Vorlesung (WS 2014/15)

Sicherheit: Fragen und Lösungsansätze

Dr. Thomas P. Ruhroth

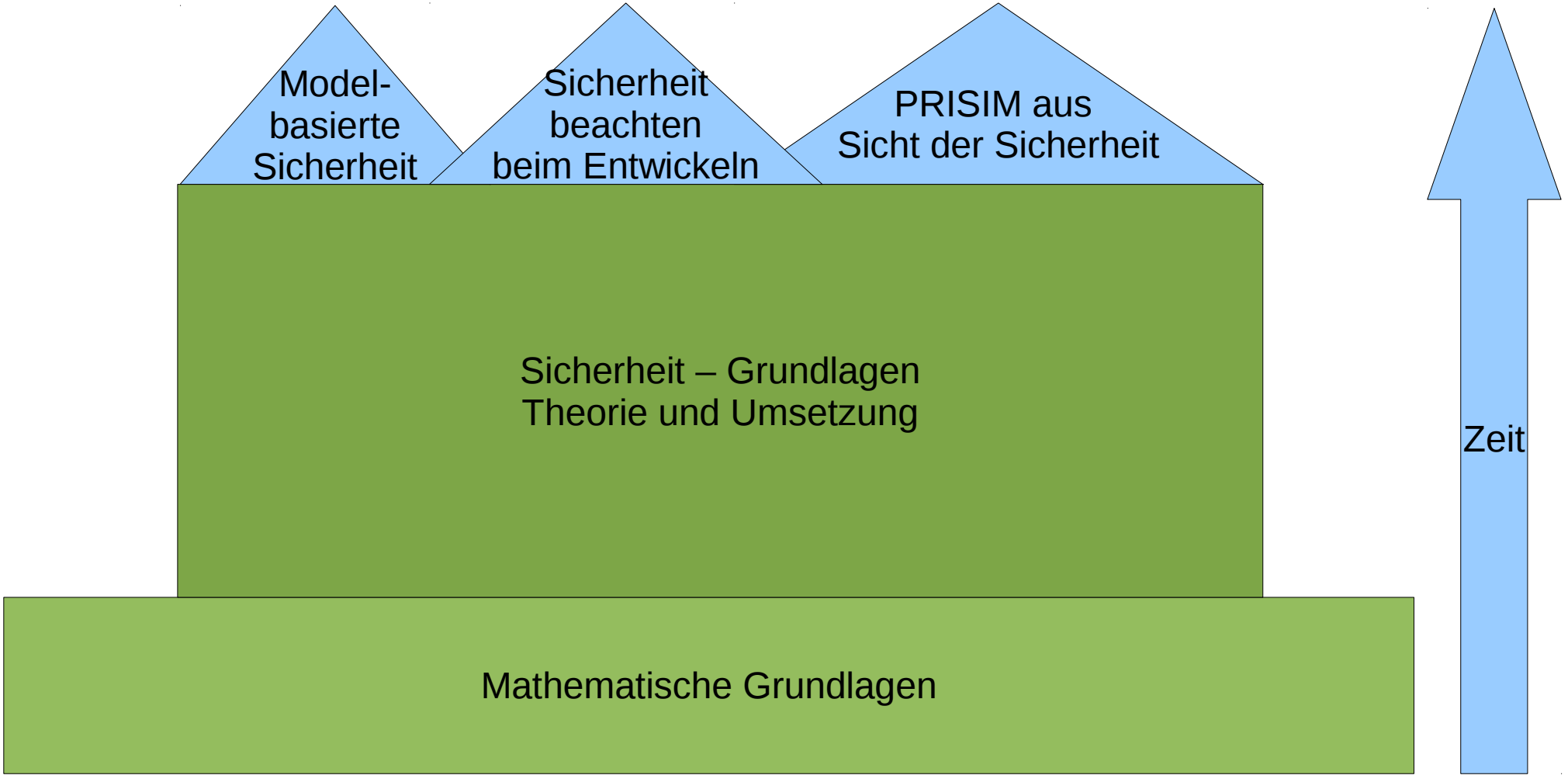
TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

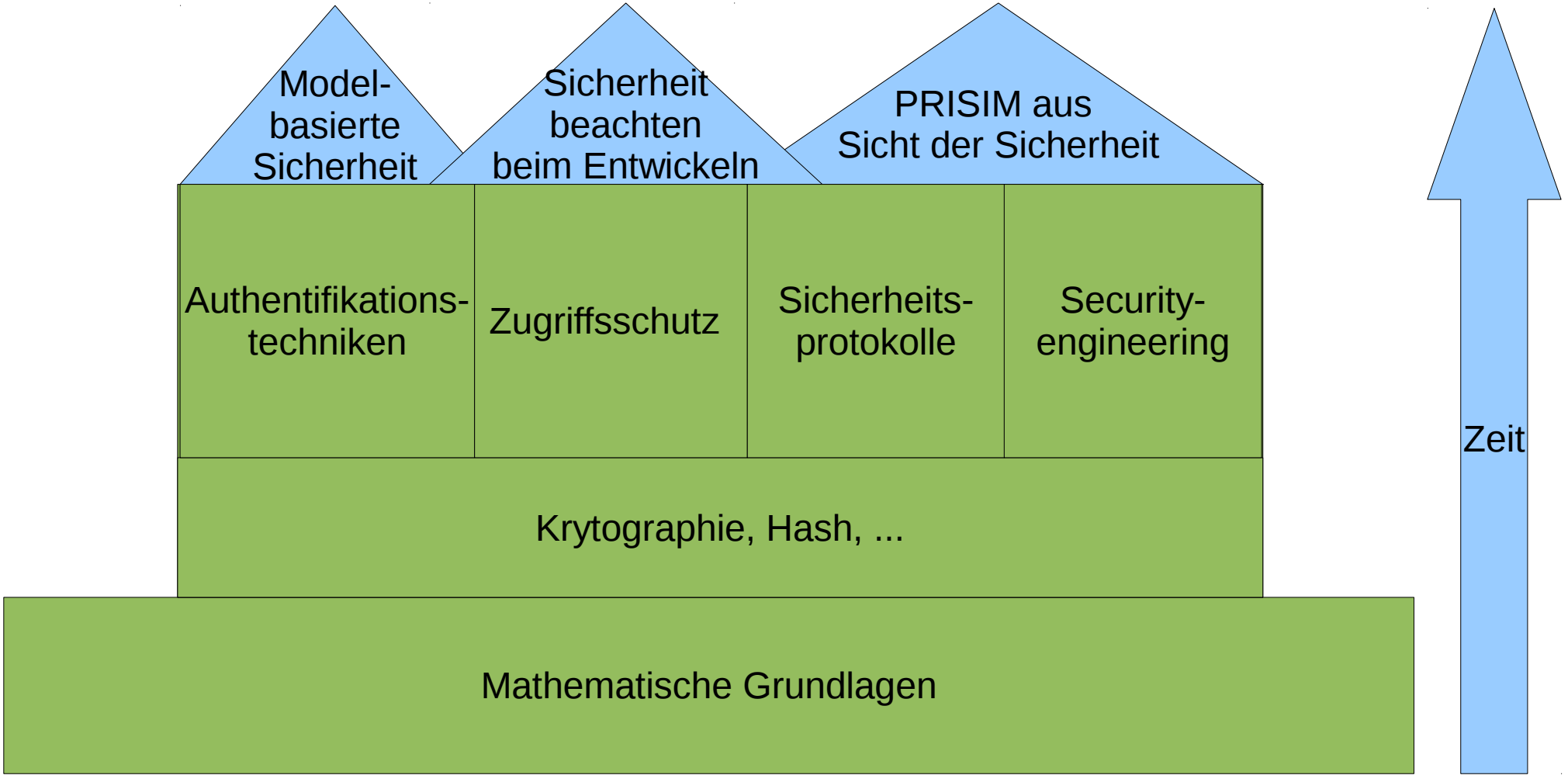
[mit freundlicher Genehmigung basierend auf einem Foliensatz von Prof. Dr. Claudia Eckert (TU München)]

Literatur:

Claudia Eckert: IT-Sicherheit: Konzept - Verfahren - Protokolle, 7., überarb. und erw. Aufl., Oldenbourg, 2012.

E-Book: http://www.ub.tu-dortmund.de/katalog/titel/1362263

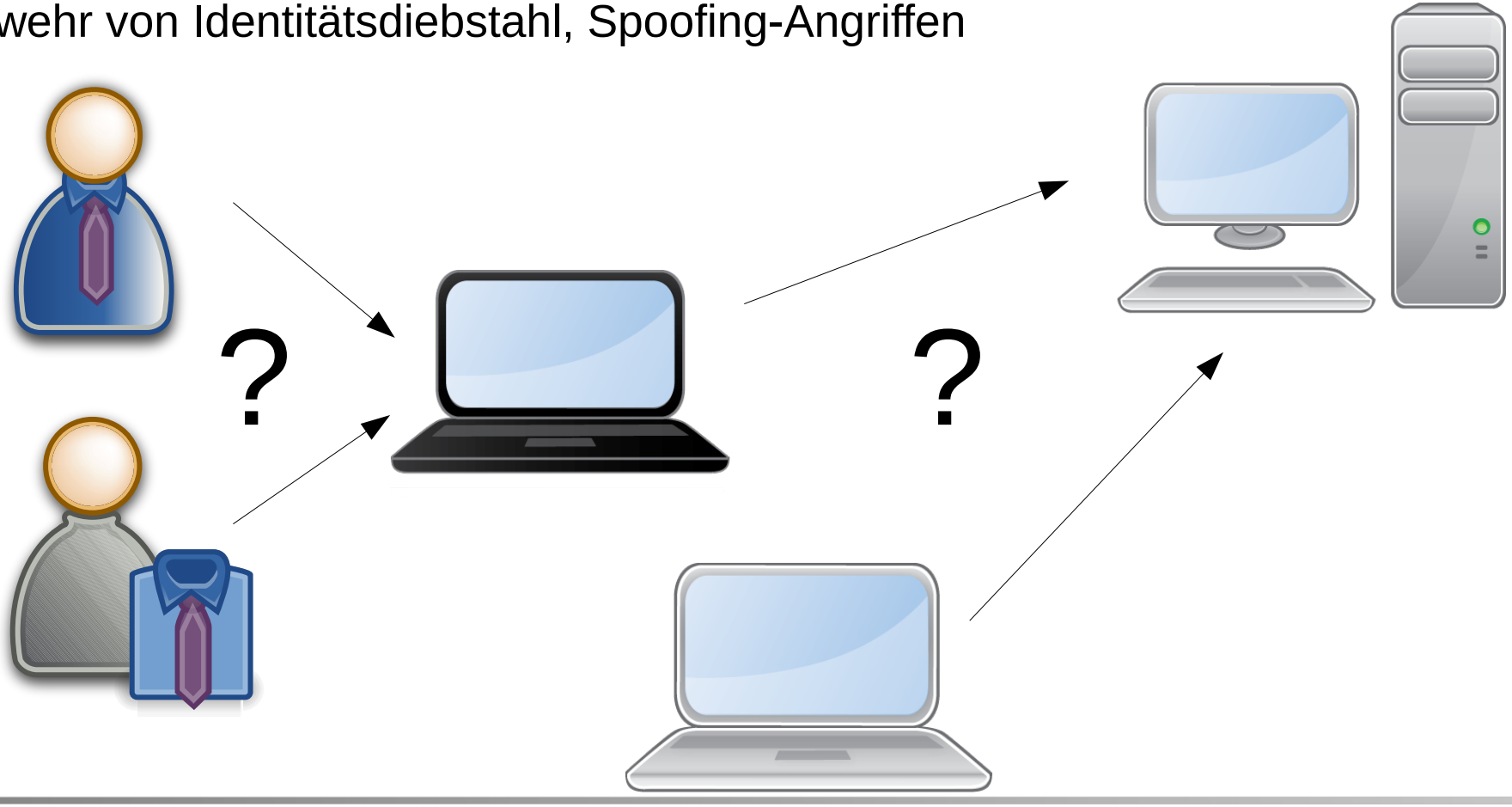

Model-
basierte
Sicherheit
Sicherheit
beachten
beim Entwickeln
PRISIM aus
Sicht der Sicherheit
Sicherheit – Grundlagen
Theorie und Umsetzung
Mathematische Grundlagen
Zeit


Model-
basierte
Sicherheit
Sicherheit
beachten
beim Entwickeln
PRISIM aus
Sicht der Sicherheit
Authentifikations-
techniken
Zugriffsschutz
Sicherheits-
protokolle
Security-
engineering
Krytographie, Hash, ...
Mathematische Grundlagen
Zeit

Authentifikation

Ziel:

- eindeutige Identifikation und Nachweis der Identität
- Abwehr von Identitätsdiebstahl, Spoofing-Angriffen



Nicht nur Mensch-zu-Gerät Interaktion, sondern auch

- Gerät-zu-Gerät (M2M)
- {Gerät, Dienst} zu {Dienst, Gerät}

  Bem.: zunehmende Vernetzung u. Miniaturisierung: M2M-Kommunikation steigt rapide an!

Notwendig: Konzepte und Verfahren, um sowohl

- menschliche Individuen eindeutig zu identifizieren
- als auch Geräte (Web-Server, Laptop, Smartphone, ...) und
- Dienste (Dateisystem, Amazon, Bankportal, ...)

Klassen von Authentifikationstechniken

Authentifikation durch

- Wissen: z.B. Passworte, PINs, kryptogr. Schlüssel
- Besitz: z.B. Smartcard, USB-Token, SIM-Karte (Handy)
- biometrische Merkmale: z.B. Fingerabdruck, Iris

Ziel:

Authentifikation eines Subjekts gegenüber einer Instanz

- Subjekt (Mensch, Gerät, Dienst, …)
- Instanz (Server, Gerät, Dienst, Mensch)

Mehrfaktor-Authentifikation

Kombination von Konzepten: z.B.

- Online-Identifizierung: PIN und nPA
- Handy: PIN und SIM-Karte

Beispiel: 2-Faktor-Authentifikation beim Handy:

(1) Authentifikation über PIN (Wissen) gegenüber SIM-Karte

(2) und Besitz der SIM-Karte (geheimer Schlüssel K_{SIM}) SIM-Karte authentifiziert sich gegenüber dem Netz mit K_{SIM}

Verfahren
-
Passwort

Passworte

- Einfachstes Verfahren
- Breite Akzeptanz
- Sicher solange Passwort geheim bleibt
- Probleme:
  - Oft einfach abhörbar, z.B.
    - Keylogging
    - Phishing
  - Oft zu einfache Passwörter gewählt

Passwortsichheit

- Regelmäßig Ändern
- Großer Passwortraum (Sonderzeichen, Zahlen etc.)
- Pro Anwendung/System anderes Passwort
- Sichere Übertragung
  - SSL/TLS (Achtung: Unsicher Verschlüsselung abschalten)
- Verhindern von Wiederholten Eingabeversuchen

Verfahren
-
Challenge-Response-Verfahren

Challenge-Response-Verfahren (CR)

Idee:

- Subjekt gibt seine Identität an: z.B. Name, IMSI, MAC-Adr.
- Instanz sendet eine Challenge (idR Zufallszahl) zum Subjekt
- Subjekt berechnet Response (z.B. mittels Verschlüsselung)
- Instanz prüft Response, falls korrekt, dann hat Subjekt ein geheimes Wissen (z.B. Schlüssel) nachgewiesen

Symmetrische CR-Verfahren

Symmetrisches CR-Verfahren

Ziel: Subjekt *A* authentifiziert sich gegenüber Instanz *B*

Basis: vorab **geheimer Schlüssel** K_{ID} *(pre-shared Secret)*

Subjekt A:		Instanz B:
Schlüssel K_{ID}, Identifikation *ID*		Schlüssel K_{ID} zu *ID*
z.B. Login:	(1) ID →	Erzeugen von *RAND*: das ist die Challenge
	← (2) RAND	
$E(RAND,K_{ID})=C$	(3) C →	$E(RAND,K_{ID})=C'$
C(=Response)		Test: *C'*=*C*?

Mögliche Problembereiche/Schwachstellen?

Verfahren
-
Einmal-Passworte

Einmal-Passworte

- Idee:
  - Für jede Authentifikation wird ein neues Passwort genutzt
- Generierung:
  - Token (ID-Token)
  - Vorgenerierte Liste (TAN-Liste)
  - Zusenden auf separaten Kanal (eTAN)

Hardware-basierte OTP-Verfahren

Hardware-basierte OTP-Verfahren: ID-Token

- OTP-Verfahren zur Authentifikation beim Server
- Benutzer erhält ein Hardware-Token
- Token besitzt eindeutige Nummer; Server kennt diese

Beispiel RSA SecureID-Token

- Admin des Servers richtet Benutzer-Account ein, mit:
  - Token-Nummer und 128-Bit Seed *s* *(früher nur 64 Bit)*
  - Seed *s* wird auch auf RSA-Token gespeichert

Erzeugen von OTPs:

- alle 60 Sekunden generieren Token u. Server neues Passwort

  AES-Hashwert: Tokencode = AES(*TokenId* | *s* | Zeit)

Validierung eines Tokencodes durch Server:

- Es werden die nächsten 3-5 Token zugelassen

S/Key Verfahren

S/Key Verfahren

Set-up: Benutzer, lokaler PC, entfernter Server

- Benutzer: besitzt geheimes Benutzer-Passwort *s*,
- *s* ist Pre-Shared Secret zwischen Benutzer und PC,
- Server kennt *s* **nicht**

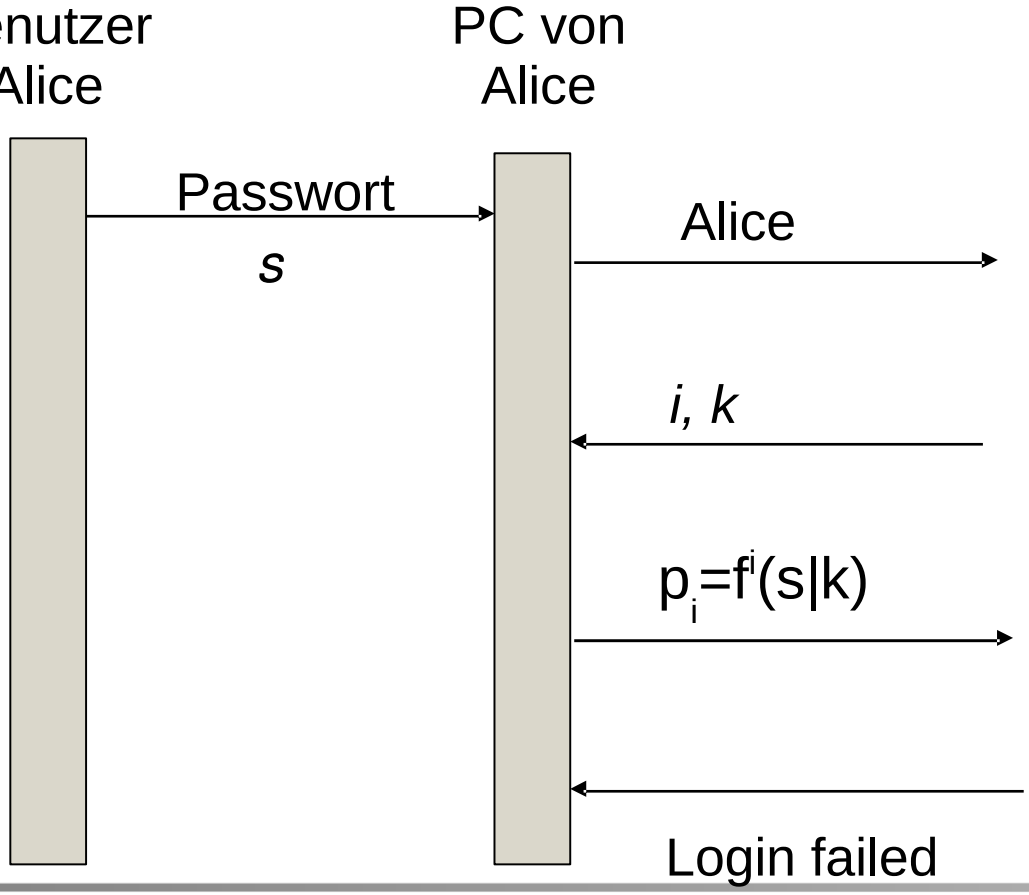
2 Phasen: (1) OTP-Berechnung, Initiierung und (2) Nutzung

(1) OTP-Berechnung

- PC berechnet aus *s* einmal benutzbare Passworte p_i
- dazu notwendig:
  - kryptographische Hashfunktion *f* (z.B. SHA)
  - Wahl einer Zahl *N*, Wahl eines Seed-Wertes *k*
  - OTPs: $p_1 = f(s|k)$, $p_2 = f(p_1)$, ..., $p_N = f^N(s|k)$

- Übertragen der Startwerte p_N , N, Seed-Wert k und Nutzer-Kennung an Login-Server

(2) Nutzung: Ablauf: *i*-te Authentifikation des Benutzers



Login-Server **hat gespeichert**:
p_{j+1}, i+1, Seed k

Kontrolle:
$f(p_i)=p_{i+1}$?
Falls ja:
Speichere: p_i, i
Anderenfalls: Failed

Verfahren
-
Zero-Knowledge-Verfahren

Zero-Knowledge-Verfahren

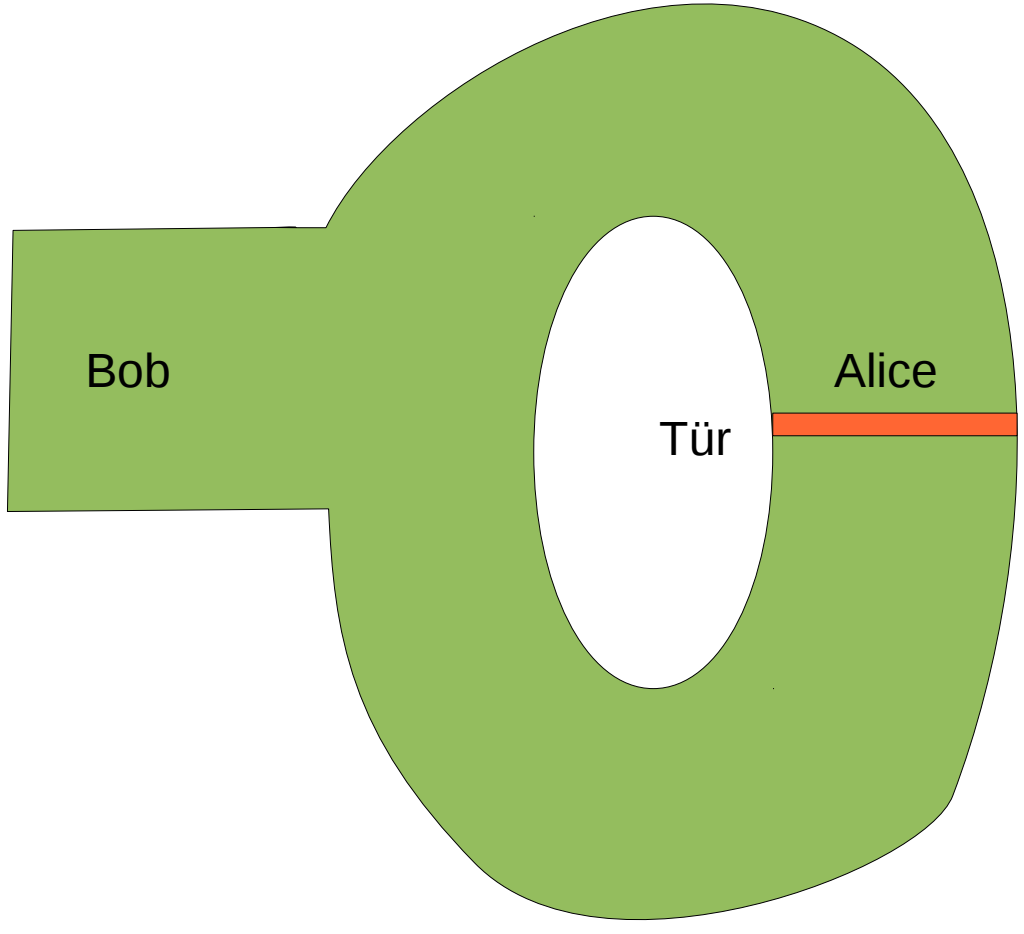
Idee: Nachweis der Kenntnis eines Geheimnisses gegenüber einem Dritten (hier Bob),

- ohne dass Bob das Geheimnis kennt und
- ohne dass Bob im Verlauf der Authentisierung Kenntnis über das Geheimnis erlangt
- Angreifer darf beliebig viele Nachrichten belauschen

Idee Zero-Knowledge

-

- Bob sieht nicht in welchen Gang Alice geht
- Bob bitte Alice durch einen bestimmten Gang herauszukommen
- Wenn Sie das Wissen hat wie die Tür geöffnet wird, kommt Sie immer aus dem richtigen Gang
- Wenn Sie nicht weiß wie die Tür zu öffnen ist, kommt Sie in 50% der Fälle durch den falschen Gang



Eigenschaften

- Vollständigkeit
  Wenn Alice das Wissen besitzt, soll fast immer akzeptiert werden.
- Zuverlässigkeit
  Wenn Alice das **nicht** Wissen besitzt, soll fast immer abgelehnt werden.
  - Dabei ist eine geringe Fehlerwahrscheinlichkeit erlaubt.
- Zero-Knowledge-Eigenschaft
  - Aus der Interaktion darf nicht mehr Wissen als die (Un-)Gültigkeit der zu beweisenden Aussage gewonnen werden.
  - Ein Dritter, der die Interaktion verfolgt, erfährt nicht einmal, ob der Beweiser überhaupt das Geheimnis kennt (oder die Interaktion zwischen B und V abgesprochen war).

Zero-Knowledge-Verfahren

Fiat-Shamir-Verfahren (1987): eines der ersten ZK-Verfahren

- Sicherheit beruht auf der Schwierigkeit, Quadratwurzeln in Z_n^* zu berechnen:

  Gegeben: $n = p \cdot q$, $x = r^2 \bmod n$, Gesucht ist : r

- Einfach, falls Primfaktoren p,q bekannt, sonst schwierig

Verfahren teilt sich auf in **2 Phasen**:

(1) Schlüsselerzeugung

(2) Anwendung

Schlüsselerzeugung

Teilnehmer A: führt Vorbereitungsschritte durch

- Erzeugt zwei große Primzahlen p, q, sind geheim
- Berechnet: $n = p \cdot q$, n ist öffentlich
- Wählt s und berechnet: $v = s^2 \bmod n$, v ist öffentlich
- s ist individuelles Geheimnis von A
- mit v ist verifizierbar, ob jemand das Geheimnis s kennt

Zero-Knowledge-Beweis (Feige-Fiat-Shamir-Protokoll)

Ziel: A(lice) will B davon überzeugen, dass sie *s* kennt

Protokollablauf:

A:

- A wählt zufälliges ***r*** aus Z_n^* und berechnet: $\boldsymbol{x} = \boldsymbol{r}^2 \bmod n$
- A sendet Wert ***x*** an B

B: B wählt zufälliges Bit ***b*** und sendet *b* an A

- A antwortet auf diese Challenge mit
  - $y = r$, falls $b = 0$ und mit $y = r \cdot s \bmod n$, falls $b = 1$
- B prüft diese Antwort von a wie folgt:
  - Falls $b = 0$, prüft B, ob gilt: $\boldsymbol{x} = y^2 \bmod n$
  - Falls $b = 1$, prüft B, ob gilt: $\boldsymbol{x} \cdot v \bmod n = y^2 \bmod n$
- *k*-malige Challenge durch B mit neuem *b* und *r*

Warum funktioniert das?

Warum funktioniert das als Authentisierungsnachweis?

- Antwort: in Z_n^* gilt:

  $y^2 = (r \cdot s^b)^2 = r^2 s^{2b} = r^2 v^b = x v^b$ *(alles* mod *n)*

  D.h. A kann B davon überzeugen, *s* zu kennen

Mögliche Angriffe?

Verfahren
-
biometrischer Merkmale

Biometrie

Biometrisches Merkmal: Bios = Leben, metron = Maß

Verhaltenstypische oder physiologische Eigenschaft eines Menschen, die diesen eindeutig charakterisieren.

Anforderungen an biometrische Merkmale:

- **Universalität**: Jede Person besitzt das Merkmal
- **Eindeutigkeit**: Merkmal ist für jede Person verschieden
- **Beständigkeit**: Merkmal ist unveränderlich
- quantitative **Erfassbarkeit** mittels Sensoren
- **Performance**: Genauigkeit und Geschwindigkeit
- **Akzeptanz** des Merkmals beim Benutzer
- **Fälschungssicherheit**

Klassen biometrischer Merkmale

Unterschiede zur wissensbasierten Authentisierung:

- Merkmal ist personengebunden: Konsequenz?
- Charakteristische Merkmale müssen extrahiert und mit Referenzwert verglichen werden: Probleme?

Klassen biometrischer Merkmale

- physiologische Merkmale (statisch): keine oder nur sehr begrenzte Möglichkeiten zur Auswahl oder Änderung von Referenzdaten
- Verhaltensmerkmale (dynamisch): Merkmal ist nur bei bestimmter Aktion vorhanden; Möglichkeiten zur Auswahl/Änderung von Referenzdaten

Beispiele für dynamische biometrische Merkmale:

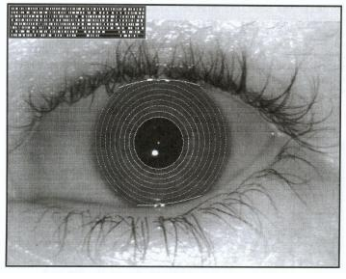
Unterschriften-Dynamik, Sprache, Tippverhalten (Keystroke).

Beispiele für statische biometrische Merkmale:

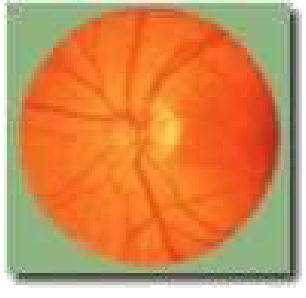
Fingerabdruck, Gesichtsbild, Handgeometrie, Retina Venenmuster, Iris.

Wirklich statisch?

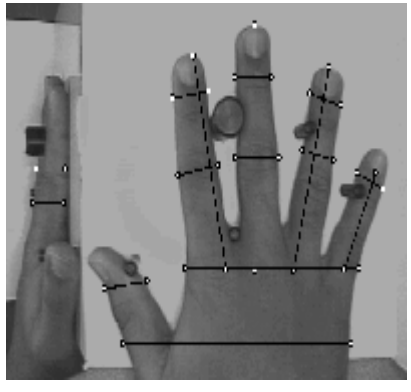
Venenmuster



Iris



Retina



Handgeometrie

Vorgehen bei biometrischer Authentifikation

1. Messdatenerfassung durch biometrischen Sensor und Digitalisierung (Feature-Extraction)
2. Enrollment: Registrierung eines Benutzers: Aufnahme, Auswahl und Speicherung der Referenzdaten z.B. 5 bis 7 verschiedene Fingerabdruck-Werte
3. Bei Authentifikation: Erfassung der aktuellen Verifikationsdaten (mittels Sensoren)

4. Verifikationsdaten digitalisieren (u.a. ggf. normieren)
5. mit gespeichertem Referenzwert vergleichen, Toleranzschwellen sind notwendig

Problembereiche

- Abweichungen zwischen Referenz- und Verifikationsdaten sind unvermeidlich,
- Zwei Fehlertypen:
  - Berechtigter Benutzer wird abgewiesen,

    ⇒ Akzeptanzproblem (false negative)
  - Unberechtigter wird authentifiziert, Kontrollen zu locker

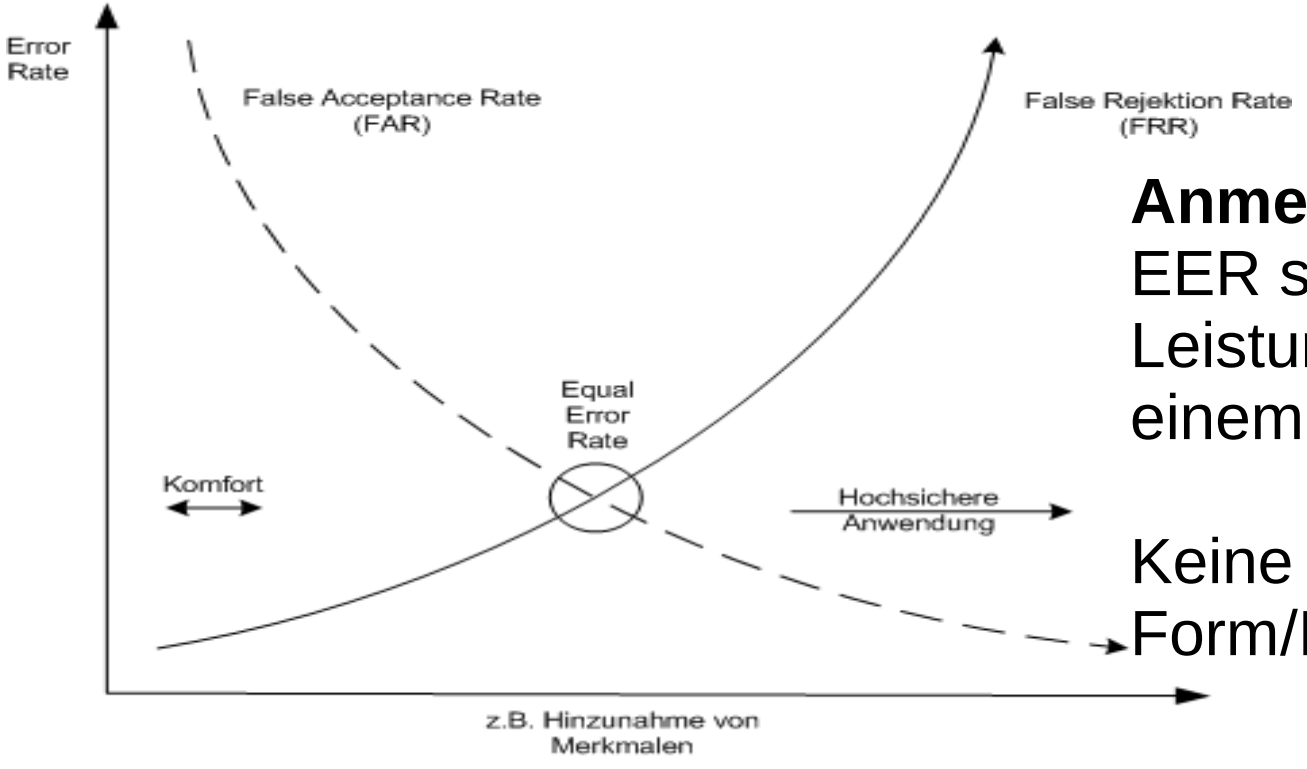
    ⇒ Sicherheitsproblem (false positive, false accept)
- Leistungsmaße zur Bewertung der Güte eines Systems
  - **False-Acceptance-Rate (FAR)**: Wahrscheinlichkeit für fälschliche Akzeptanz einer unberechtigten Person (false accepts)

- **False Rejection Rate** (FRR):

  Wahrscheinlichkeit für fälschliche Rückweisung einer berechtigten Person

- **Equal Error Rate** (EER):

  Gleichfehlerrate Gemeinsamer Wert für FAR und FRR



Anmerkung:
EER sagt etwas über die Leistungen des Systems in einem bestimmten Punkt aus.

Keine Aussage über Form/Entwicklung der Kurven.

Beispiel: Fingerabdruck

Beispiel: Fingerabdruck

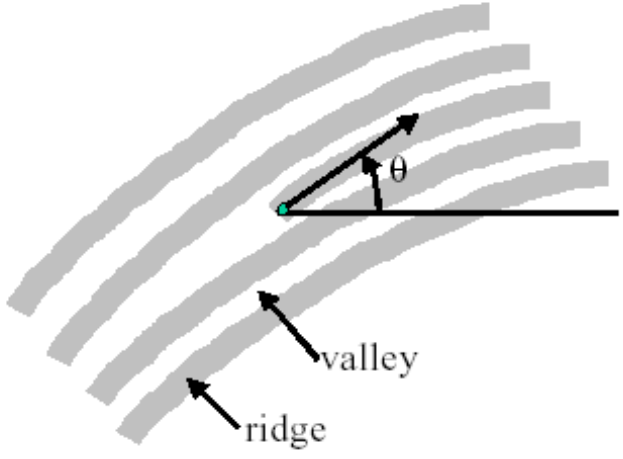
Sensor

Graustufen-Bild
eines Fingerabdrucks

Merkmal: Minutien (lat. für Kleinigkeit, Detail):

End- und Verzweigungspunkte, Wirbel, Tälern mit Ortskoordinaten und Tangentenwinkel.

Muster von Rillen (ridge) und Tälern
ist charakteristisch für jeden Mensch.



Verarbeitung von Fingerabdrücken

Rillenmuster nach dem Herausfiltern des Hintergrunds, z.B. Filtern von Veränderungen durch Schmutz, Verletzungen etc.

Feature-Extraktion: Minutienbestimmung als Endpunkte u. Verzweigungswinkel, Richtungen, idR 30-60 Minutien, Danach: Abgleich des Minutien-Musters mit Referenzwerten: Vergleich (match) benachbarter Minutien

Verifikation: One-to-one mit gespeichertem Referenzwert
Identifikatikon One-to-Many: Suche nach Referenzwert

Häufig betrachtete Merkmale für Fingerabdruck-Minutien

- Ortskoordinaten *x* und *y*
- Linienwinkel (Tangentenwinkel) θ

Darstellung einer Minutie als Tripel (x, y, θ)

Darstellung eines Fingerabdrucks als Folge von Tripeln:

$((x_1, y_1, \theta_1), (x_2, y_2, \theta_2), \ldots, (x_n, y_n, \theta_n))$

Einfacher Algorithmus zum Minutienvergleich

- Abstand und Winkeldifferenz zweier Minutien

  $(x_i, y_i, \theta_i), (x_j, y_j, \theta_j)$: $d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$

$$\Delta\theta = \begin{cases} |\theta_i - \theta_j|, \; falls \; |\theta_i - \theta_j| \leqslant 180^\circ \\ 360^\circ - |\theta_i - \theta_j|, \; falls \; |\theta_i - \theta_j| > 180^\circ \end{cases}$$

Vergleichskriterium für zwei Fingerabdrücke

- Wähle Toleranzschwellen $\boldsymbol{d}_{\mathbf{Tol}}$ und $\theta_{\mathbf{Tol}}$ und einen
- „Match-Score“ ***k*** abhängig vom gew. Sicherheitsniveau.
- Zwei Minutien gelten als übereinstimmend, falls $\boldsymbol{d} \leq \boldsymbol{d}_{\mathbf{Tol}}$ und $\Delta\theta \leq \theta_{\mathbf{Tol}}$.
- Zwei Fingerabdrücke gelten als übereinstimmend, wenn mindestens ***k*** übereinstimmende Minutien im Rahmen der Toleranzen $\boldsymbol{d}_{\mathbf{Tol}}$ und $\theta_{\mathbf{Tol}}$ gefunden wurden.

Sicherheitsprobleme bei biometrischen Techniken

Angriffsstrategien:

- Direkte Täuschung des biometrischen Sensors durch Attrappen, u.a. Gummi-Finger
- Einspielen von Daten unter Umgehung des biometrischen Sensors
  - Wiedereinspielen abgehörter Daten (Replay-Angriffe)
  - Einspielen eigens verschaffter, digitalisierter Daten

Probleme: enge Kopplung zwischen Merkmal und Person

(1) Bedrohung der informationellen Selbstbestimmung

(2) Gefahren durch gewaltsame Angriffe gegen Personen

(3) Problem der öffentlichen Daten und rechtliche Aspekte

Nächste Woche

Authentifikationsprotokolle