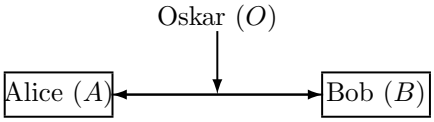
11 Instanzauthentisierung

B (Prüfer) kann die Identität von A (Beweisender) zweifelsfrei feststellen
Angreifer O versucht, Identität von A zu übernehmen (aktiver Angriff)



Faktoren für die Überprüfung:

- Wissen (z.B. ein Passwort, eine PIN (Personal Identification Number))
- Besitz (z.B. ein Schlüssel in einer sicheren Chipkarte)
- Eigenschaften (z.B. ein biometrisches Merkmal)

$n-$Faktor-Authentisierung:

- 1-Faktor-Authentisierung: Nutzt nur einen Faktor
  z.B. Abrufen von E-Mails: Benutzername/Passwort (Wissen)
- 2-Faktor-Authentisierung: Nutzt zwei verschiedene Faktoren
  z.B. Auszahlung am Geldautomaten: Karte (Besitz) und PIN (Wissen)

Faktor Wissen Typisches Beispiel: Benutzername/Passwort

Nachteile: Anfällig gegen

- Ausspähen (z.B. über Phishing, Keylogging, Abhören der Verbindung)
  Replay-Attacken: Abhören der Verbindung und Wiedereinspielen
- Man-in-the-Middle Attacken

Verbesserung: Einmalpasswörter

- Jedes Passwort wird nur einmal verwendet: Verhindert Replay-Attacken

Problem: Beide Seiten müssen die Passwörter kennen. Zwei Möglichkeiten: Passwortlisten, Passwortgeneratoren

Passwortlisten: z.B. Transaktionsnummern (TAN) im Online-Banking
Beide Kommunikationspartner erhalten eine Liste mit Passwörtern

Indizierte Auswahl ist ein Challenge-Response Protokoll

$\boxed{B}$ Prüfer (Liste L)		$\boxed{A}$ Beweisender (Liste L)
wähle Index i		
(challenge)	$\xrightarrow{i}$	suche $i-$tes Passwort p_i
prüfe	$\xleftarrow{p_i}$	(response)

Passwortgeneratoren: Ableitung von Passwörtern aus einem Geheimnis

- zeitgesteuerte, ereignisgesteuerte und Challenge-Response-Generatoren

Zeitgesteuerte Generatoren: Beispiel Google Authenticator

- key $= g$ (vorab ausgetauschtes Geheimnis)
- $t =$ time (in sec (Zeitpunkt der Authentisierung))
- message $= t/30$ (Toleranzbereich: Zeitintervall von 30 Sekunden)
- $p =$ MAC(key, message) (Einmalpasswort für Zeitpunkt t)
  MAC: Message Authentication Codes, z.B. HMAC

Ereignisgesteuerte Generatoren: Beispiel Lamport-Hash

Basiert auf einer kryptographischen Hashfunktion H (Einweg)

- g (vorab ausgetauschtes Geheimnis)
- Zufallszahl r (muss nicht geheim gehalten werden)
- Startwert $S = H(r||g)$
- Generierung der Einmalpasswörter:

  - Erstes Passwort: $p_1 = H^N(S)$ (N mal Anwenden von H)
  - Zweites Passwort: $p_2 = H^{N-1}(S)$ ($N-1$ mal Anwenden von H)
  - $t-$tes Passwort: $p_t = H^{N-(t-1)}(S)$
- Aus $p_t = H^{N-(t-1)}(S)$ lässt sich nicht $p_{t+1} = H^{N-(t-2)}(S)$ berechnen
  $H^{N-(t-1)}(S) \mapsto H^{N-(t-2)}(S)$ ist die Umkehrung von H auf $H^{N-(t-1)}(S)$

Problem: Irgendwann wurden N Passwörter erzeugt

- Reinitialisierung: Wähle einen neuen Zufallswert r
- Bilde neuen Startwert $S = H(r||g)$

Challenge-Response Verfahren

Vorab ausgetauschtes Geheimnis: Ein symmetrischer Schlüssel k

$\boxed{B \text{ (Prüfer)}}$ Schlüssel: k		$\boxed{A \text{ (Beweisender)}}$ Schlüssel: k
choose random c		
(challenge)	$\xrightarrow{c}$	compute $r := \text{MAC}(k, c)$
	$\xleftarrow{r}$	(Einmalpasswort, response)
compute $r' := \text{mac}(k, c)$		
if $r' = r$ then accept		
else reject		

Anstelle eines (symmetrischen) MACs kann auch ein (asymmetrisches) Signaturverfahren genutzt werden

Einmalpasswörter: Vorteile:

- Sicher gegen Ausspähen: Jedes Mal ein neues Passwort
- Verfahren verhindert somit Replay-Attacken

Übung: Untersuhen Sie die Verfahren hinsichtlich ihrer Resistenz gegen Phishing und Key-Logging.

Weiterhin möglich: Man-in-the-Middle-Angriff
Verhinderung des Angriffs: Gegenseitige Authentisierung

- Nicht nur A muss sich gegenüber B authentisieren,
- sondern auch B gegenüber A

Faktor Besitz

- A (Beweisender) besitzt einen geheimen Schlüssel k
- Ziel: Sichere Speicherung des Schlüssels
  - Schlüssel soll von keinem Unbefugten ausgelesen werden können
  - Schlüssel soll von keinem Unbefugten genutzt werden können

Sicherheitselemente Nutzung sicherer Hardware (Sicherheitschips)
Microprozessoren, die gegen Angriffe geschützt sind, z.B. gegen

- physikalische Attacken (bohren, fräsen, ...)
- elektrische Angriffe (mehr Strom, als Spezifikation erlaubt)
- Angriffe mit Licht und Laser

Detektoren erkennen Angriffe, Schlüsselspeicher wird gelöscht

2-Faktor-Authentisierung basierend auf

- Besitz (Sicherheitselement) und Wissen (PIN)

Umsetzung

- Speicherung von Schlüssel und PIN im sicheren Bereich des Chips
- Authentisierung:
  - über Challenge-Response Verfahren
  - Nutzung des Schlüssels wird über PIN freigegeben

Anwendungsbeispiele:

- Bankkarten (Geldabheben an Bankautomaten)
- Kreditkarten (Bezahlen am Point of Sale)
- Personalausweis (Authentisieren mit der Online-Ausweisfunktion)

Biometrie Beobachtung und Messung von Merkmalen des Menschen
Ziel: (Wieder-)Erkennung

- Biologische Merkmale: Fingerabdruck, Iris, Gesicht, ...
- Verhaltensmerkmale: Stimme, Gang, Tippverhalten, ...

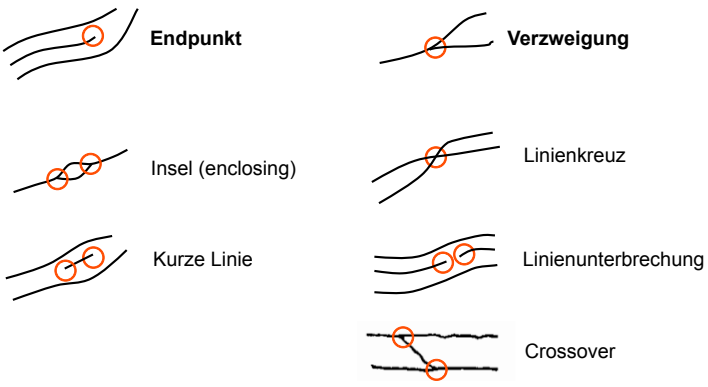
Grundsätzliches Vorgehen:

- Enrolment: Aufnahme biometr. Merkmale, Verknüpfung mit Person
- Authentisierung: Aufnahme biometr. Merkmale, Vergleich

Enrolment Häufig Bearbeitung der biometr. Merkmale
Nicht Speicherung der gesamten Information (z.B. Fingerabdruck), sondern Extraktion und Speicherung der charakteristischen Merkmale (Template)

- Vergleiche von Templates effizienter und fehlertoleranter
- Keine Rekonstruktion der vollen Information aus Template möglich Datenschutz

Beispiel (Fingerabdruck). Charakteristische Merkmale: Minutien



Extraktion der Minutien und Speicherung als 2-dim. Vektor (Template)

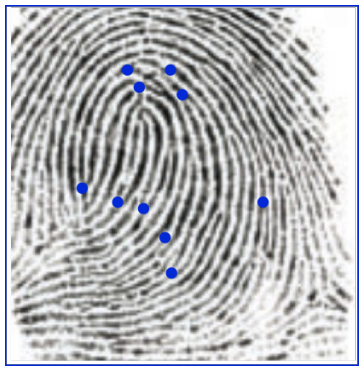
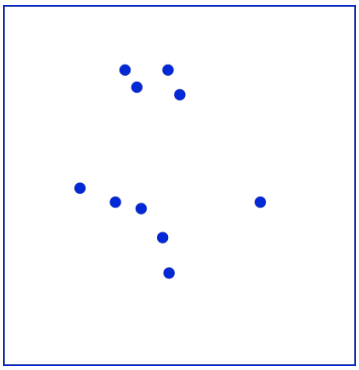


Bild Fingerabdruck



Template

Authentisierung Zwei Schritte: Aufnahme und Vergleich

- Aufnahme des biometrischen Merkmals
  Prozesse zur Erkennung von Angriffen (z.B. Lebenderkennung)
- Vergleich:
  - Extraktion der charakteristischen Merkmale
  - Vergleich mit Template und Entscheidung

Beispiel (Fingerabdruck).

- Zwei 2-dim. Vektoren $((x_1, y_1), \ldots, (x_n, y_n))$, $((x'_1, y'_1), \ldots, (x'_n, y'_n))$ vor
- Vergleich in geeigneter Metrik (Minkowski-Metrik, Summe der Abstände)

$$s = \sum_{i=1}^{n} \sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2}$$

- Im Idealfall gilt $s = 0$ (klappt aber nicht)
- Wir müssen s so wählen, dass
  - falschen Merkmale nicht akzeptiert werden (False Accept Rate)
  - richtige Merkmale nicht abgelehnt werden (False-Reject-Rate)