XII. Authentifikation

XII.1. Definition

Authentifizierung bindet eine Identität an ein Subjekt.

XII.2. Ansätze

- Entität weiß etwas (Kennwort)
- Entität ist etwas (Biometrie)
- Entität hat etwas (Chipkarte)
- Entität kann etwas (Captcha)
- Entität befindet sich an einem bestimmten Ort

XII.3. Komponenten

- Menge A der Authentifizierungsinformation
  (Information, mit der Identität bewiesen wird)
- Menge C der Komplementärinformationen
  (was das System speichert, um Authentifizierung zu validieren)
- Menge $\mathcal{F} \in C^A$ der Komplementierungsfunktionen
  (leitet aus gegebenem $a \in A$ das entsprechende $c \in C$ ab)
- Menge $L \subseteq \{true, false\}^{A \times C}$ der Authentifikationsfunktionen
  (verifiziert Identifikation)
- Menge S der Auswahlfunktionen
  (zum Anlegen, Ändern, Entfernen von Entitäten und entsprechenden Daten)

XII.4. Typische Anwendung: Kennworte

- 1. Ansatz: System speichert Kennworte explizit
  $\rightarrow$ Problem: Diebstahl des Passwordfiles
- 2. Ansatz: kryptographische Hashwerte der Kennworte speichern
  $\rightarrow$ Problem: Offline-Wörterbuchattacke sehr effizient
- 3. Ansatz: Saltung (pro Benutzer andere Hashfunktion): $H_s(pw) := H(pw \| s)$
- 4. Ansatz: „Remote-Login“ mit dediziertem Authentifizierungsserver

XII.5. Maßnahmen gegen Offline-Attacken

XII.5.1. Wahl guter Kennworte

- vorgegebene Zufallsstrings (werden aufgeschrieben und am Rechner deponiert)
- „Key-Crunching“ → Hashing langer Passphrases
- Verschleierung aufgeschriebener Kennworte (einfache Transformation)
- proaktive Kennwortwahl
- zeitliche Variation (ganz gut: ab und zu Kennwort verlängern)
- Security Awareness

XII.6. Maßnahmen gegen Online-Attacken

- Backoff → nach n Fehleingaben Sperrung für x_n Sekunden
- Disconnection → nach n Fehleingaben Verbindungstrennung
- Jailing → begrenzter Zugriff wird trotz Fehleingabe gewährt, oft mit Honeypots kombiniert

XII.7. Beispiel: CAPTCHAs

automatisch generierte Rätsel, die Maschinen nur sehr schwer lösen können, Menschen dagegen sehr leicht

XII.8. Raffiniertere Verfahren (Challenge-Response)

XII.8.1. Schema

Benutzer hat Geheimnis s

FIXME: Bild Schema, S. 53

Das Geheimnis S soll nicht aus c und $c(r, s)$ rekonstruierbar ein (selbst bei böswillig gewähltem c).

XII.8.2. Beispiele

RSA-Signaturen

Server schickt String, lässt ihn sich signieren → in der Praxis manchmal zu aufwändig

mittels Hashfunktion oder Verschlüsselung

$r(s.c) = h(s, c)$ oder $r(s, c) = Enc_s(c)$ → Server muss Geheimnis S kennen

Zero Knowledge

→ in der Praxis zu aufwändig

SPEKE (Simple Password Encrypted Key Exchange)

Parameter:

- $p = 2q + 1$, $p, q \in \mathbb{P}$ („safe prime“)
- Hashfunktion H
- $g = H(Passwort)^2 \mod p$ (erzeugt die Gruppe der quadratischen Reste $\mod p$)

Ablauf: wie Diffie-Hellman, aber $key := H(g^{ab})$ sowie mit Key Confirmation

FIXME: Bild Ablauf, S. 53

möglicher Angriff: schicke $g^a = 1$ oder g^a mit kleiner Ordnung → Schlüssel unabhängig von Passwort → Lösung: Protokollabbruch, falls $Ord(g^{ab} < q$