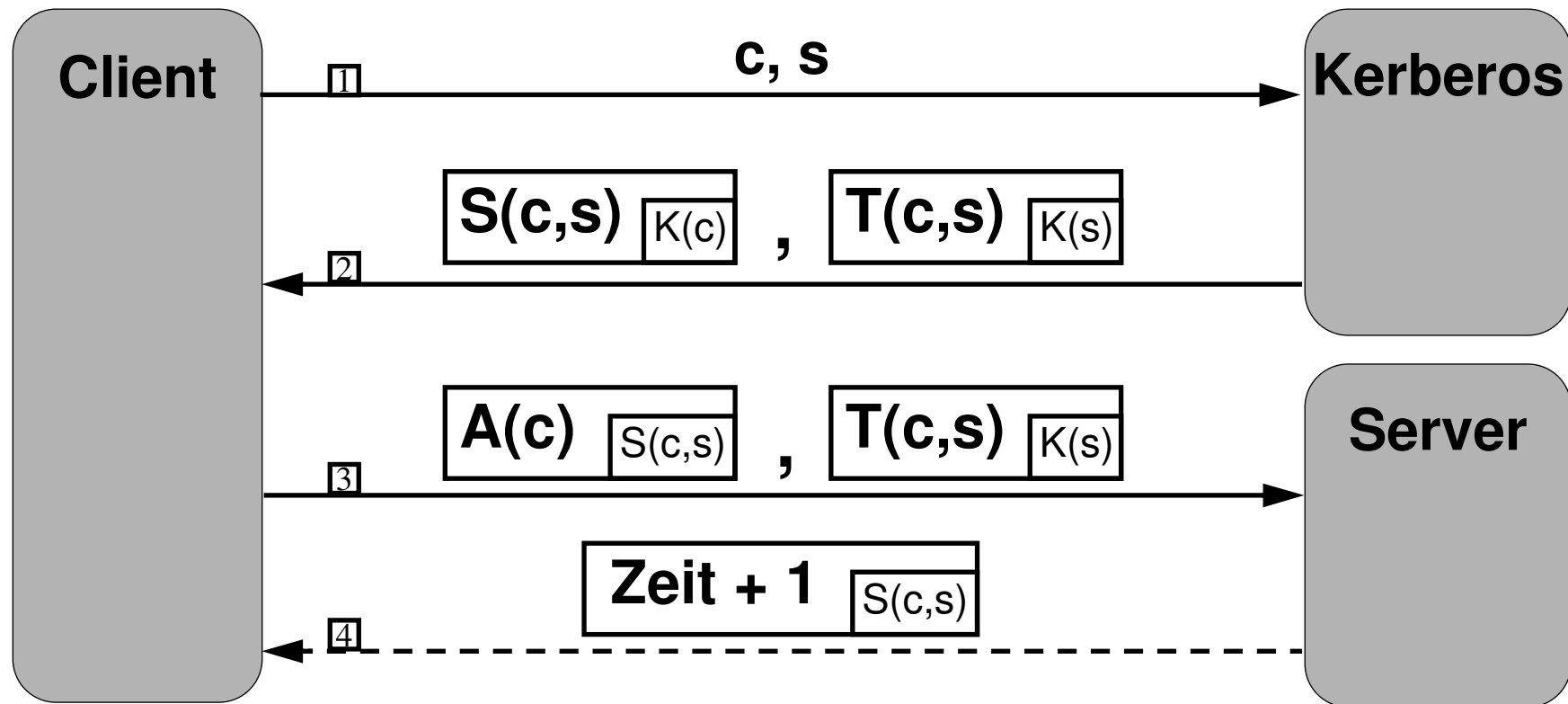
# Tickets

---

- **der Klient erhält auf Verlangen vom Kerberos-Server ein Ticket als fälschungssicheren Identitätsnachweis**
- **das Ticket enthält, unter anderem:**
  - den Namen von Klient und Server
  - den Session Key
  - Zeit und Lebensdauer
- **mit dem Ticket weist sich der Klient beim Server aus**
- **das Ticket ist im Schlüssel des Servers verschlüsselt**
  - es kann also nur von diesem Server gelesen werden
  - der Klient kann es weder lesen noch modifizieren
- **man benötigt also ein Ticket pro Server / Service**



# Das Authentisierungs-Protokoll



$K(x)$  = privater Key von  $x$

$S(x,y)$  = Session Key für  $x$  und  $y$

$T(c,s)$  = Ticket von  $c$  für  $s = \{c, s, S(c,s), \text{Zeit, Lebensdauer, ...}\}$

$A(c)$  = Authenticator von  $c = \{c, \text{Zeit, ...}\}$

$\boxed{abc \mid K}$  =  $abc$ , verschlüsselt mit  $K$



# Das Authentisierungs-Protokoll 1

---

## Dialog zwischen Klient und Kerberos Server:

1. der Klient  $c$  verlangt die Authentisierung für Server  $s$
2. Kerberos schickt:
  - a) Session-Key  $S(c,s)$  für den Verkehr zwischen  $c$  und  $s$   
(und weiteres wie Lebensdauer, ...), mit dem Schlüssel  $K(c)$  des Klienten verschlüsselt
    - nur Kerberos und der Klient  $c$  kennen  $K(c)$
    - also erlangt nur der Richtige Klient Kenntnis von  $S(c,s)$
  - b) Ticket  $T(c,s)$ , um  $c$  bei  $s$  auszuweisen, mit dem Schlüssel  $K(s)$  des Servers verschlüsselt
    - dieses kann der Klient nicht entziffern, sondern nur aufheben und an den Server weitergeben



# Das Authentisierungs-Protokoll 2

---

## Dialog zwischen Klient und Server:

3. der Klient  $c$  schickt zum Server  $s$ :

a) das in Schritt 2b vom Kerberos erhaltene Ticket  $T(c,s)$

→ wenn der Server dieses entschlüsseln kann, muß es vom Kerberos erzeugt worden sein

→ somit kennt nun auch der Server den Session-Key  $S(c,s)$

b) einen im Session-Key  $S(c,s)$  verschlüsselten Authentikator  $A(c)$ , der unter anderem den Namen des Klienten und vor allem einen Zeitstempel enthält

→ kann der Server diesen mit  $S(c,s)$  entschlüsseln, so muß er vom Klienten  $c$  erzeugt worden sein, da sonst niemand weiteres den Session-Key kennt

→ ist der Zeitstempel außerhalb der Toleranz, so ist das allerdings verdächtig



# Das Authentisierungs-Protokoll 3

---

Nur wenn der Klient dies verlangt, findet der letzte Schritt zur **Wechselseitigen Authentisierung** statt:

4. der Server schickt dem Klienten einen mit dem Session-Key  $S(c,s)$  verschlüsselten um 1 inkrementierten Zeitstempel zurück

→ somit weiß der Klient, daß der Server:

1. den Session-Key  $S(c,s)$  kennt
2. also das Ticket  $T(c,s)$  entschlüsselt hat
3. folglich den Schlüssel  $K(s)$  des Servers kennen muß, und
4. also folglich auch der verlangte Server ist



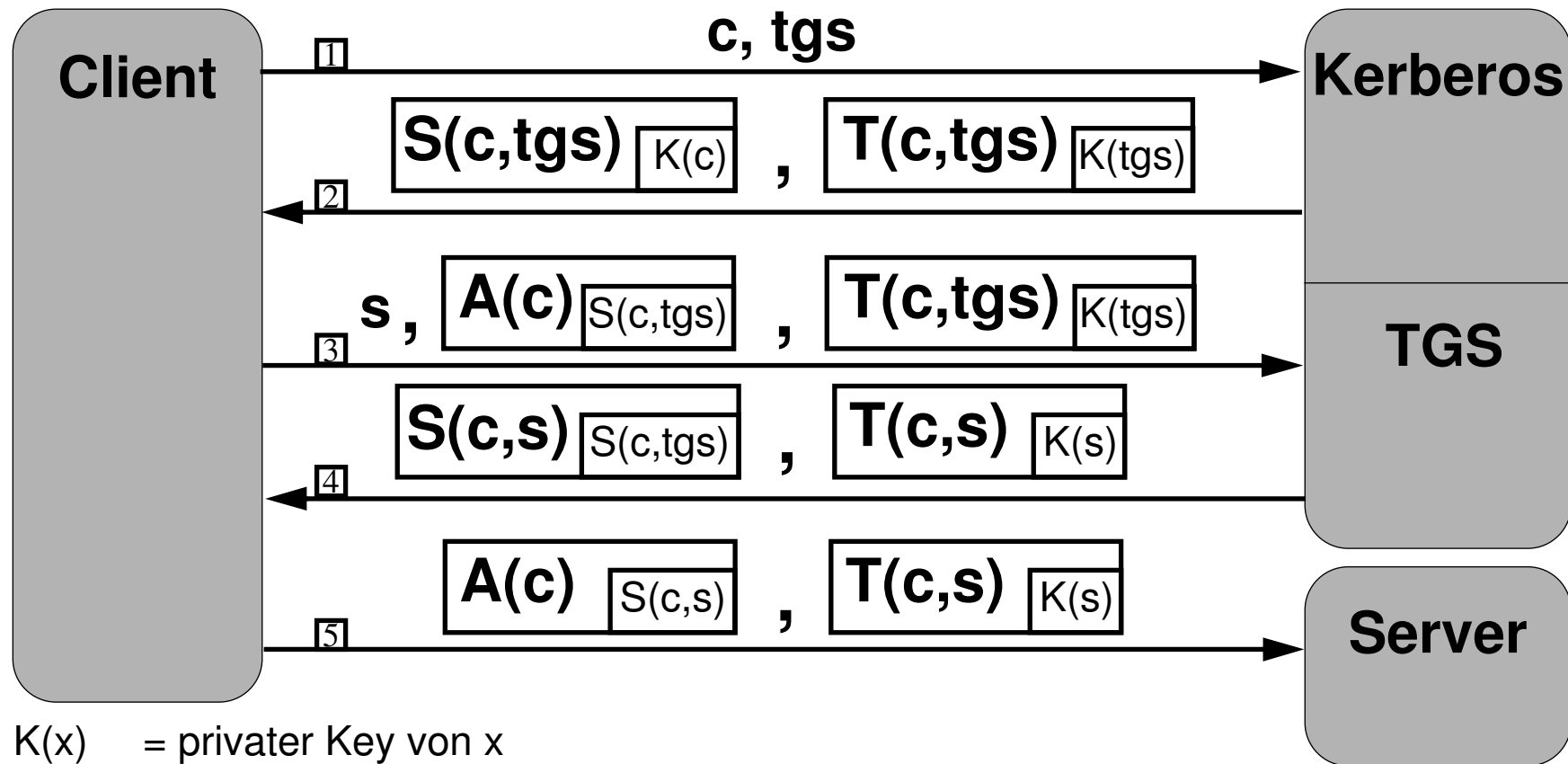
# Der Ticket Granting Service

---

- bei diesem Protokoll muß der Klient für jeden Server im Schritt 2 seinen Schlüssel  $K(c)$  verwenden
- dies ist für menschliche Principals nicht praktikabel
- der Trick, um dies zu vermeiden, ist der Ticket Granting Service
- dieser liefert einem von ihm authentifizierten Principal Tickets und Session-Keys für andere Server
- da der TGT hierfür Zugriff zur Kerberos-Datenbasis haben muß, muß dieser als Teil des Kerberos-Servers realisiert werden



# Das Ticket Granting Protokoll



$K(x)$  = privater Key von  $x$

$S(x,y)$  = Session Key für  $x$  und  $y$

$T(c,s)$  = Ticket von  $c$  für  $s = \{c, s, S(c,s), \text{Zeit, Lebensdauer, ...}\}$

$A(c)$  = Authenticator von  $c = \{c, \text{Zeit, ...}\}$

$\boxed{abc \mid K}$  =  $abc$ , verschlüsselt mit  $K$



# Das Ticket Granting Protokoll

---

- die Schritte 1 und 2 sind identisch mit dem normalen Authentisierungs-Protokoll
- anschließend hat der Klient Ticket und Session-Key für einen speziellen Service: den TGS
- normale authentifizierte Anfragen an den TGS fordern sodann für weitere Server jeweils Ticket  $T(c,s)$  und Session-Key  $S(c,s)$  an
- nur dieser Dialog wird für jeden weiteren Server wiederholt
- hierbei geschieht der verschlüsselte Teil des Dialogs zwischen Klient und TGS mittels des gemeinsamen Session-Keys  $S(c,tgs)$
- nach diesem jeweiligen 4.Schritt besitzt der Klient für den gewünschten Server die gleiche Information wie nach Schritt 2 des einfachen Authentisierungs-Protokolls
- damit kann er sich dann bei dem gewünschten Server ausweisen

**Achtung:** die Schritte zur wechselseitigen Authentisierung sind hier weggelassen



# Anmerkungen

---

- **im originalen Protokoll gibt der Kerberos-Server jedem auf Anfrage ein TGT und einen Session-Key**
- **dieses kann dann offline zum Versuch, Passwörter zu 'erraten', verwendet werden**
- **Abhilfe schafft die sog. Pre-Authentication: nur wenn der Klient einen in seinem Schlüssel verschlüsselten Zeitstempel schickt, wird ihm überhaupt geantwortet**
- **aber anschließend kann man mit den selben Methoden versuchen, den Schlüssel des TGS zu erraten**
- **Moral: dieser sollte häufig, am besten automatisch, geändert werden**



# Zu Guter Letzt

---

**die Ultimate Quiz-Frage:**

**reicht es für ein 'kerberisiertes Login', ein TGT anzufordern und zu prüfen, ob der im Schlüssel des Benutzers verschlüsselte Teil mittels des aus dem eingegebenen Passwort erzeugten Schlüssels entschlüsselt werden kann???**

**und warum???**

