Westfälische Hochschule
Gelsenkirchen Bocholt Recklinghausen
University of Applied Sciences

Identifikation und Authentifikation

- Vorlesung Cyber-Sicherheit -

Prof. Dr. (TU NN)
Norbert Pohlmann

Institut für Internet-Sicherheit – if(is)
Westfälische Hochschule, Gelsenkirchen
http://www.internet-sicherheit.de

Identifikation und Authentifikation
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Identifikation und Authentifikation
- Generelle Authentifikationsverfahren
- Passwort-Verfahren
- Einmal-Passwort-Verfahren
- Challenge-Response-Verfahren
- Biometrische Verfahren
- Mehrfaktor-Authentifizierung
- Moderne Authentifizierungssysteme
- FIDO
- Zusammenfassung

Identifikation und Authentifikation
→ Inhalt

- **Ziele und Ergebnisse der Vorlesung**
- Identifikation und Authentifikation
- Generelle Authentifikationsverfahren
- Passwort-Verfahren
- Einmal-Passwort-Verfahren
- Challenge-Response-Verfahren
- Biometrische Verfahren
- Mehrfaktor-Authentifizierung
- Moderne Authentifizierungssysteme
- FIDO
- Zusammenfassung

Ziele und Ergebnisse der Vorlesung
→ Identifikation und Authentifikation

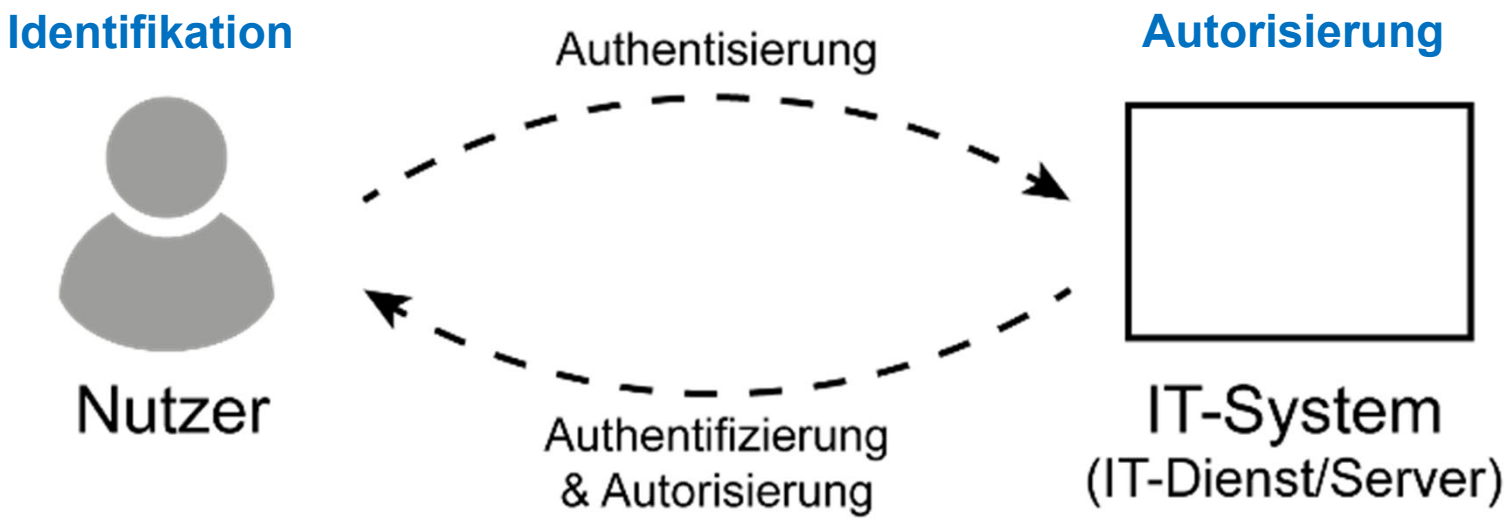
- Gutes **Verständnis** zu der Bedeutung von **Identifikation** und **Authentifikation** im **Cyber-Raum**.
- Gutes **Verständnis für** die aktuellen **Identifikation-** und **Authentifikationsmechanismen** und **-methoden**.
- Erlangen der Kenntnisse über prinzipielle **Identifikation-** und **Authentifikationsverfahren** und zur Umsetzung von konkreten **Lösungen**.

Identifikation und Authentifikation
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- **Identifikation und Authentifikation**
- Generelle Authentifikationsverfahren
- Passwort-Verfahren
- Einmal-Passwort-Verfahren
- Challenge-Response-Verfahren
- Biometrische Verfahren
- Mehrfaktor-Authentifizierung
- Moderne Authentifizierungssysteme
- FIDO
- Zusammenfassung

Identifikation und Authentifikation
→ Das Problem

Wer ist tatsächlich der Nutzer und welche Rechte hat er?



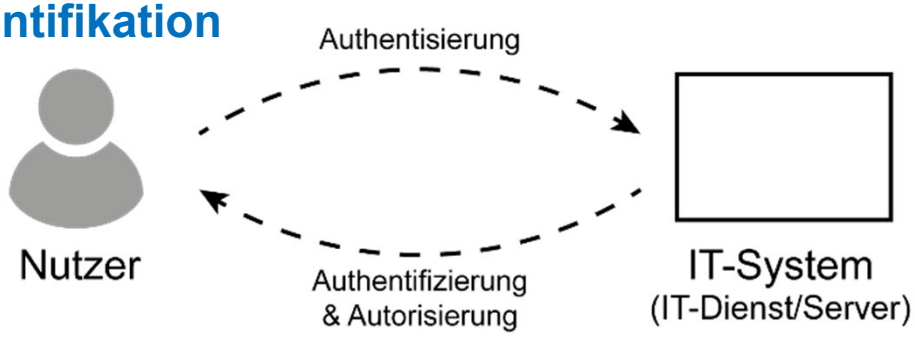
- Wenn ein Nutzer Zugang haben möchte, muss er sich dem IT-System gegenüber identifizieren und authentisieren.
- Das IT-System muss den Nutzer erst authentifizieren und anschließend autorisieren.

Identifikation und Authentifikation
→ Identifikation (1/2)

- Die Identifikation ist der Nachweis oder die Überprüfung einer behaupteten Eigenschaft einer Identität.
  - Nutzer identifiziert sich gegenüber einem IT-System (z.B. Nutzername).
  - IT-System überprüft die behauptete Eigenschaft.

- In Deutschland wird die Eindeutigkeit der Identifikation von den Standesämtern garantiert.
  - Eine Person wird eindeutig durch die Angabe von Vorname, Nachname, Geburtsort und Geburtstag identifiziert.

Identifikation

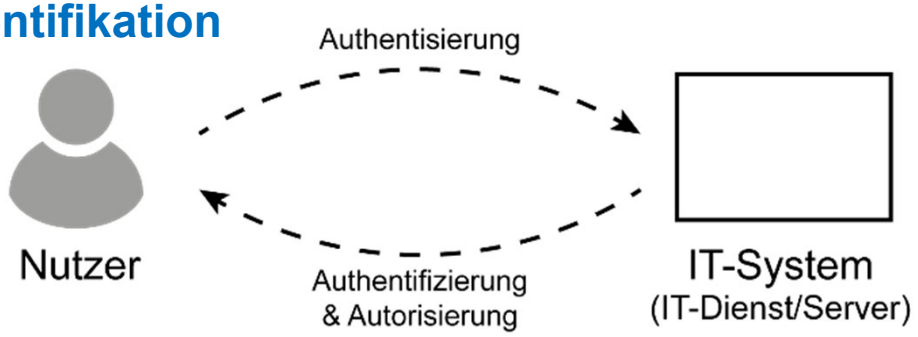


Identifikation und Authentifikation
→ Identifikation (2/2)

- Eine Identifikation muss immer innerhalb eines Systems (Organisation) abgesprochen sein, damit sie eindeutig ist.
  - Damit eine solche Absprache mit verschiedenen Nutzern zustande kommt, müssen klar definierte Regeln eingehalten werden.

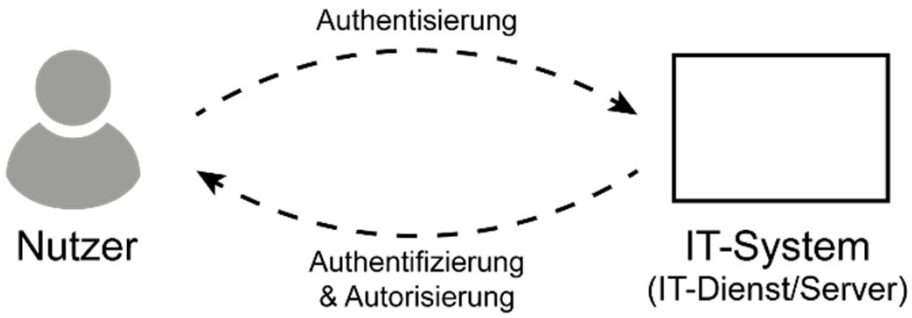
- Beispiel:
  - CCITT »Recommendation« X.509 bzw. ISO 9594-8 (Ein Konzept eindeutiger, kennzeichnender Namen oder »distinguishing identifier«)

Identifikation



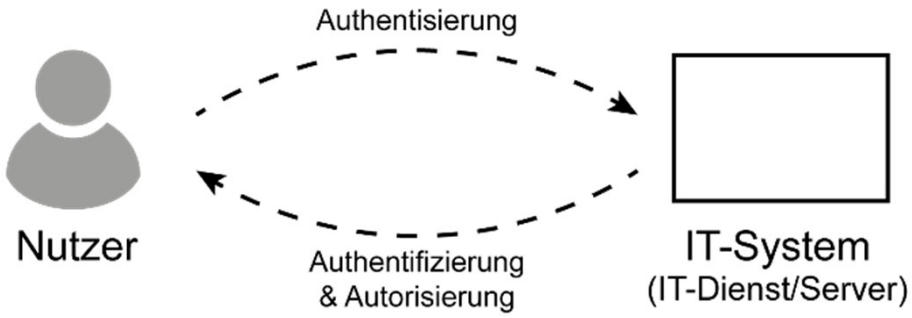
Identifikation und Authentifikation
→ Authentisierung

- Die Authentisierung ist der Nachweises darüber, dass ein Kommunikationsteilnehmer tatsächlich die behauptete Identität besitzt/repräsentiert.
  - In der Praxis werden hierfür verschiedene Identifikationsverfahren verwendet.



Identifikation und Authentifikation
→ Authentifizierung

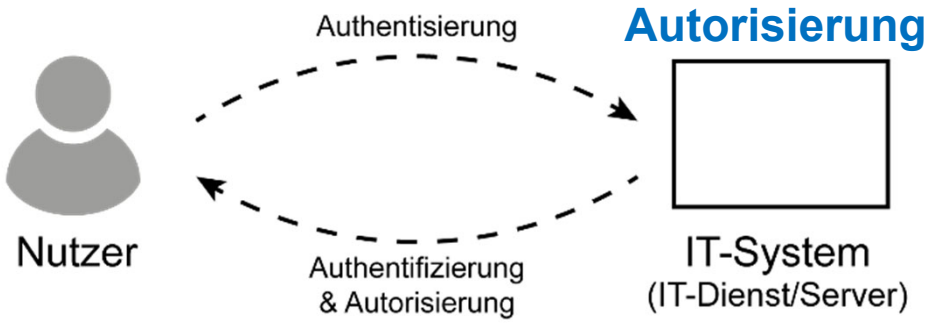
- Authentifizierung bezeichnet einen Prozess in dem überprüft wird, ob eine behauptete Identität echt oder berechtigt ist.
  - In der Praxis werden hierfür verschiedene Authentifikationsverfahren verwendet.
  - Die Überprüfung des Personalausweises einer Person ist eine solche Authentifizierung.



Identifikation und Authentifikation
→ Autorisierung

- Bei der Autorisierung werden dem authentifizierten Kommunikationsteilnehmer vorher definierte Rechte für die Ressourcen des IT-Systems eingeräumt.

- Was muss und kann z.B. identifiziert, authentisiert, authentifiziert und autorisiert werden?
  - Kommunikationspartner: z.B. Nutzer, Prozesse, Instanzen, das Security Management
  - Kommunikationsmedien: z.B. Workstation, Serversysteme, Firewall-Elemente (Packet Filter, Application Gateway, Proxy, Security Management), Security Token usw.
  - Nachrichten: z.B. Mails, Dateien, Java-Applets usw.



Identifikation und Authentifikation
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Identifikation und Authentifikation
- **Generelle Authentifikationsverfahren**
- Passwort-Verfahren
- Einmal-Passwort-Verfahren
- Challenge-Response-Verfahren
- Biometrische Verfahren
- Mehrfaktor-Authentifizierung
- Moderne Authentifizierungssysteme
- FIDO
- Zusammenfassung

Generelle Authentifikationsver.
→ Übersicht (1/3)

Wissen

Besitz

Biometrie

Generelle Authentifikationsver.
→ Übersicht (1/2)

- **Passwort-Verfahren**
  - Einfachste Authentifikationsverfahren
  - Wenn das Passwort im Klartext über das Internet übertragen wird, dann kann es mitgelesen und missbräuchlich verwendet werden
  - Passwortregeln müssen eingehalten werden
- **Einmal-Passwort**
  - Jedes Passwort wird nur einmal verwendet
  - Zwei unterschiedliche Methoden:
    - Passworte werden im Vorfeld bestimmt und verteilt (z.B. TAN-Listen)
    - Nutzer kann sie nach einem definierten Verfahren berechnen

Generelle Authentifikationsver.
→ Übersicht (2/2)

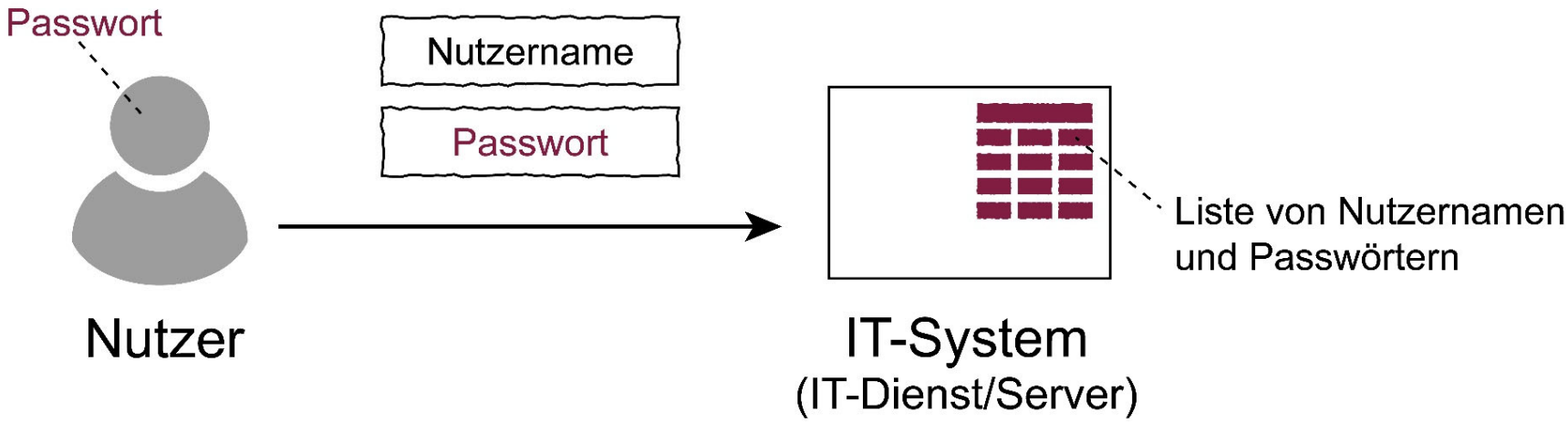
- **Challenge-Response-Verfahren**
  - Nutzer muss sich spontan kryptographisch beweisen
  - Dazu braucht er einen Schlüssel und ein Verfahren
  - Z.B. Zufallszahl als Challange, Signatur dieser als Response
- **Biometrische Verfahren**
  - Identifikation und Authentifikation mittels biometrischer Merkmale
    - Aktiv: Stimme, Unterschrift, Gestik, Tippverhalten
    - Passiv: Fingerabdruck, Retina, Iris, Gesicht, Ohr
  - Zur Authentifikation im Internet kaum anwendbar
  - Nutzbar als Zugangskontrolle (Pässe, Türen, USB-Token)

Identifikation und Authentifikation
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Identifikation und Authentifikation
- Generelle Authentifikationsverfahren
- **Passwort-Verfahren**
- Einmal-Passwort-Verfahren
- Challenge-Response-Verfahren
- Biometrische Verfahren
- Mehrfaktor-Authentifizierung
- Moderne Authentifizierungssysteme
- FIDO
- Zusammenfassung

Passwort-Verfahren
→ Übersicht (1/2)

- Passwörter sind das einfachste, prinzipiell unsicherste, aber meist verwendete Authentifizierungsverfahren im Internet.
- Nutzer registriert im Vorfeld ein Nutzername und Passwort bei einem IT-System.
- Die Authentifikation ist der Nachweis des Wissens über den registrierten Nutzernamen und das Passwort.
  - IT-System verwaltet dazu eine Liste von Nutzernamen und Passwörtern.



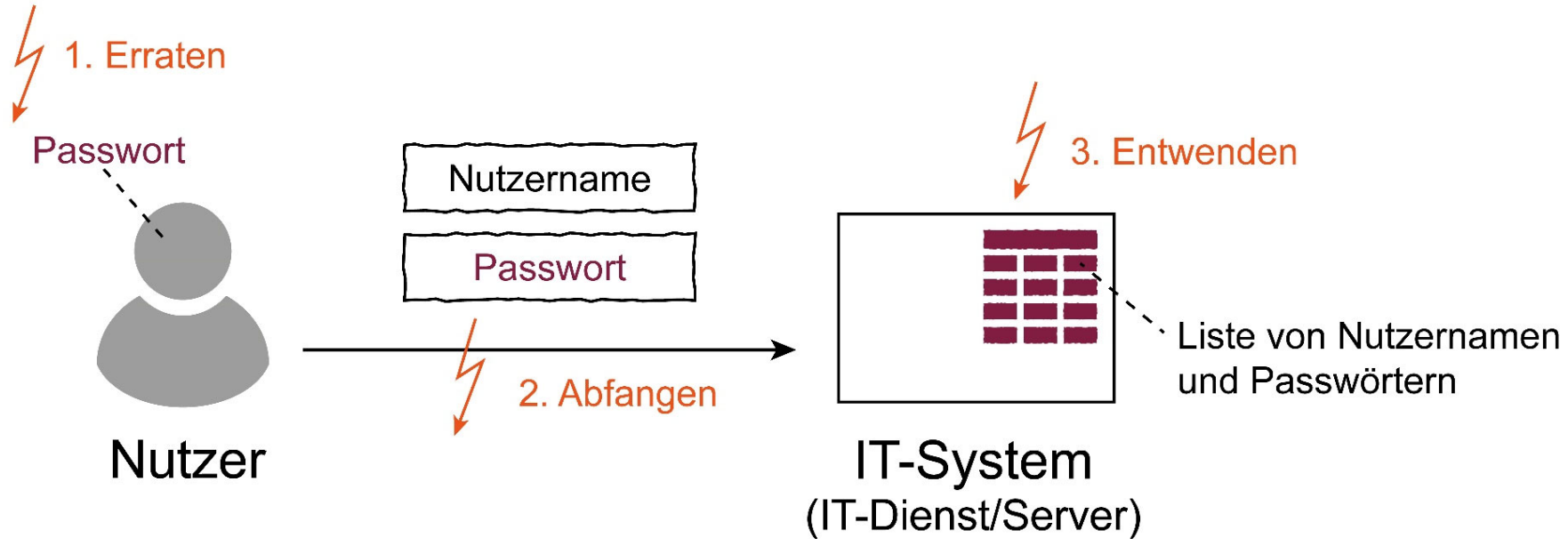
Passwort-Verfahren
→ Übersicht (2/2)

- Die Stärke dieses Authentifizierungsverfahrens beruht letztlich auf der Geheimhaltung und der Qualität des Passwortes.

- Eine besondere Schwäche des Passwortverfahrens ist unteranderem, dass gestohlene Zugänge nicht direkt als Angriff erkannt werden können.

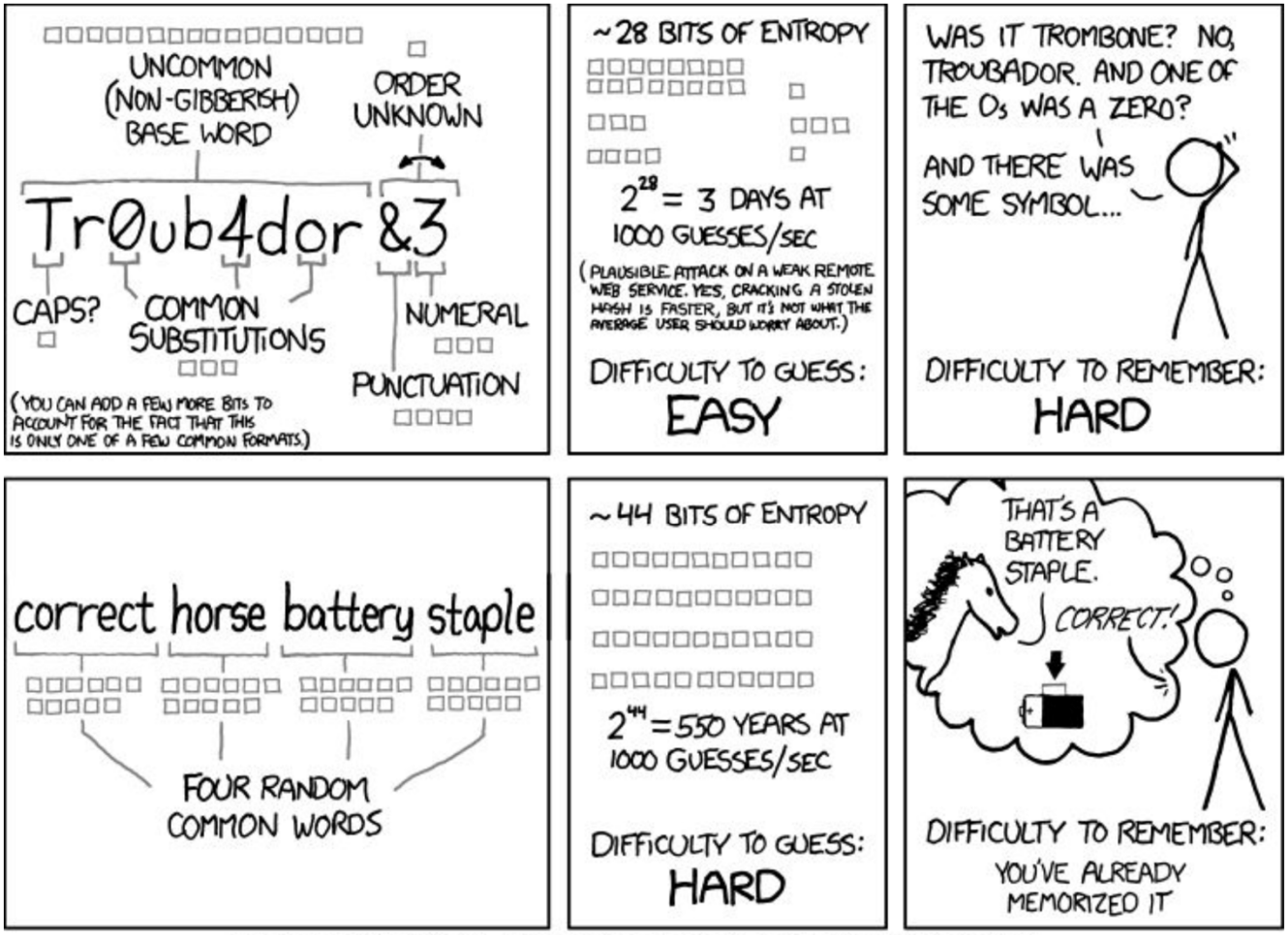
Passwort-Verfahren
→ Angriffsvektoren

- Erraten des Passwortes durch Social Engineering oder Ausprobieren.
- Abfangen von Nutzername und Passwort während der Eingabe oder Übertragung.
- Entwenden der Liste von Nutzernamen und Passwörtern auf dem IT-System.



Passwort-Verfahren
→ Angriffsvektor: Erraten

- Wir haben „gelernt“ Passwörter zu Erstellen, die sich Menschen nur schwer merken können aber Maschinen leicht raten können.



Passwort-Verfahren
→ Angriffsvektor: Erraten

- Der Angreifer kann unterschiedliche Angriffsmethoden durchführen:
  - Der Angreifer kann mit Hilfe von Social Engineering das Umfeld einer Person ausspionieren und mit den Informationen auf das Passwort schließen (z.B. Fußballfan in Gelsenkirchen = Passwort „Schalke04“).
  - Durch Phishing entstehen große Schäden (BKA).
  - Er kann Passwörter, die oft genutzt werden, ausprobieren (Wörterbuchangriff).
  - Er kann durch systematisches Durchprobieren aller möglichen Kombinationen (Brute-Force-Angriff) das richtige Passwort ermitteln.

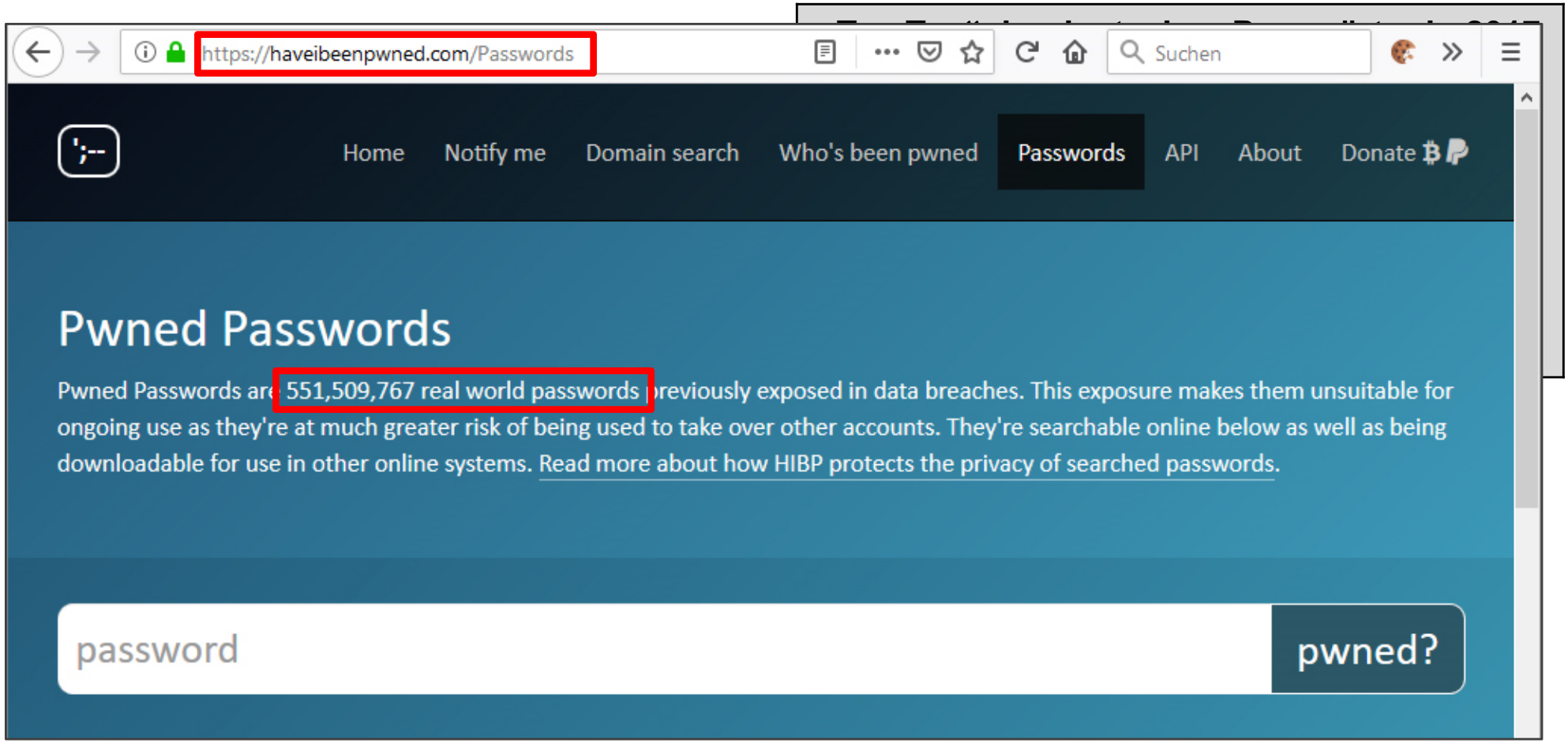
Passwort-Verfahren
→ Wörterbuchangriff (1/2)

- Idee:
  - Passwörter mit Hilfe einer Passwörterliste ermitteln.
  - In der Passwortliste stehen Passwörter (Wörter, Phrasen, Zeichenketten, …), die oft von Personen genutzt werden (z.B. Schalke04, Porsche911, Sabine906090, …).

- Voraussetzung:
  Gesuchtes Passwort besteht aus einer vorhersagbaren Zeichenkombination.
  - Anzahl der möglichen Passwortkandidaten wird ggf. eingeschränkt → Schnellere Suche möglich.
  - Nutzer im Internet verwenden dieselben unsicheren Passwörter.
  - Passwortlisten können durch zusätzliche sinnvolle Muster erweitert werden (z.B. Schalke → Schalke1234 → Schalke123456789, …).

Passwort-Verfahren
→ Wörterbuchangriff (2/2)

- Passwortlisten bestehen häufig aus geklauten/geleakten Datensätzen von Onlineplattformen.



Passwort-Verfahren
→ Brute-Force-Angriff (1/2)

- Idee:
  - Jede mögliche Kombination eines Passwortes automatisiert ausprobieren.
  - Ohne geeignete Sicherheitsmechanismen führt dieser Angriff immer zum Erfolg.

- Voraussetzung:
  - Risiko hängt von den Ressourcen eines Angreifers ab (z.B. AWS p3.16xlarge für 25 $/h → SHA-256: 59971.8 MH/s, spezielle Hardware der NSA, …).

Passwort-Verfahren
→ Brute-Force-Angriff (2/2)

- Annahme: 1 Millionen Versuche in der Sekunde möglich.

Verwendetes Alphabet	Anzahl der möglichen Zeichen	Länge des Passwortes	Anzahl der möglichen Kombinationen (vollständiger Schlüsselraum)	Zeit der vollständigen Suche
0-9	10	6	10^6 = 1.000.000	1 Sekunde
		8	10^8 = 100.000.000	100 Sekunde
		10	10^{10} = 10.000.000.000	2,8 Stunden
A-Z, a-z, 0-9	62	6	62^6 = 56.800.235.584	0,66 Tage
		8	62^8 = 218.340.105.584.896	6.9 Jahre
		10	62^{10} = 839.299.365.868.340.224	26.614 Jahre
A-Z, a-z, 0-9 ()[]{}?!$%&/= *+~,.;:<>-_	86	6	86^6 = 404.567.235.136	4,68 Tage
		8	86^8 = 2.992.179.271.065.856	94,9 Jahre
		10	86^{10} = 22.130.157.888.803.070.976	701.743 Jahre

$$Mögliche\ Kombinationen = Zeichenanzahl^{Passwortlänge}$$

Passwort-Verfahren
→ Schutzvorkehrung: Passwortregeln (1)

- Das Passwort nirgends notieren und niemandem mitteilen (Verhinderung eines Social-Engineering-Angriffes)
- Das Passwort darf nur dem Nutzer bekannt sein (Verhinderung, dass jemand anders mit dem Passwort zugreifen kann)
- Mindestlänge: zehn Stellen, besser zwölf Stellen (Verhinderung des Brute-Force-Angriffes)
- Es sollen Klein- und Großbuchstaben in Kombination mit Zahlen und Sonderzeichen verwendet werden (Verhinderung des Brute-Force-Angriffes)

Passwort-Verfahren
→ Schutzvorkehrung: Passwortregeln (2)

- Die verwendeten Zeichen sollen auf den ersten Blick eine sinnlose Zusammensetzung sein
  (Verhinderung des Wörterbuchangriffes)

- Ein Passwort nur für einen Dienst verwenden
  (Verhinderung, dass der Diebstahl eines Passwortes die Sicherheit alles IT-Dienste betrifft)

- In angemessenen Zeitabständen ändern
  (Verhinderung des Brute-Force-Angriffes)

Passwort-Verfahren
→ Passwortlänge: Empfehlungen (1/4)

- Nachlässig:
  - Weniger als 8 Zeichen
  - Wörter, die in Wörterbüchern zu finden sind
  - Bieten praktisch keinen Schutz
- Niedrig:
  - 8 oder 9 Zeichen
  - Mindestens zwei der folgenden Arten enthalten:
    - Großschreibung, Kleinschreibung, Zahlen, Sonderzeichen
  - Keine sinnvollen Wörter
  - Lebensdauer maximal 90 Tage
  - Geringe Sicherheit

Passwort-Verfahren
→ Passwortlänge: Empfehlungen (2/4)

- Mittel:
  - 10 oder 11 Zeichen
  - Drei der unter „Niedrig“ aufgeführten Arten enthalten
  - Lebensdauer maximal 60 Tage
- Hoch:
  - 12 bis 15 Zeichen
  - Alle 4 Arten unter „Niedrig“ müssen erfüllt sein
  - Lebensdauer maximal 30 Tage

Passwort-Verfahren
→ Passwortlänge: Empfehlungen (3/4)

- Sehr hoch:
  - Mindestens 16 Zeichen
  - Kein „aussprechbares“ Wort („n8“ = „night“ etc)
  - Lebensdauer maximal 2 Wochen

Passwort-Verfahren
→ Passwortlänge: Empfehlungen (4/4)

- Betrachtung der Sicherheit der unterschiedlichen Stufen:
  - *mit einem einzelnen High-End-PC

Passwort-Stärke	Vollständige Suche*	Passwort-Lebensdauer	erfasster Suchraumanteil* innerhalb der Lebensdauer
niedrig (8)	104 Tage	90 Tage	86,2336%
mittel (10)	2 635 Jahre	60 Tage	0,0062%
hoch (12)	24 Millionen Jahre	30 Tage	0,000000338%
sehr hoch (16)	2 Billiarden Jahre	14 Tage	0,000000000000001859%

- Die Kombination von Passwort-Stärke und Lebensdauer sorgt dafür, dass deutlich mehr Rechenaufwand betrieben werden muss.
- Bei mittlerer Sicherheit werden schon 16 000 Systeme für zwei Monate benötigt!

Passwort-Verfahren

→ Schutzvorkehrung: Fehlbedienungszähler

- Gegen Brute-Force-Attacken hilft eine Limitierung bei der Eingabe der Passwörter.
- Echte Nutzer geben ihr Passwort in der Regel spätestens im dritten Versuch richtig ein.
- Danach sollten IT-Systeme eine Pause vor der nächsten Eingabe erzwingen.
  - Diese kann zunächst wenige Sekunden betragen, sollte aber immer länger werden.
  - Nach einer unrealistischen Zahl von falsch eingegebenen Passwörtern sollte der Zugang gesperrt werden.

Passwort-Verfahren

→ Schutzvorkehrung: Passwörter überprüfen

- Immer dann, wenn ein Passwort erstellt wird, werden als erstes die Passwortregeln überprüft.

- Anschließend sollte überprüft werden, ob das neue Passwort in bekannten Passwortlisten vorhanden ist.

Passwort-Verfahren
→ Angriffsvektor: Abfangen (1/2)

- Idee:
  - Der Angreifer versucht das Passwort während der Übertragung zwischen zwei IT-Systemen mitzulesen.

- Voraussetzung:
  - Das Passwort wird ungeschützt im Klartext über das Kommunikationsnetz übertragen.
  - Angriffe werden häufig als „Man-in-the-Middle-Angriff“ durchgeführt.
  - Angreifer befindet sich physisch (z.B. mit zwei Netzwerkinterfaces in der Übertragungsleitung) oder logisch (z.B. mittels Schadsoftware auf dem Router) zwischen den beiden Kommunikationspartnern.

Passwort-Verfahren
→ Angriffsvektor: Abfangen (2/2)

- Sicherheitsmechanismus: Verschlüsselung
  - Kommunikation zwischen den IT-Systemen wird kryptographisch verschlüsselt (z.B. mittels TLS/SSL oder SSH)

Passwort-Verfahren
→ Angriffsvektor: Entwenden

- Idee:
  - Angreifer verschafft sich Zugang zu der Liste mit den registrierten Nutzernamen und Passwörtern auf einem IT-System.

Passwort-Verfahren
→ Angriffsvektor: Entwenden

- Idee:
  - Angreifer verschafft sich Zugang zu der Liste mit den registrierten Nutzernamen und Passwörtern auf einem IT-System.

- Voraussetzungen:
  - Die Liste wurde im Klartext ohne geeignete Sicherheitsmechanismen auf dem IT-System gespeichert.

Nutzername	Passwort
rainer.maier@gmx.de	Schalke04
peter.hop@gmail.com	Ulrike2003
klaus.mueller@t-online.de	X23y9gl!§0_R

Passwort-Verfahren
→ Passwort-Hash-Verfahren – 1/5

- Passwörter sollten immer als Hashwert gespeichert werden.

Nutzername	Passwort
rainer.maier@gmx.de	3C CB 4D A8 26 66 1D … BF
peter.hop@gmail.com	87 30 28 B3 43 A3 17 … 15
klaus.mueller@t-online.de	3C CB 4D A8 26 66 1D … BF

Passwort-Hash = H (Passwort)

H: One-Way-Hashfunktion
Passwort-Hash: Hashwert des Passwortes

```
Check-Passwort ( Nutzername, Passwort, Liste mit Nutzernamen, Passwort-Hashes )
        suche nach Nutzername in der Liste
        berechne: Passwort-Hash = H ( Passwort )
        if ( überprüfen, ob der Passwort-Hash mit dem in der Liste übereinstimmt )

                return OK
        else
                return ERROR
```

Passwort-Verfahren
→ Passwörter sym. Verschlüsseln

- Passwörter symmetrisch verschlüsseln

Nutzername	Passwort
rainer.maier@gmx.de	3C CB 4D A8 26 66 1D … BF
peter.hop@gmail.com	87 30 28 B3 43 A3 17 … 15
klaus.mueller@t-online.de	3C CB 4D A8 26 66 1D … BF

- Beispiel: (Adobe Passwort Leak, 2013)
  - 153 Millionen Accounts
    - Nur 56 Millionen unterschiedliche Passwörter
  - Sobald der sym. Schlüssel „gefunden“ wurde konnten alle Passwörter entschlüsselt werden

Passwort-Verfahren
→ Passwörter Hashen

- Rainbow-Table enthalten vorberechnete Hashwerte
  - Die Angreiferin berechnet zu „allen Passwörtern“ Hashwerte, speichert diese und nutzte diese später, um die Hashwerte nachzuschlagen.
  - Trade-Off zwischen Speicherplatz und Laufzeit

SHA1 Rainbow Tables

Table ID	Charset	Plaintext Length	Key Space	Success Rate	Table Size	Files	Performance
sha1_ascii-32-95#1-7	ascii-32-95	1 to 7	70,576,641,626,495	99.9 %	52 GB 64 GB	Perfect Non-perfect	Perfect Non-perfect
sha1_ascii-32-95#1-8	ascii-32-95	1 to 8	6,704,780,954,517,120	96.8 %	460 GB 576 GB	Perfect Non-perfect	Perfect Non-perfect
sha1_mixalpha-numeric#1-8	mixalpha-numeric	1 to 8	221,919,451,578,090	99.9 %	127 GB 160 GB	Perfect Non-perfect	Perfect Non-perfect
sha1_mixalpha-numeric#1-9	mixalpha-numeric	1 to 9	13,759,005,997,841,642	96.8 %	690 GB 864 GB	Perfect Non-perfect	Perfect Non-perfect
sha1_loweralpha-numeric#1-9	loweralpha-numeric	1 to 9	104,461,669,716,084	99.9 %	65 GB 80 GB	Perfect Non-perfect	Perfect Non-perfect
sha1_loweralpha-numeric#1-10	loweralpha-numeric	1 to 10	3,760,620,109,779,060	96.8 %	316 GB 396 GB	Perfect Non-perfect	Perfect Non-perfect

http://project-rainbowcrack.com

Passwort-Verfahren
→ Passwort-Hash-Verfahren (2/5)

- Mit einem Passwort-Hash kann sich ein Angreifer nicht an einem IT-System anmelden.
  - Ein Anmeldeprotokoll erwartet immer ein Passwort im Klartext.
  - Angreifer müssen nun ggf. sehr aufwendig alle möglichen Kombinationen ausprobieren, um das eigentliche Passwort herauszufinden.
- Passwort-Hashes sollten weiter abgesichert werden.
  - Rainbow-Tables mit bereits errechneten und optimierten Passwort-Hashes können Angriffe erleichtern.

Passwort-Verfahren
→ Passwörter Hashen

- Rainbow-Table enthalten vorberechnete Hashwerte
  - Die Angreiferin berechnet zu „allen Passwörtern“ Hashwerte, speichert diese und nutzte diese später, um die Hashwerte nachzuschlagen.
  - Trade-Off zwischen Speicherplatz und Laufzeit

SHA1 Rainbow Tables

Table ID	Charset	Plaintext Length	Key Space	Success Rate	Table Size	Files	Performance
sha1_ascii-32-95#1-7	ascii-32-95	1 to 7	70,576,641,626,495	99.9 %	52 GB 64 GB	Perfect Non-perfect	Perfect Non-perfect
sha1_ascii-32-95#1-8	ascii-32-95	1 to 8	6,704,780,954,517,120	96.8 %	460 GB 576 GB	Perfect Non-perfect	Perfect Non-perfect
sha1_mixalpha-numeric#1-8	mixalpha-numeric	1 to 8	221,919,451,578,090	99.9 %	127 GB 160 GB	Perfect Non-perfect	Perfect Non-perfect
sha1_mixalpha-numeric#1-9	mixalpha-numeric	1 to 9	13,759,005,997,841,642	96.8 %	690 GB 864 GB	Perfect Non-perfect	Perfect Non-perfect
sha1_loweralpha-numeric#1-9	loweralpha-numeric	1 to 9	104,461,669,716,084	99.9 %	65 GB 80 GB	Perfect Non-perfect	Perfect Non-perfect
sha1_loweralpha-numeric#1-10	loweralpha-numeric	1 to 10	3,760,620,109,779,060	96.8 %	316 GB 396 GB	Perfect Non-perfect	Perfect Non-perfect

http://project-rainbowcrack.com

Passwort-Verfahren
→ Passwort-Hash-Verfahren (3/5)

- Angriffe mit Rainbow-Tables können durch das Hinzufügen von Zufallszahlen (Salt und Pepper) unwirtschaftlich gemacht werden.
  - Angreifer muss alle möglichen Passwörter in Kombination mit allen möglichen Salts und Peppers ausprobieren.

Passwort-Hash = H (Passwort || Salt || Pepper)

H: One-Way-Hashfunktion
Passwort-Hash: Hashwert des Passwortes
Salt: Zufallszahl, die in der Liste steht.
Pepper: Zufallszahl, die geheim ist.

Nutzername	Passwort	Salt	Pepper
rainer.maier@gmx.de	B1 F3 7C AD F4 8D AC … 55	FA 9C 13 0D 66 … 1D	
peter.hop@gmail.com	FA 2F B3 AB 95 26 B7 … 3A	10 29 C6 EC A3 … 17	AA 14 BD 2C 77 … 35
klaus.mueller@t-online.de	29 4C CB FB 33 29 FD … 3D	02 EE 74 1B 66 … 1D	

Passwort-Verfahren
→ Passwort-Hash-Verfahren (4/5)

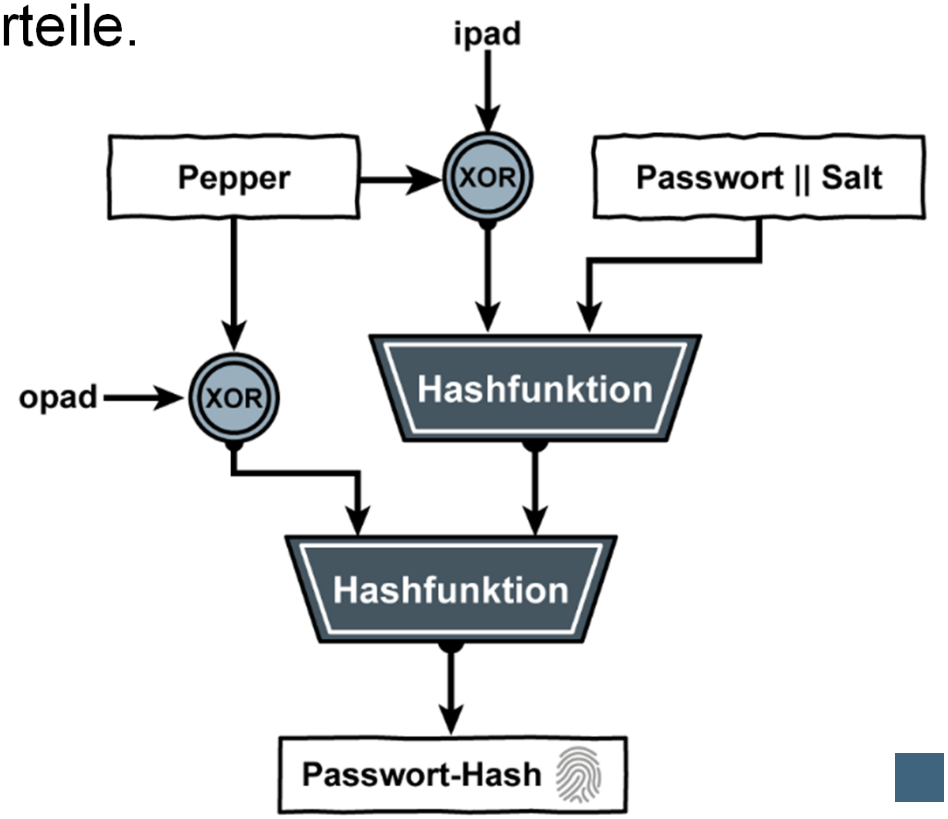
```
Check-Passwort(Nutzername,Passwort,Liste mit Nutzernamen,Passwort-Hashes,Salts,Pepper)
        suche nach Nutzername in der Liste
        berechne: Passwort-Hash = H ( Passwort || Salt || Pepper )
        if ( überprüfen, ob der Passwort-Hash mit dem in der Liste übereinstimmt )
                return OK
        else
                return ERROR
```

- **Randbedingungen für die Nutzung von Salt:**
  - Salt muss eine hochwertige Zufallszahl sein.
  - Salt muss für jeden Nutzer zufällig und unabhängig gewählt werden.
  - Faustregel: Salt soll so groß sein wie die Ausgabegröße der Hashfunktion (SHA3: 256-Bit Hashwert).
  - Ein n-Bit Salt verlangsamt einen Angriff um den Faktor 2^n.

Passwort-Verfahren
→ Passwort-Hash-Verfahren (5/5)

- **Randbedingungen für die Nutzung von Pepper:**
  - Die Zeichenfolge ist für alle Passwörter gleich.
  - Die Zeichenfolge muss geheim und sicher gespeichert werden.
  - Kennt der Angreifer die Zeichenfolge, so bringt der Sicherheitsmechanismus keinerlei Vorteile.

- Eine andere Variante ist, Pepper als Schlüssel für den HMAC zu verwenden.



Passwort-Verfahren
→ Hash Stretching

- Starkes Hash Stretching ist nicht trivial zu entwerfen
  - Beispiel 1: $H(H(H(H \ldots H(pw))))$
    - Ab der 2ten ist der Eingabe-Zeichenraum auf den Ausgabezeichenraum der Hashfunktion beschränkt
  - Beispiel 2: $H(\,pw\,||H(\,pw\,||H(\,pw\,||H \ldots H(pw))))$
    - So wir zumindest der Eingabezeichenraum durch mehrere Runden verwendet

Passwort-Verfahren
→ Übersicht

Summary

Method	Examples	Security for passwords	Can be broken by
Plain text	`password.txt`	**Completely insecure**	Anyone
Hashing	`md5`, `SHA-1`, `SHA-256`	**Very insecure**	Rainbow tables Cracking Parallel cracking Predictive cracking Human error
Hashing + Salting	`SHA-1 with salt` `SHA-256 with salt`	**Very insecure**	Cracking Parallel cracking Predictive cracking Human error
Hashing + Salting + Key stretching	`Repeated MD5`	**Insecure**	Parallel cracking Predictive cracking Human error
	`Repeated HMAC-SHA256`	**Nonstandard practice**	
	`PBKDF2-HMAC-SHA256`	**Still good**	
Hashing + Salting + Memory-hard key stretching	`bcrypt`	**Good practice**	Predictive cracking Human error
	`scrypt`	**Good practice**	
	`Argon2`	**Best practice**	
Usable security innovations	Password managers	**Best practice**	Human error
	Security keys	**Best practice**	

Passwort-Verfahren
→ Weitere Angriffsvektoren: Keylogger

- Spezielle Malware, die Eingaben auf einem IT-System abfängt und speichert.
  - Diese Informationen sind hauptsächlich Nutzernamen und Passwörter.

- In regelmäßigen Abständen werden von der Malware die gespeicherten Informationen in sogenannte Drop-Zonen im Internet gesendet.
  - Drop-Zonen sind Speicherbereiche von beliebigen Servern im Internet, von denen sich die Angreifer die Informationen unentdeckt holen können.
  - Mit den Daten werden Angriffe auf die Internet-Dienste der Opfer durchgeführt.

Passwort-Verfahren
→ Weitere Angriffsvektoren: Phishing (1)

- Spezielle Form des Social Engineerings.
- Werkzeuge: Gefälschte Webseiten, E-Mails, Kurznachrichten, psychologische Tricks, ...
- Beispiel: Eine Phishing-Webseite sieht aus wie eine Original-Webseite, ist jedoch eine vom Angreifer präparierte Webseite.
  - Nachahmung des Corporate Designs (dieselben Firmenlogos, Schriftarten, Layouts, usw.).
- Angreifer versuchen an die persönliche Daten eines Nutzers zu gelangen, um damit Identitätsdiebstahl zu begehen.
- Beim klassischen Phishing werden große Mengen von E-Mails wahllos an Empfänger verschickt.

Passwort-Verfahren
→ Weitere Angriffsvektoren: Phishing (2)

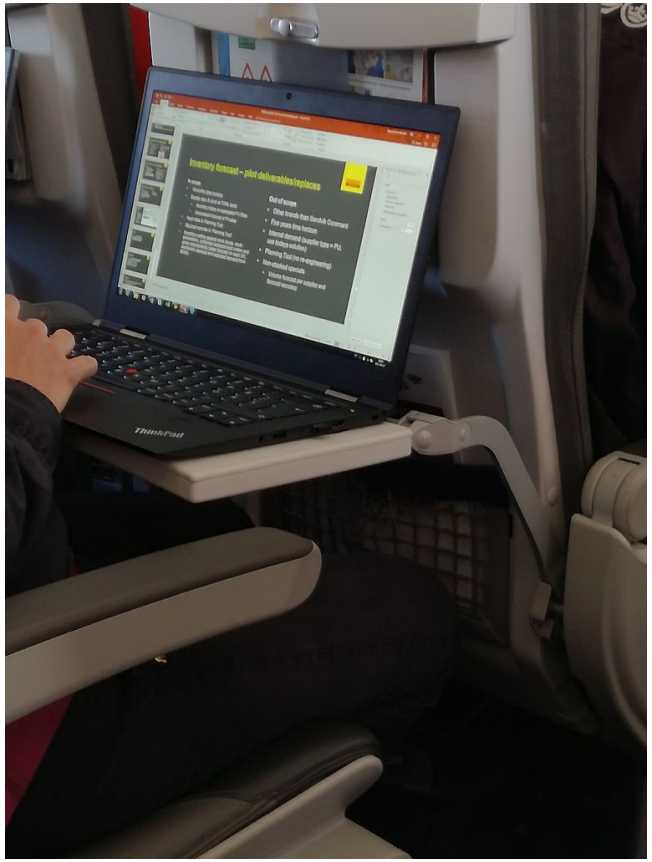
- Beim Spear-Phishing werden die Empfänger sorgfältig recherchiert und erhalten z.B. E-Mails, die auf sie persönlich zugeschnitten sind und viel glaubwürdiger wirken.
  - Richtet sich in der Regel gegen Mitarbeiter einer konkreten Organisation und zielt darauf ab, nicht autorisierten Zugriff auf vertrauliche Daten zu erhalten.

- Whaling ist ein Spear-Phishing Angriff, der gezielt gegen hohe Führungskräfte gerichtet ist.

Passwort-Verfahren

→ Weitere Angriffsvektoren: Shoulder Surfing

- Spezielle Form des Social Engineerings.
- Beobachten der Eingabe von sensiblen Informationen in öffentlichen Bereichen (z.B. im Flugzeug, Zug, Bus, Cafe, …).
- Das Beobachten der Eingabe eines Passwortes ist eine prinzipielle Möglichkeit an Passwörter zu kommen.



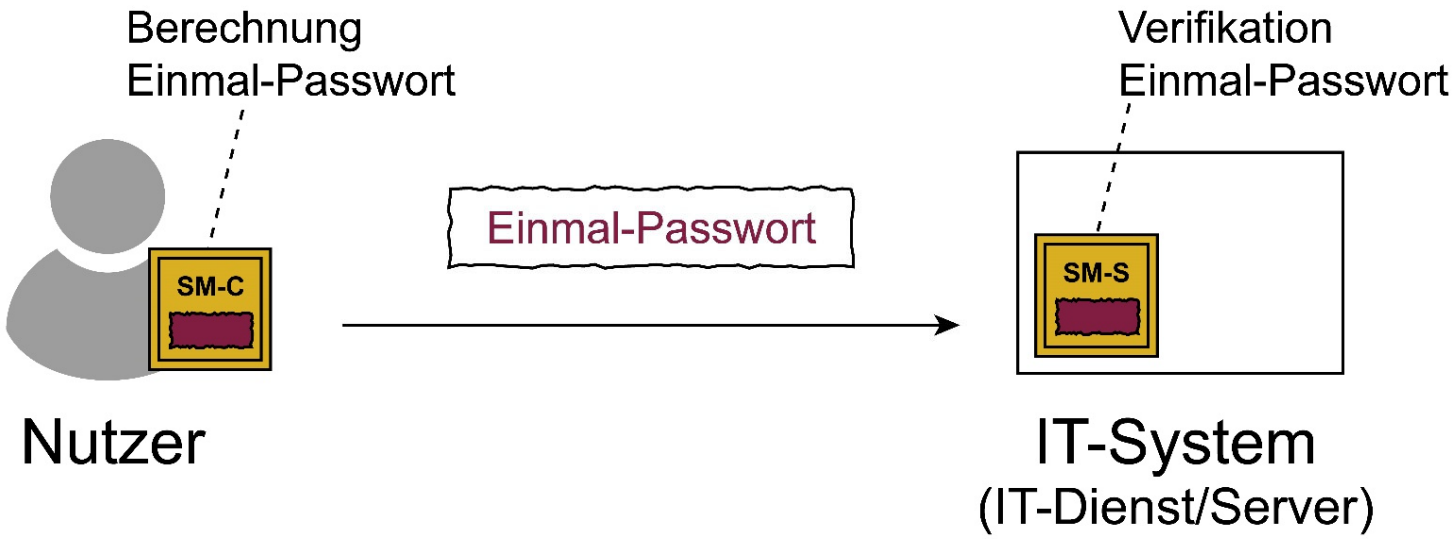
Identifikation und Authentifikation
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Identifikation und Authentifikation
- Generelle Authentifikationsverfahren
- Passwort-Verfahren
- **Einmal-Passwort-Verfahren**
- Challenge-Response-Verfahren
- Biometrische Verfahren
- Mehrfaktor-Authentifizierung
- Moderne Authentifizierungssysteme
- FIDO
- Zusammenfassung

Einmal-Passwort-Verfahren
→ Übersicht (1/2)

- Ein Einmalpasswort (engl. One-Time Password - OTP) ist ein Authentifikationsverfahren bei dem ein Passwort nur einmal für eine Session benutzt werden kann.
- Damit wird ausgeschlossen, dass ein Angreifer ein Passwort abhören und erneut verwenden kann (Replay-Attacken)
- „Man in the Middle Attacken“ sind immer noch möglich!
- Durch Einmal-Passworte können Nachteile des Passwort-Verfahrens überwunden werden.
  - In der Praxis ist der Aufwand für das Verfahren hoch und wird nur für wichtige Anwendungen akzeptiert.

Einmal-Passwort-Verfahren
→ Übersicht (2/2)



- Die Einmal-Passwörter werden in der Regel in einem Hardware-Sicherheitsmodul (SM-C) des Nutzers generiert.
- Die Verifikation findet in einem Hardware-Sicherheitsmodul (SM-S) des IT-Systems statt.
- Das Einmal-Passwort ist nur für eine bestimmte Zeit nach der Generierung gültig.

Einmal-Passwort-Verfahren
→ Vorgenerierte Listen

- Es werden vorgenerierte Listen verwendet, die dem Benutzer vorher vertraulich über einen sicheren Kanal mitgeteilt wurden.
- Limitierung der Anmeldevorgänge durch Begrenzung der Anzahl an vorgenerierten Einmal-Passwörtern.
- Wenn die Einmal-Passworte zu Ende sind, muss der Nutzer neue beantragen.
- Bei dieser Methode werden keine Sicherheits-Module für die Berechnung und Verifikation der Einmal-Passworte gebraucht.
- Ein Beispiel für diese Methode sind iTAN-Listen im Bankenumfeld (heute veraltet).

Einmal-Passwort-Verfahren
→ Kennwortgeneratoren (1/2)

- Nutzer und das IT-System berechnen nach einem definierten Verfahren das Einmal-Passwort während des Authentisierungsprozesses.
- Es werden z.B. kryptographische Hash-Funktionen zur Generierung von nur kurzzeitig gültigen Einmalpasswörtern verwendet.
- Alternative: Zeitgesteuerte Generatoren.
  - Beispiele sind SecurID von der Firma RSA Security oder das von Bellcore entwickelte S/Key.

Einmal-Passwort-Verfahren
→ Kennwortgeneratoren (2/2)

Einmal-Passwort = f (Zeit || GX)

f: Kryptographische Funktion (One-Way-Hashfunktion, Verschlüsselungsverfahren)
Zeit: Eine relative oder absolute Zeitangabe
GX: Geheimnis des Nutzers (X)

- Voraussetzungen: Beide Seiten kennen die Funktion f und das Geheimnis GX.
- Damit auf der überprüfenden Seite nicht für jeden Nutzer (X) ein Geheimnis (GX) gespeichert werden muss, wird die in der Regel auf der Basis eines Master-Schlüssels berechnet.

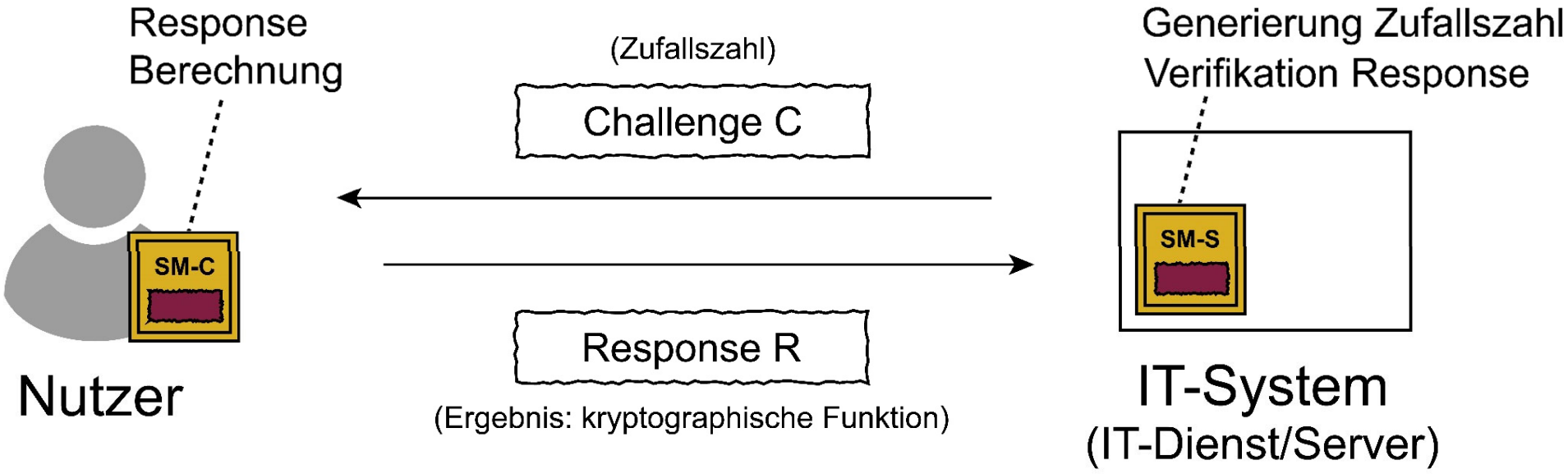
GX = H (Nutzername || Master-Schlüssel)

Identifikation und Authentifikation
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Identifikation und Authentifikation
- Generelle Authentifikationsverfahren
- Passwort-Verfahren
- Einmal-Passwort-Verfahren
- **Challenge-Response-Verfahren**
- Biometrische Verfahren
- Mehrfaktor-Authentifizierung
- Moderne Authentifizierungssysteme
- FIDO
- Zusammenfassung

Challenge-Response-Verfahren
→ Generelle Idee (1/2)

- Der Besitz eines Geheimnisses (z.B. privater Schlüssel) wird gegenüber einem IT-Systems mittels Durchführung einer spontanen kryptographischen Operation bewiesen.



Challenge-Response-Verfahren
→ Generelle Idee (2/2)

- Aufgezeichnete Informationen können kein zweites Mal verwendet werden, da immer neue Zufallszahlen als Challenge gesendet werden.
- Bei einer Authentifizierung über unsichere Netze müssen Challenge-Response-Verfahren eingesetzt werden, um ein Abhören und daraus resultierende missbräuchliche Verwendung zu verhindern.
- Für die Speicherung der geheimen Schlüssel und die Berechnung der kryptographischen Verfahren können z.B. Hardware-Sicherheitsmodule verwendet werden.
- Es gibt unterschiedliche Methoden, wie Challenge-Response-Verfahren umgesetzt werden können.

Challenge-Response-Verfahren
→ mit Hashfunktion und Geheimnis (1/2)

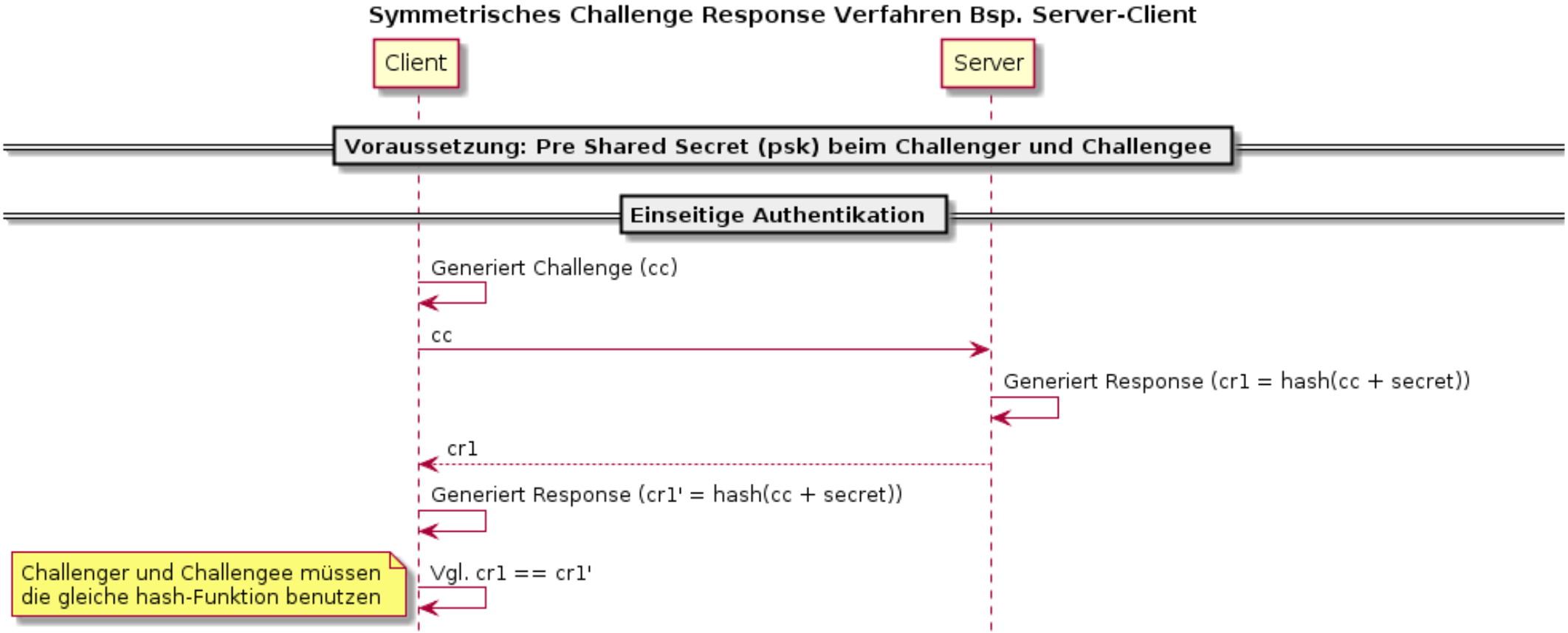
Response = H (C || G_X)

H: One-Way-Hashfunktion
C: Zufallszahl (Challenge), die gehasht werden soll
G_X: Geheimnis des Nutzers (X), dessen Besitz bewiesen werden soll

- Voraussetzungen: Beide Seiten kennen die Funktion f und das Geheimnis G_X.
- Damit auf der überprüfenden Seite nicht für jeden Nutzer (X) ein Geheimnis (G_X) gespeichert werden muss, wird die in der Regel auf der Basis eines Master-Schlüssels berechnet.

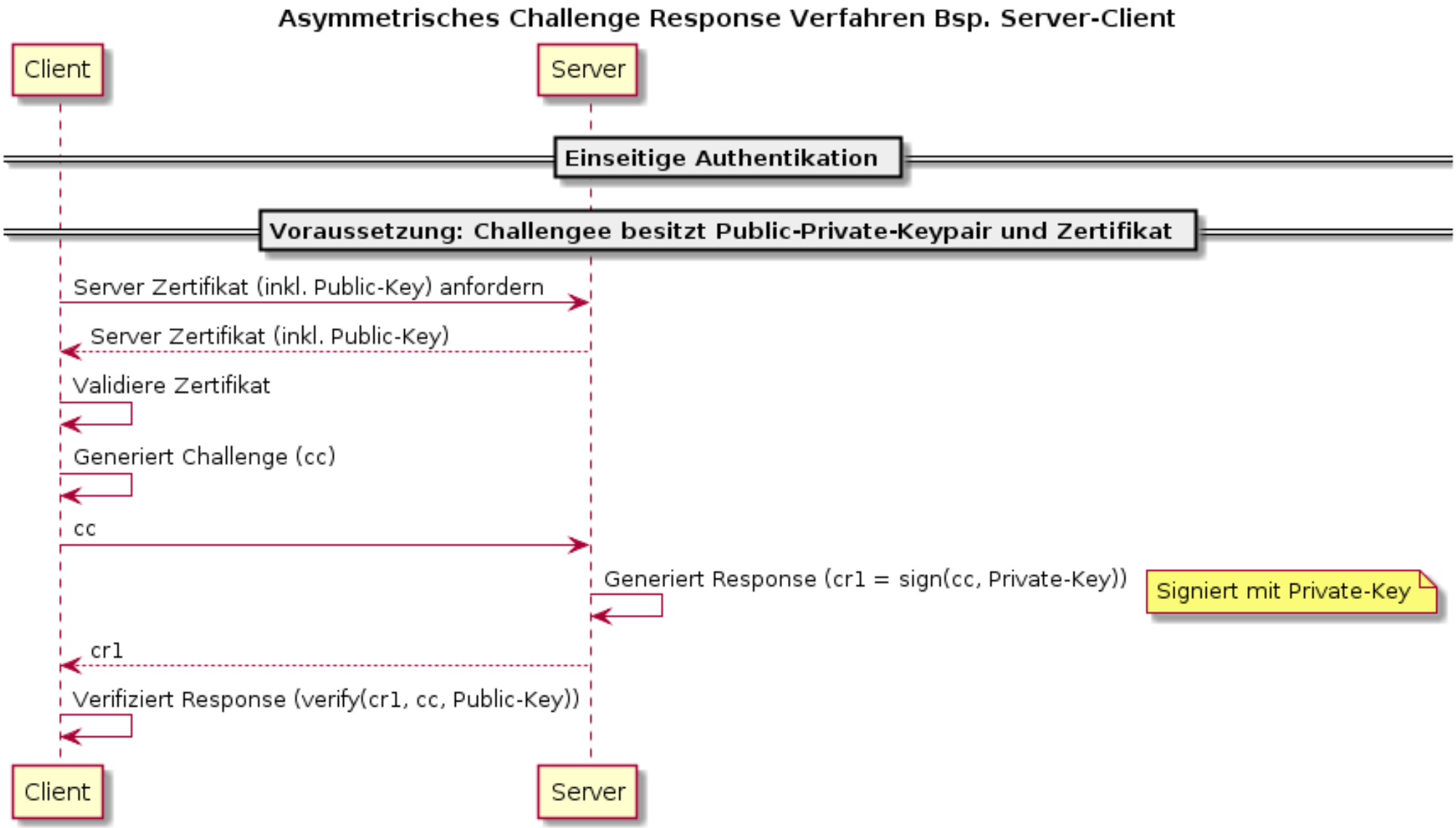
G_X = H (C || Master-Schlüssel)

Challenge-Response-Verfahren
→ mit Hashfunktion und Geheimnis (2/2)



- Alternative: Symmetrisches Verschlüsselungsverfahren (z.B. AES) anstelle der Hashfunktion.

Challenge-Response-Verfahren
→ mit Public-Private-Key und Zertifikaten

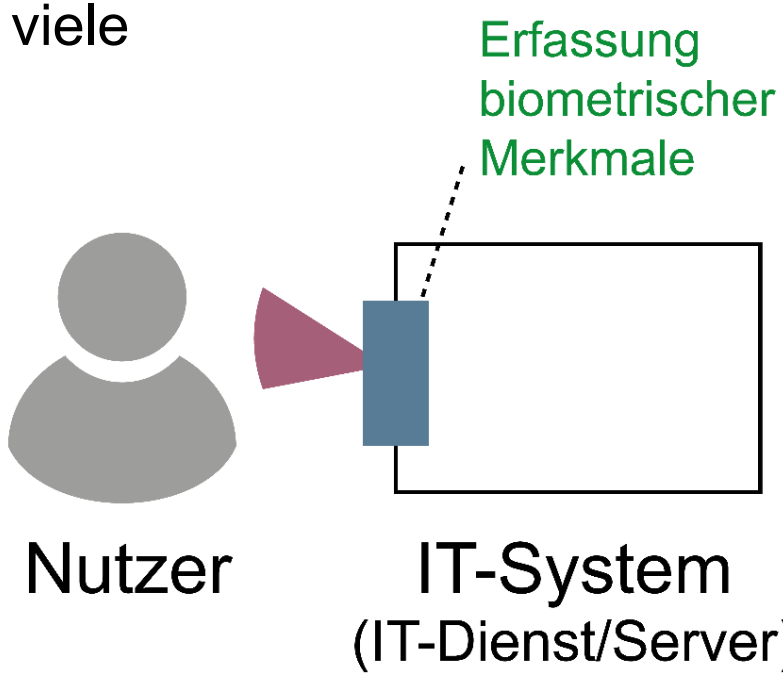


Identifikation und Authentifikation
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Identifikation und Authentifikation
- Generelle Authentifikationsverfahren
- Passwort-Verfahren
- Einmal-Passwort-Verfahren
- Challenge-Response-Verfahren
- **Biometrische Verfahren**
- Mehrfaktor-Authentifizierung
- Moderne Authentifizierungssysteme
- FIDO
- Zusammenfassung

Biometrische Verfahren
→ Generelle Idee

- Biometrie ist die Identifikation und Authentifizierung mittels biologischer Merkmale.
  - Es werden physiologische oder verhaltenstypische, also personengebundene Charakteristika verwendet.
  - Biometrische Merkmale können nicht unmittelbar gestohlen und im Allgemeinen nur schwer kopiert werden.
  - Es gibt zahlreiche Merkmale, die auf viele Arten gemessen werden können.



Biometrische Verfahren
→ Biologische Merkmale

Aktive Merkmale	Passive Merkmale
Unterschriftdynamik	Gesichtserkennung
Schreibverhalten	Retinamuster
Tippverhalten an der Tastatur	Irismuster
Stimmerkennung	Fingerabdruck (Daktylogramm)
Lippenbewegung beim Sprechen	Form des Ohres
Gestik/Mimik beim Sprechen	Handgeometrie
Bewegung (Gangartzyklus)	Venenmuster auf dem Handrücken
	Geruch
	DNA
	Thermogramm

- Beispiel: Erfassung des Gesichts und der Gesichtsdynamik beim Sprechen, in Kombination mit der Stimmerkennung.

Biometrische Verfahren
→ Nutzbarkeit von biomet. Merkmalen (1)

- Jede physiologische oder verhaltensbedingte Eigenschaft kann als biometrisches Merkmal zur Personenidentifikation verwendet werden, sofern sie folgende Anforderungen erfüllt:
- Universalität:
  - Jede Person muss dieses Merkmal besitzen
- Einzigartigkeit / Einmaligkeit:
  - Das Merkmal ist bei unterschiedlichen Menschen hinreichend verschieden.
  - Vielmehr gibt es keine zwei oder mehr Personen mit dem gleichen Merkmal (Zwillinge).
- Erfassbarkeit:
  - Das Merkmal ist quantitativ messbar.

Biometrische Verfahren
→ Nutzbarkeit von biomet. Merkmalen (2)

- Konstanz:
  - Das Merkmal sollte sich im Laufe der Zeit möglichst wenig ändern.
  - Kleinere Veränderungen können durch adaptive biometrische Verfahren ausgeglichen werden.

- Merkmalsverbreitung:
  - Kleine Bevölkerungsgruppen weisen gewisse Merkmale nicht auf bzw. für sie sind bestimmte Verfahren nicht geeignet.
  - So besitzt zum Beispiel ein kleiner Bevölkerungsanteil keine ausgeprägten Fingerabdruckstrukturen.
  - Besteht die Gefahr des Verlustes oder der Nichtverwendbarkeit eines biometrischen Merkmals, sollte ein Ersatzsystem vorgesehen werden.

Biometrische Verfahren
→ Nutzbarkeit von biomet. Merkmalen (3)

- Möglichkeit zur willentlichen Beeinflussung durch den Nutzer:
  - Einige biometrische Merkmale bieten die Möglichkeit, neben dem Hauptmerkmal eine zusätzliche Information zu übermitteln.
  - Bei der Stimmerkennung besteht z.B. die Möglichkeit verschiedene Schlüsselwörter anzulernen und zu speichern.
  - Diese Eigenschaft gewinnt besondere Bedeutung in Anwendungsszenarien, in denen mit einer Erpressung des Merkmalsträgers gerechnet werden muss.
  - Der Erpresste kann auf diese Weise einen stillen Alarm abgeben, ohne dass der Erpresser das erkennt.

Biometrische Verfahren
→ Akzeptanzraten (1/2)

- FAR = False Acceptance Rate
  - Sicherheitsmerkmal
  - Erkennung einer nichtberechtigten Person als berechtigt

$$FAR = \frac{fälschlich\ akzeptierte\ Zugriffe}{unberechtigte\ Zugriffsversuche}$$

- **FRR = False Rejection Rate**
  - Komfortmerkmal
  - Unberechtigte Abweisung berechtigter Personen

$$FRR = \frac{fälschlich\ zurückgewiesene\ Zugriffe}{berechtigte\ Zugriffsversuche}$$

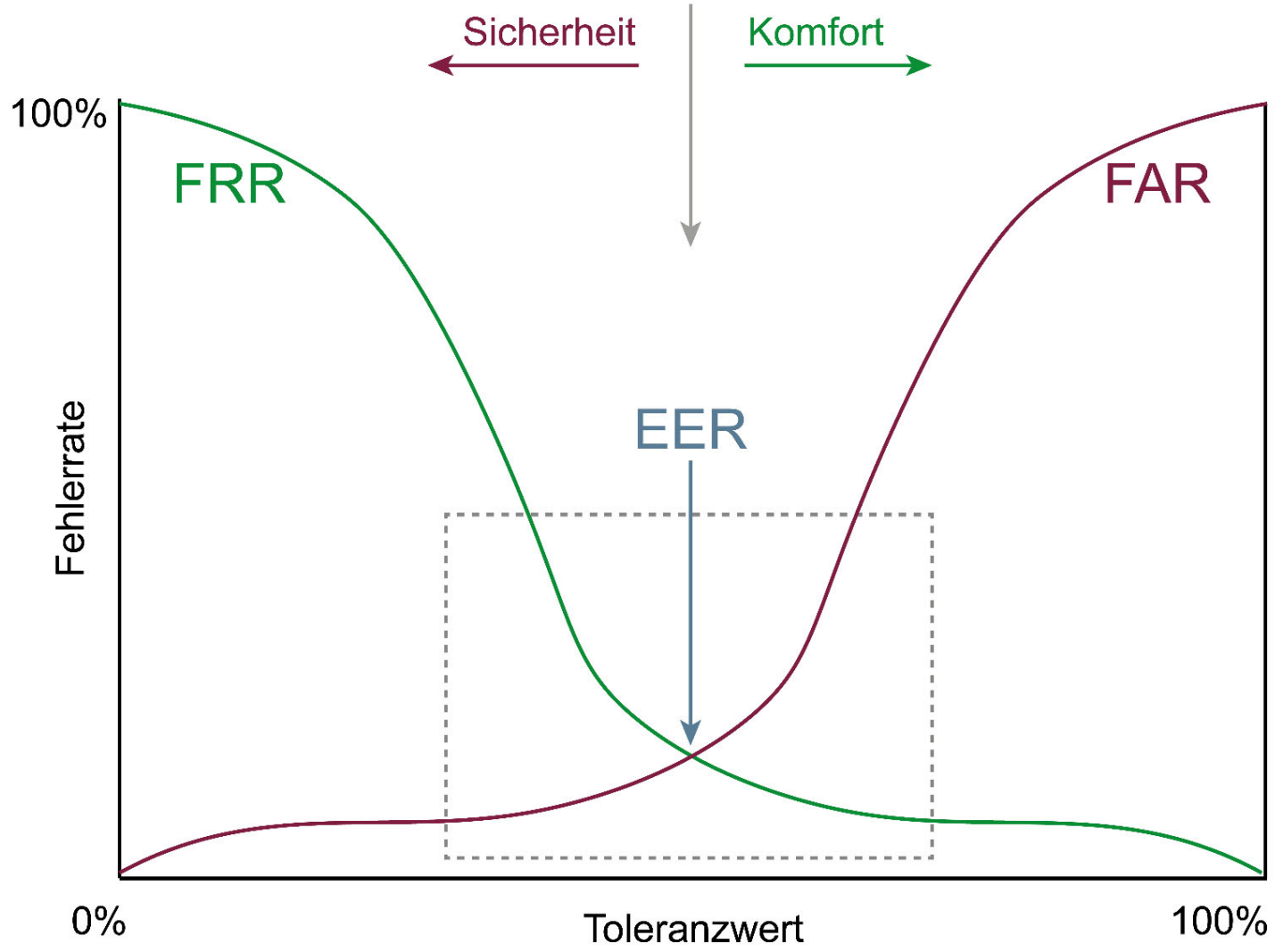
Biometrische Verfahren
→ Akzeptanzraten (2/2)

- Die Übereinstimmungserfordernisse bei biometrischen Merkmalen müssen immer einen gewissen Spielraum offen halten.
  - Abweichungen müssen ggf. toleriert werden (z.B. Rückstände von Staub, Schmutz oder Fett auf der Haut, unterschiedliche Stimmungslagen einer Person, physiologische Temperaturschwankungen, …)
  - Die Wahrscheinlichkeit der Falschakzeptanz und der Falschrückweisung müssen in eine akzeptable Relation zum Sicherheitslevel gebracht werden.
- Aus biometrischen Merkmalen kann kein kryptographischer Schlüssel abgeleitet werden.
  - Ein solcher beruht immer auf einer genauen mathematischen Berechnung, die keine Schwankungen zulässt.

Biometrische Verfahren
→ Equal Error Rate (1/3)

- Je kritischer die Sicherheit für ein IT-System ist, desto eher sollte von einem Nutzer erwartet werden, eine fälschliche Abweisung hinzunehmen (z.B. im Hochsicherheitsbereich).
- In alltäglichen Massenanwendungen sollte die Nutzerakzeptanz nicht durch häufige fälschliche Rückweisungen verringert werden.
- Ein gutes Konzept ist, wenn die Akzeptanz und Rückweisung gleich groß sind → Equal Error Rate (EER).
  - Je niedriger die EER, desto besser die Leistung des biometrischen Verfahrens und desto geringer die Gesamtfehlerrate.

Biometrische Verfahren
→ Equal Error Rate (2/3)



Biometrische Verfahren
→ Equal Error Rate (3/3)

Biometrisches Verfahren	FAR in %	FFR in %
Fingerabdruck	0,001 ··· 2	0,1 ··· 5
Iriserkennung	0,0001 ··· 1	0,1 ··· 2
Gesichtserkennung	0,5 ··· 2	1 ··· 3
Handgeometrie	1 ··· 4	1 ··· 5

Biometrische Verfahren
→ Nutzerakzeptanz (1/2)

- Komfort / Praktikabilität
  - Einfachheit der Handhabung
  - Zeitaufwand bei der Registrierung
  - Zeitaufwand im Normal- und im Sonderfall (False Rejection)
  - Häufigkeit der Aktualisierung des Musters
  - Aufwand zur Referenzdatenerfassung
  - Möglichkeiten einer zeitweiligen Ersatzlösung und Aufwand dieser Ersatzlösung für den Nutzer
- Belästigung
  - Eindringen in die persönliche Schutzsphäre
  - Hygiene

Biometrische Verfahren
→ Nutzerakzeptanz (2/2)

- Vertrautheit / Transparenz
  - Vertrautheit mit bereits bekannten und etablierten Vorgängen
  - Zusammenhänge und Abläufe werde verstanden
  - Bereitschaft zur Kooperation

- Vorurteile und Ängste
  - Vorurteile gegen den Vorgang der Registrierung im System oder der Anwendung
  - Angst vor Missbrauch (z.B. wird die Methode auch erkennungsdienstlich verwendet?)
  - Angst vor Verletzungen (z.B. beim Netzhaut-Scanning)

Biometrische Verfahren
→ Vergleich

- Die Nutzbarkeit, Nutzerakzeptanz, sowie technische und finanzielle Aufwendungen müssen in Relation zur Sicherheit gesetzt werden.

Rank	Accuracy	Convenience	Cost	MOC integration
1	DNA	Voice	Voice	Finger
2	Iris	Face	Signature	Voice
3	Retina	Signature	Finger	
4	Finger	Finger	Face	
5	Face	Iris	Iris	
6	Signature	Retina	Retina	
7	Voice	DNA	DNA	

Morgan Keegan & Co (Januar 2001)

Fahndung nach Schwerverbrechern

Hochsicherheitsanwendungen

Alltägliche Sicherheitsanforderungen

Biometrische Verfahren
→ Anwendung

- Identifikation: Feststellung der Identität
  - Die aktuellen biometrischen Daten einer Person werden erfasst und mit Referenzdaten einer Vielzahl von Individuen verglichen (1:n-Vergleich).
  - Der Referenzdatensatz mit der geringsten Abweichung zu den aktuellen biometrischen Daten legt die festgestellte Identität fest.
  - Hierfür muss im Vorfeld eine Toleranzgrenze definiert werden.

- Verifikation: Bestätigung der Identität
  - Die aktuellen biometrischen Daten einer Person werden erfasst und mit Referenzdaten eines Individuums verglichen (1:1-Vergleich).
  - Eine behauptete Identität wird bestätigt, wenn die beiden Datensätze innerhalb einer Toleranzgrenze übereinstimmen.

Biometrische Verfahren
→ Angriffe

- Angreifer habe eine Vielzahl an Möglichkeiten, Biometrie-Verfahren zu überlisten
  - z.B. Vorzeigen eines Fotos, Erstellung eines künstlichen Fingers mit geklautem Fingerabdruck.

- Durch die Erweiterung der Biometrie-Verfahren mit einer Lebendanalyse, kann solchen Angriffen entgegengewirkt werden.
  - z.B. durch Analyse der Pupillenbewegung oder Messung des Blutdruckes

Identifikation und Authentifikation
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Identifikation und Authentifikation
- Generelle Authentifikationsverfahren
- Passwort-Verfahren
- Einmal-Passwort-Verfahren
- Challenge-Response-Verfahren
- Biometrische Verfahren
- **Mehrfaktor-Authentifizierung**
- Moderne Authentifizierungssysteme
- FIDO
- Zusammenfassung

Mehrfaktor-Authentifizierung
→ Generelle Idee (1/2)

- Die Mehrfaktor-Authentifizierung (engl. Multi-factor authentication, kurz MFA) verwendet unterschiedliche und insbesondere unabhängige Klassen von Authentifizierungsverfahren.
  - „Out-of-band“-Verfahren bieten hohe Sicherheit.

- Die Klassen der Multi-Faktor-Authentifizierung sind:
  - etwas, das der Nutzer besitzt, wie zum Beispiel ein Hardware-Sicherheitsmodul
  - etwas, das der Nutzer weiß, wie zum Beispiel ein Passwort oder eine PIN
  - etwas, das als körperliches Charakteristikum untrennbar zum Nutzer gehört (das Sein), wie zum Beispiel ein Fingerabdruck oder die menschliche Stimme.

Mehrfaktor-Authentifizierung
→ Generelle Idee (2/2)

- Beispiel: Challenge-Respons-Verfahren mit Hilfe eines Hardware-Sicherheitsmoduls, das mit einem Passwort oder PIN aktiviert werden muss. Um den Nutzerbezug zu verstärken, muss der Nutzer noch mit Hilfe eines Fingerabdrucks seine Identität zusätzlich verifizieren lassen.

- Eine häufige Variante ist die Zwei-Faktor-Authentifizierung (2FA) mit Besitz und Wissen.
  - Beispiel: Hardware-Sicherheitsmodul (Smartcard, USB-Token, …) mit PIN zur Aktivierung des Hardware-Sicherheitsmoduls.

Mehrfaktor-Authentifizierung
→ Roll-Out von Faktoren (1/2)

- Missbrauch kann nur verhindert werden, wenn die Faktoren gemäß den Cyber-Sicherheitsbedürfnissen verteilt, registriert und verifiziert wurden.

- Die Faktoren für die Authentifikation sind nur so stark, wie die Kanäle über die sie verteilt wurden.
  - z.B. mittels SMS, Scannen eines QR-Codes, E-Mail, …

- Identitäten müssen die Faktoren registrieren.
  - z.B. durch Festlegung eines Passwortes, Eingabe einer Mobilfunknummer, E-Mail-Adresse, Seriennummer, …

Mehrfaktor-Authentifizierung
→ Roll-Out von Faktoren (2/2)

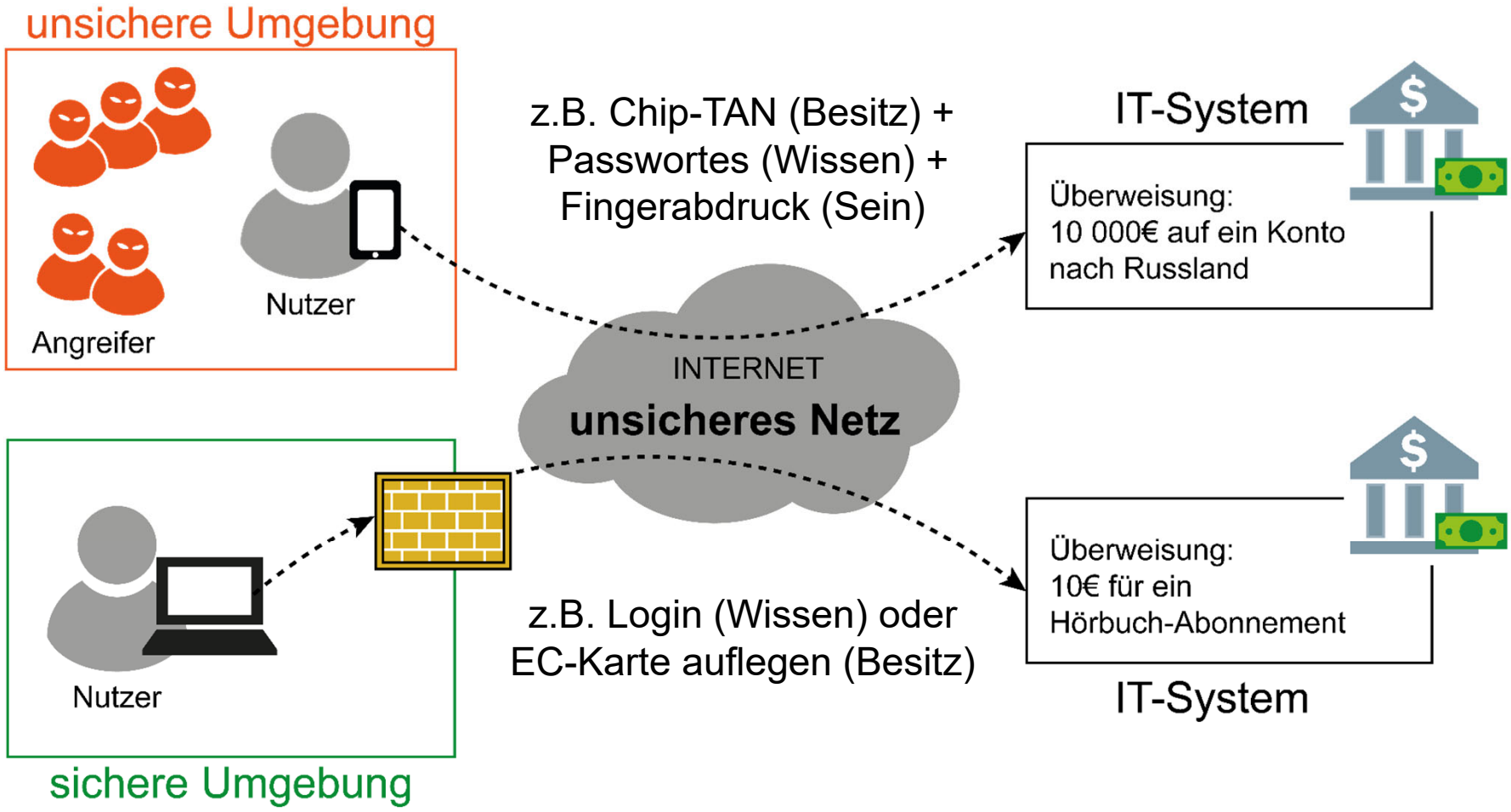
- Die Zugehörigkeit eines registrierten Faktors zu einer Identität muss vor der Verwendung verifiziert werden.
  - z.B. mittels Verifizierungslink per E-Mail, Personalisierung eines Smartphones mittels SMS, …

Mehrfaktor-Authentifizierung
→ Risikobasiert und adaptiv (1/6)

- Auswahl der Authentifikationsverfahren auf Basis der
  - Vertrauenswürdigkeit des zugreifenden Nutzers,
  - Kritikalität der konkreten Anwendung/Aktion,
  - Rahmenbedingungen des aktuellen Zugriffes.

- Die passenden Authentifikationsverfahren werden in Abhängigkeit des gerade notwendigen Sicherheitsniveaus ausgewählt.
  - Verschiedene Kombinationen der MFA sollen bedarfsgerecht zum Einsatz kommen.
  - Es wird das Optimum zwischen Sicherheit und Komfort angestrebt (z.B. Minimierung der Anzahl nicht notwendiger starker Authentifizierungen).

Mehrfaktor-Authentifizierung
→ Risikobasiert und adaptiv (2/6)



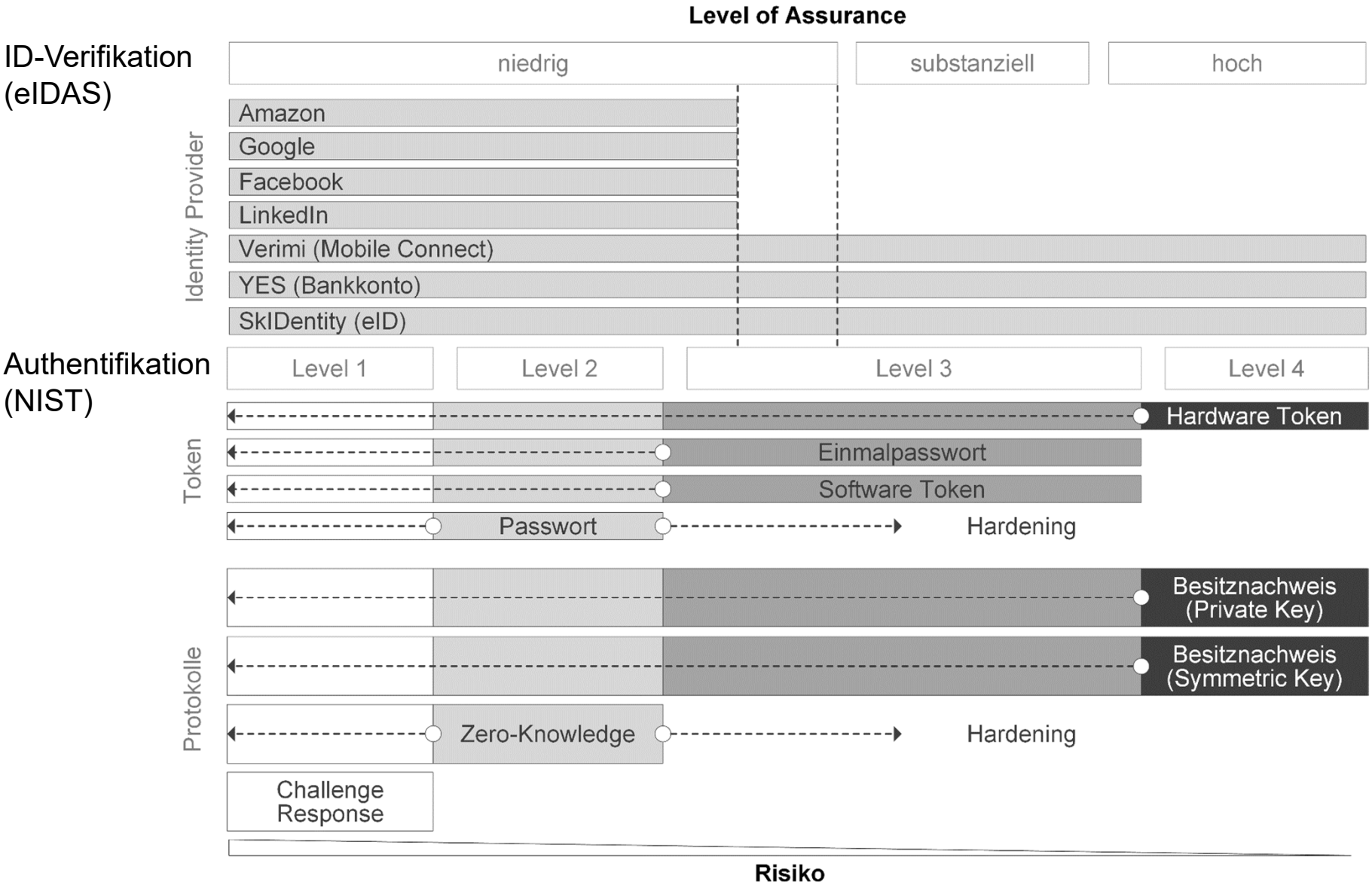
Mehrfaktor-Authentifizierung
→ Risikobasiert und adaptiv (3/6)

- Für die Bestimmung des Sicherheitsniveaus kann eine vierte Klasse für die MFA gebildet werden.

Faktor	Beispiele
Wissen	Benutzername, Kundennummer, Geburtsort, Geburtsdatum, PIN, Passwort
Besitz	Kryptographische Schlüssel, Hard –und Software Token, Sicherheitsmodule, Smartcards
Inhärenz	Unterschrift, Fingerabdruck, Stimme, Tippverhalten, Mausbewegungen
Verhalten	**Vergangene Transaktionen, verwendete Geräte, Besuchte Orte, verwendete Softwareversionen, Aktivitäten in sozialen Medien, Timing**

- Es gibt viele potenzielle Datenquellen für die Bewertung des Sicherheitsniveaus.

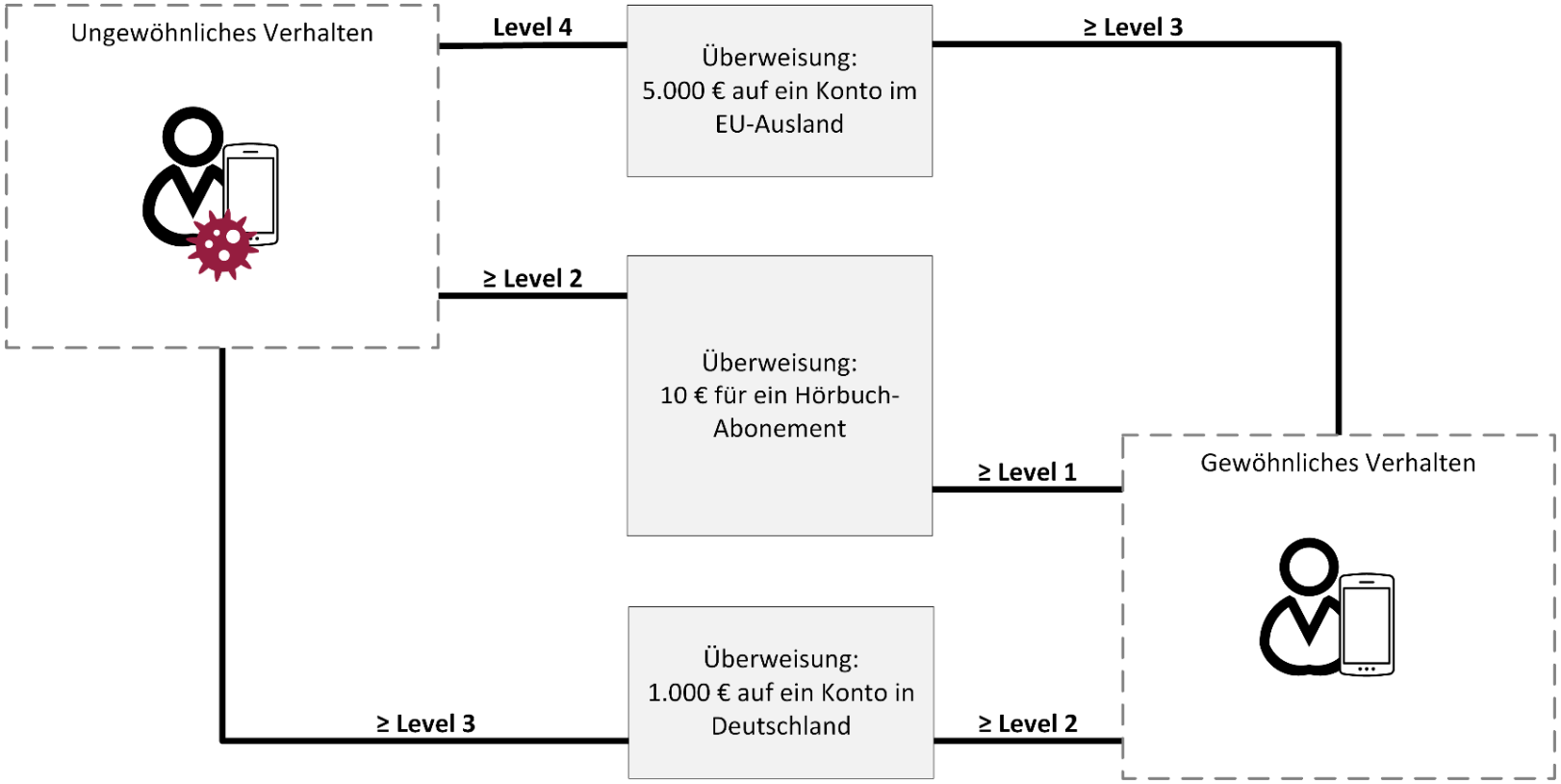
Mehrfaktor-Authentifizierung
→ Risikobasiert und adaptiv (4/6)



Mehrfaktor-Authentifizierung
→ Risikobasiert und adaptiv (5/6)

		Auswirkungen durch Schäden		
		Low	**Moderate**	**High**
Eintrittswahr-scheinlichkeit von Schäden	**Low**	Level 1	Level 2	Level 3
	Moderate	Level 2	Level 3	Level 3-4
	High	Level 3	Level 3-4	Level 4

Mehrfaktor-Authentifizierung
→ Risikobasiert und adaptiv (6/6)



Identifikation und Authentifikation
→ Inhalt

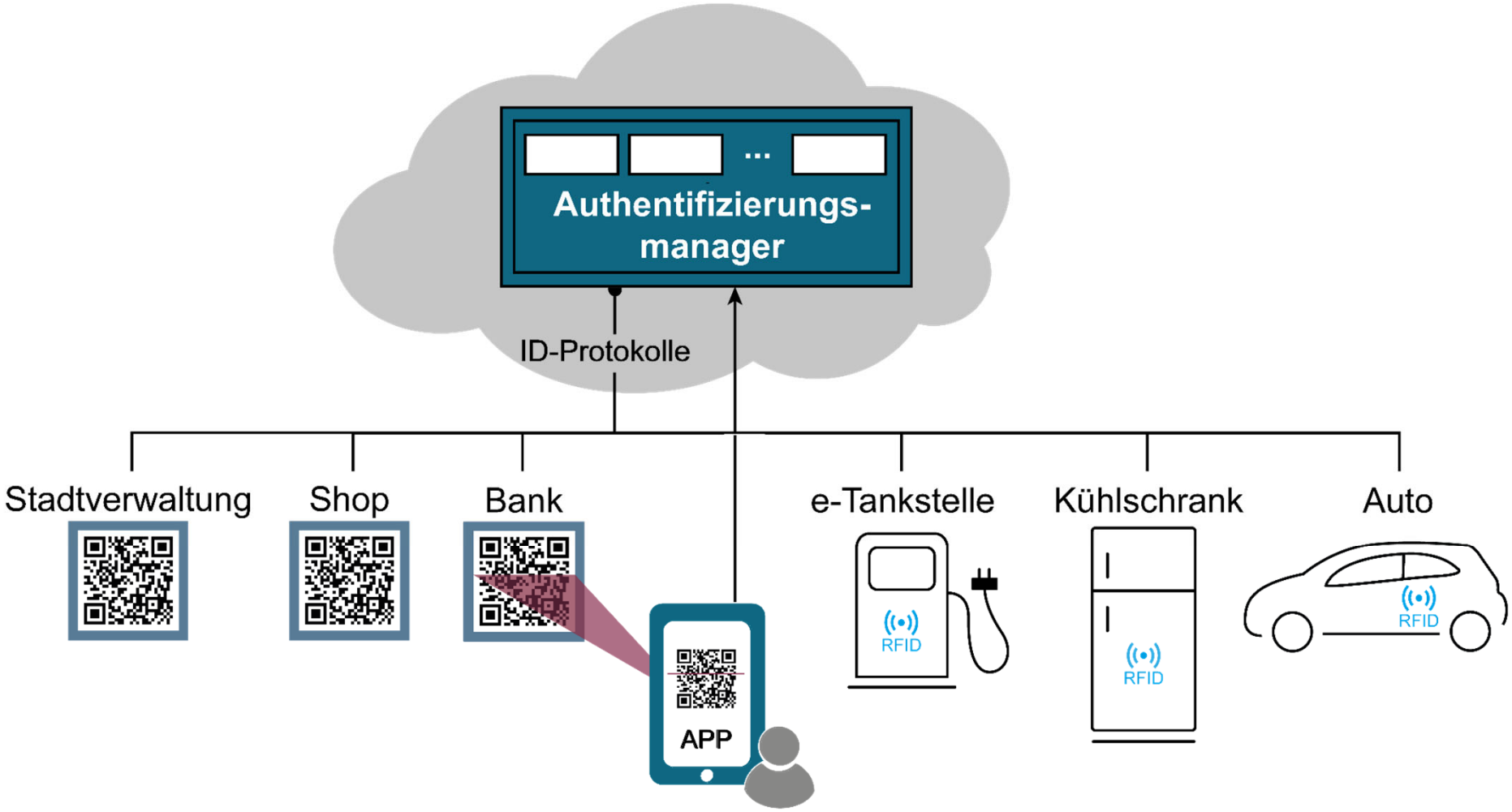
- Ziele und Ergebnisse der Vorlesung
- Identifikation und Authentifikation
- Generelle Authentifikationsverfahren
- Passwort-Verfahren
- Einmal-Passwort-Verfahren
- Challenge-Response-Verfahren
- Biometrische Verfahren
- Mehrfaktor-Authentifizierung
- **Moderne Authentifizierungssysteme**
- FIDO
- Zusammenfassung

Moderne Auth-Services
→ Anforderungen

- Moderne Authentifizierungssysteme sollten die Chancen und Herausforderungen der MFA berücksichtigen:
  - Hohe Sicherheit bei geringer Komplexität
  - Adaptive Balance zwischen Sicherheit und Nutzerfreundlichkeit
  - Einfache Integration
  - Interoperabilität und Flexibilität
  - Datenschutz und -sparsamkeit
  - Hohe Nutzerakzeptanz durch Verzicht auf zusätzlicher Hardware, Transparenz, Informationelle Selbstbestimmung und einfache Verwaltung und Nutzung

- Im Folgenden werden Konzepte für eine handhabbare und modernen Multifaktor-Authentifizierung vorgestellt.

Moderne Auth-Services
→ Konzept (1/2)



Moderne Auth-Services
→ Konzept (2/2)

- Die Trennung von Authentifizierungsmedium und Authentifizierungsmanager bringt Vorteile:
  - Kein direkter Eingriff in die bestehende IT-Infrastruktur nötig.
  - Parallelbetrieb in der eigenen IT-Infrastruktur möglich.
  - Komfortable Nutzung aus der Cloud möglich.

- Für die einfache Integration sollten standardisierte Protokolle (wie z.B. SAML, OpenID Connect, Websocket-Protokoll mit JSON-Nachrichten, LDAP, RADIUS, …) verwendet werden.

Moderne Auth-Services
→ Authentifizierungsmanager

- Als Trusted Third Party bildet der Authentifizierungsmanager die zentrale Komponente und somit den Vertrauensanker.
  - Vermittler zwischen dem Nutzer mit der APP und dem Dienstanbieter.
  - Liefert den QR-Code an den Dienstanbieter aus.
  - Bescheinigt dem Nutzer über die APP, um welchen Dienstanbieter es sich handelt und welche Informationen für die Erfüllung des Dienstes an den Dienstanbieter übermittelt werden müssen.
  - Versichert dem Dienstanbieter, dass der Nutzer ordentlich und sicher authentifiziert wird.

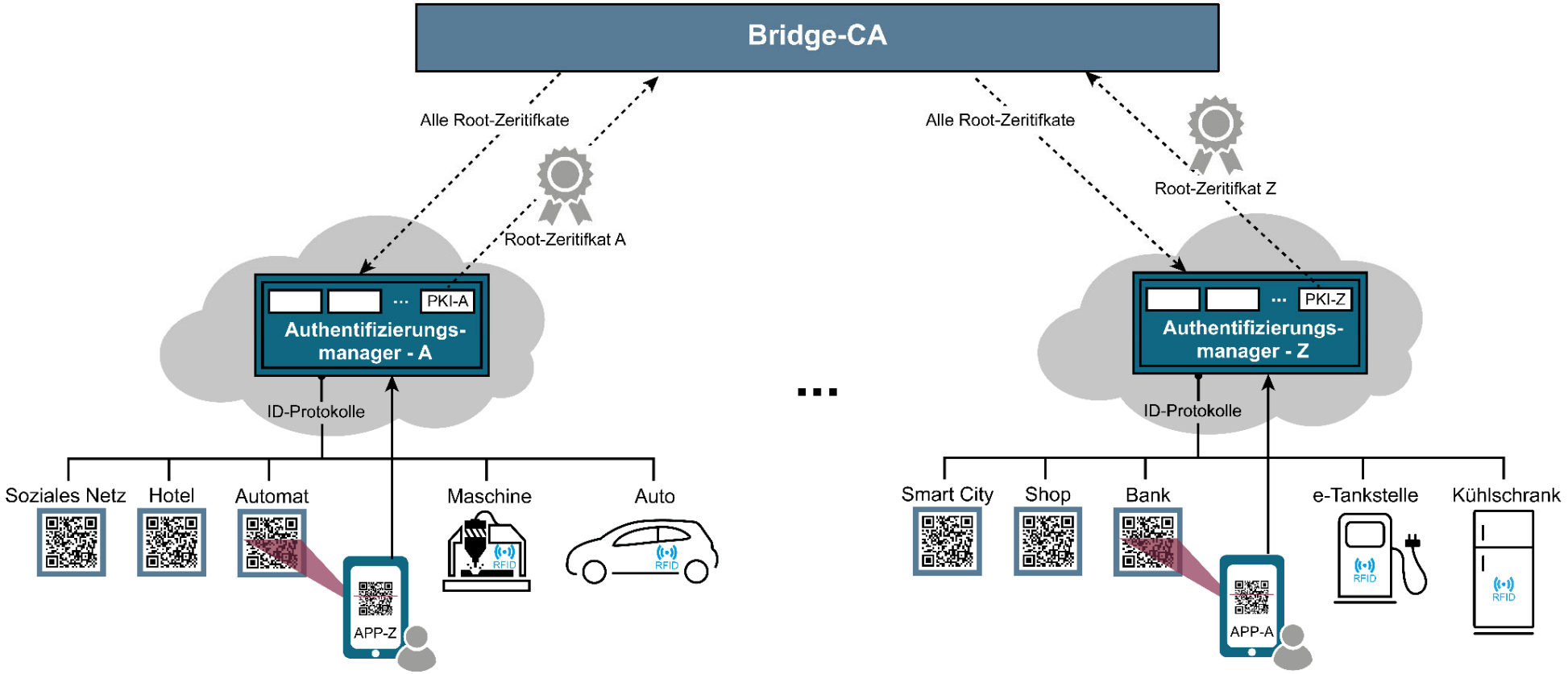
Moderne Auth-Services

→ Smartphone als Authentifizierungsdevice

- Die APP agiert als vertrauenswürdige Nutzerschnittstelle, Kontrollkanal, QR-Code Scanner, Token-Lesegerät und Software-Token.
  - Verwendung einer PIN möglich.
  - PKI-basiertes Challenge-Response-Protokoll möglich.
  - Die APP ist mit Schlüsselpaaren und den dazugehörigen Zertifikaten ausgestattet.
  - Hiermit wird das Smartphone zum Personal Authentitication Device (PAD).

- Realisierung einer sicheren nutzerfreundlichen und adaptiven Multifaktor-Authentifizierung:
  - Kombinationen aus Besitz (Smartphone), Wissen (PIN), Sein (Biometrie) möglich.

Moderne Auth-Services
→ Bridge-CA



Moderne Auth-Services
→ Vertrauensniveaus (1/3)

- Vertrauensniveau 1: Nicht verifizierte Registrierung
  - Der Nutzer gibt seine Daten persönlich ein.
  - Die Daten werden nicht weiter mit einem Vertrauensanker verifiziert und beruhen nur auf der Selbstauskunft des Nutzers.
  - Aufgrund des einfachen Vertrauensniveaus werden hier nur wenige Nutzerdaten erfasst, im einfachsten Falle handelt es sich um den Nutzernamen oder ein generiertes Pseudonym.
- Vertrauensniveau 2: E-Mail verifizierte Registrierung
  - Der Nutzer beweist seine Identität, indem er zeigt, im Besitz der angegebenen und funktionierenden E-Mail Adresse zu sein.
  - Damit wird nur die E-Mail Adresse verifiziert, was allerdings für viele Dienste, wie zum Beispiel soziale Netzwerke oder Blogs, ausreichend ist.

Moderne Auth-Services
→ Vertrauensniveaus (2/3)

- Vertrauensniveau 3: Registrierung per VideoIdent
- Vertrauensniveau 4: Registrierung mit der eID-Funktion des Personalausweises
- Vertrauensniveau 3: Registrierung per VideoIdent
  - Dieses Verfahren gibt die Möglichkeit, jederzeit, ohne Zusatzhardware, eine starke Identifizierung durchzuführen.

Moderne Auth-Services
→ Vertrauensniveaus (3/3)

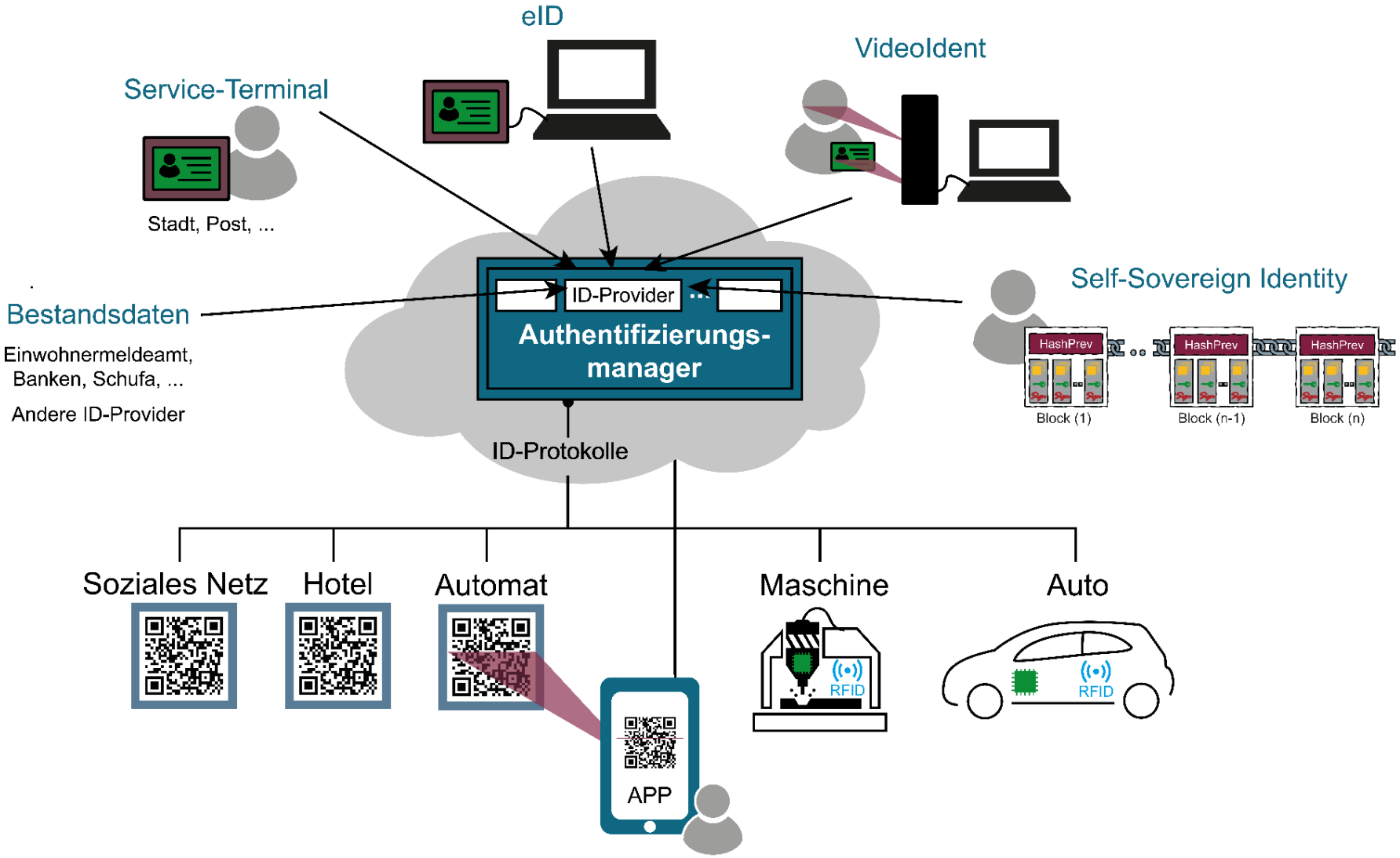
- Vertrauensniveau 4: Registrierung mit der eID-Funktion des Personalausweises
  - Die Registrierung per eID resultiert im höchsten Sicherheitslevel.
  - Diese Form des Identitätsnachweises ist besonders sicher, da es sich hier um einen rein elektronischen Nachweis mit einem sehr starken Vertrauensanker handelt.
  - Der Nutzer benötigt für die Verwendung des nPA ein Lesegerät, eine aktivierte Online-Funktion des Ausweises und eine spezielle Software auf seinem IT-System (AusweisApp2 oder OpenECardApp).

Moderne Auth-Services
→ Abgeleitete Identitäten

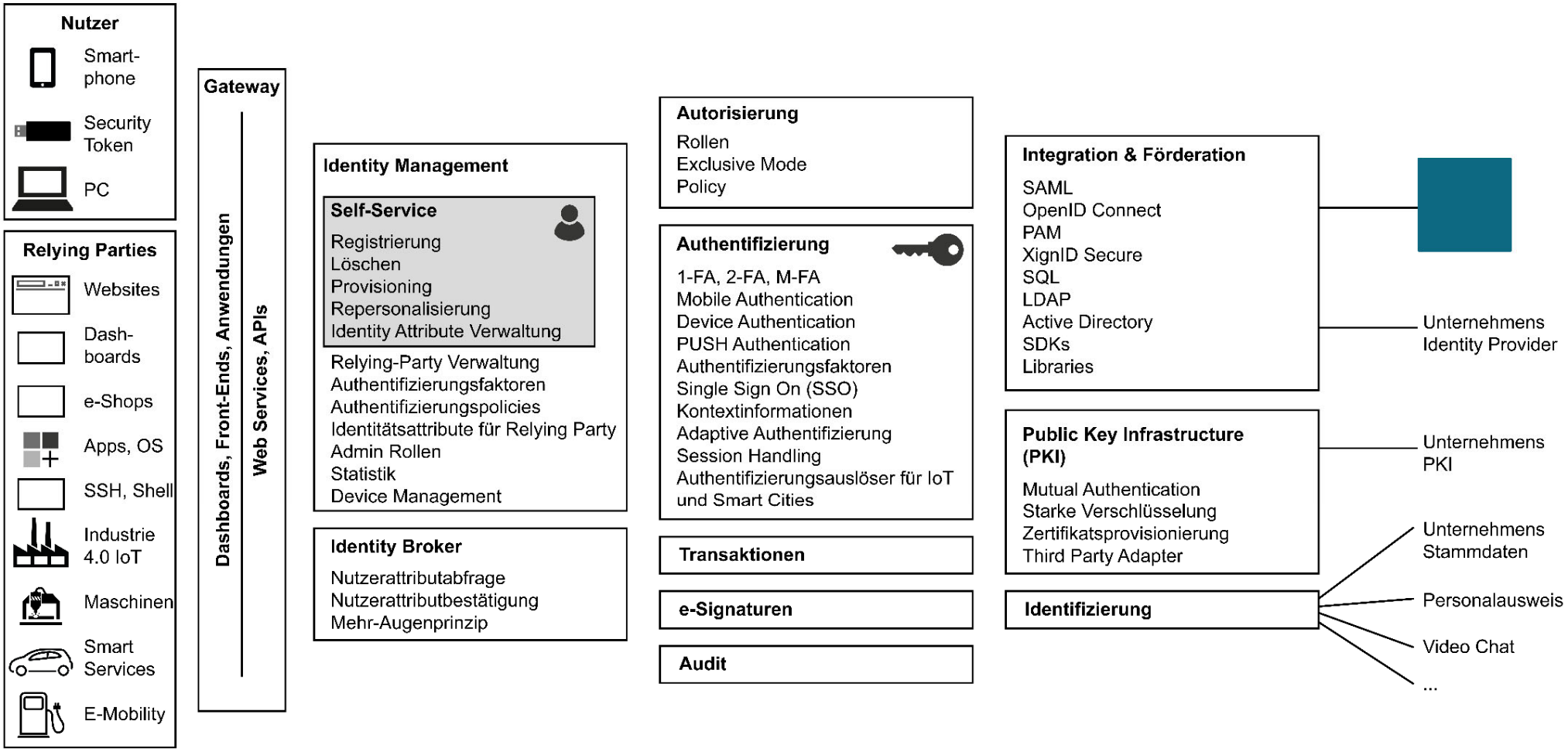
- Aus den erfassten Daten wird eine Kennung (digitale Identität oder abgeleitete Identität) erzeugt, die zur Personalisierung der APP verwendet wird.
  - Dazu wird dem Nutzer ein sehr kurzlebiger QR-Code angezeigt.
  - Nach dem Scannen des QR-Codes mit der APP, erhält der Nutzer über einen zweiten Kanal (in Abhängigkeit des Vertrauensniveaus) einen Verifizierungscode, der durch die APP verarbeitet wird.

- Ist der Verifizierungscode gültig, wird die Registrierung abgeschlossen:
  - Generieren der Schlüsselpaare
  - Erstellen und sicheres Speichern der Zertifikate
  - Festlegen der PIN

Moderne Auth-Services
→ Nutzung unterschiedlicher Identitäten



Moderne Auth-Services
→ Weitere Attribute



Moderne Auth-Services
→ Identity Provider

- Ein Identity Provider (IdP) stellt Dienstanbietern Services für die Identifikation und Authentifikation zur Verfügung.
  - Nutzer registrieren und identifizieren sich im Vorfeld bei einem IdP (z.B. Facebook, Google oder Verimi).
  - Dienstanbieter (z.B. ein Webshop oder Kundenportal) implementiert eine Schnittstelle zum IdP.
  - Für eine Anmeldung leitet der Dienstanbieter den Nutzer zum IdP weiter.
  - Der Nutzer muss sich gegenüber dem IdP authentifizieren.
  - Der Status der Authentifizierung wird dem Dienstanbieter übermittelt.

- Die Kommunikation erfolgt über standardisierte Protokolle, wie z.B. SAML, OpenID oder OAuth.

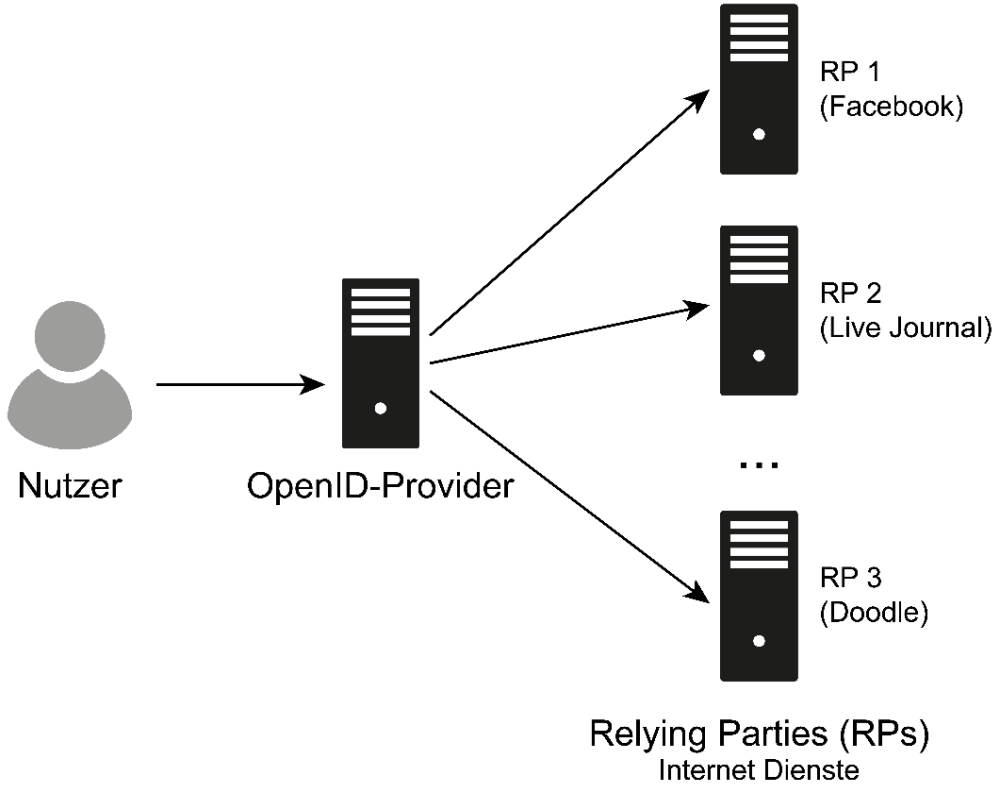
Moderne Auth-Services
→ Identity Provider

- Vorteile:
  - Die Zugangsdaten eines Nutzers werden nur noch an einer zentralen Stelle gespeichert.
  - Nutzer müssen sich weniger Zugangsdaten merken, wodurch die Passwortregeln besser beachtet werden können.
  - Internetdienste müssen sich nicht um die Implementierung und IT-Sicherheit der Authentifizierungsverfahren kümmern.
  - Internetdienste können verschiedene Methoden zur Authentisierung anbieten.

- Nachteile:
  - Single Point of Failure (z.B. erhöhte Gefahr durch DDoS-Angriffe oder Phishing).

Moderne Auth-Services
→ OpenID (1/5)

- OpenID ist ein offener Standard für Single Sign-On im Internet.
  - Der Sicherheitsdienst agiert dezentral und URL-basiert.
  - Ein Nutzer kann sowohl seine Identität als auch seinen Identity Provider frei wählen.



Moderne Auth-Services
→ OpenID (2/5)

- Zuerst muss sich ein Nutzer eine Open-ID-Identität erstellen. Hierfür sind grundsätzlich vier Schritte notwendig:

- 1. Wahl des OpenID-Providers:
  - Der Nutzer wählt einen Provider (z.B. Google, Yahoo, Amazon, …), der für die Bestätigung der digitalen Identität zuständig ist.

- 2. Wahl des Identifikators:
  - Der Nutzer wählt eine URL (die eigentliche OpenID-Identität).
  - Die URL repräsentiert die digitale Identität des Nutzers und wird den Internetdiensten anstelle eines Nutzernamens präsentiert.
  - Ein Nutzer hat fortan nicht mehr viele verschiedene Nutzernamen, sondern nur noch einen Identifikator (z.B. https://openid.internet-sicherheit.de/NorbertPohlmann).

Moderne Auth-Services
→ OpenID (3/5)

- 3. Eingabe persönlicher Informationen:
  - Wenn gewünscht, kann der Nutzer Informationen wie etwa Vor- und Zuname oder E-Mail-Adresse hinterlegen.
  - Mittels des OpenID-Protokolls kann ein Internetdienst nicht nur die Authentisierung des Nutzers anfragen, sondern optional auch weitere Informationen. (Diese muss der Nutzer in jedem Fall gesondert frei geben)

- 4. Festlegung der Zugangsdaten:
  - Bei der Registrierung der OpenID-Identität hinterlegt der Nutzer seine Zugangsdaten.
  - Das ist für gewöhnlich eine Kombination aus Nutzername und Passwort oder weitere Faktoren.

Moderne Auth-Services
→ OpenID (4/5)

- Nachdem die OpenID-Identität angelegt wurde, können sämtliche unterstützenden Internetdienste genutzt werden.

- **1. Aufruf der Login-Seite:**
  - Die Webseite mit dem Login-Formular des unterstützenden Internetdienstes wird aufgerufen.

- **2. Behauptung der Identität:**
  - Statt wie gewohnt eine Nutzername-Passwort-Kombination einzugeben, übermittelt der Nutzer lediglich die OpenID-Identität.

- **3. Beweis der behaupteten Identität:**
  - Der Internetdienst leitet den Nutzer zu dem entsprechenden OpenID-Provider weiter.
  - Der Nutzer führt die Authentifizierung auf Basis der festgelegten Faktoren durch.

Moderne Auth-Services
→ OpenID (5/5)

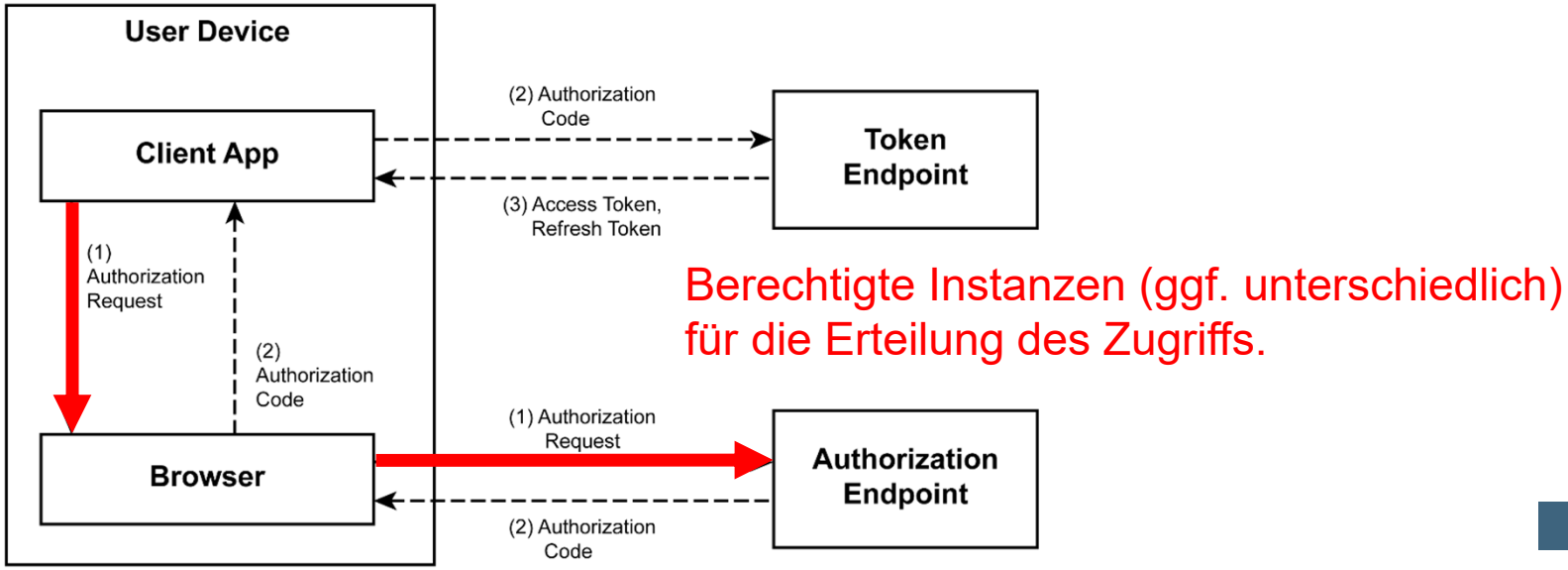
- **4. Nutzung des Internetdienstes:**
  - Sollte die Antwort des OpenID-Providers positiv ausfallen, kann der Internetdienst die Identität des Nutzers als bestätigt ansehen und die Nutzung freigeben.

Moderne Auth-Services
→ OAuth 2.0 (1/4)

- Protokoll zur Autorisierung von Zugriffen auf schützenswerte Ressourcen über verschiedene Instanzen hinweg.
  - Ein erfolgter Autorisierungsnachweis kann ebenfalls als eine Pseudo-Authentifizierung betrachtet werden.
  - Für eine höhere Cyber-Sicherheit sollte OAuth 2.0 jedoch grundsätzlich um ein zusätzliches Protokoll für die Authentifizierung ergänzt werden.
  - Hierfür kann beispielsweise der offene Standard OpenID Connect verwendet werden.

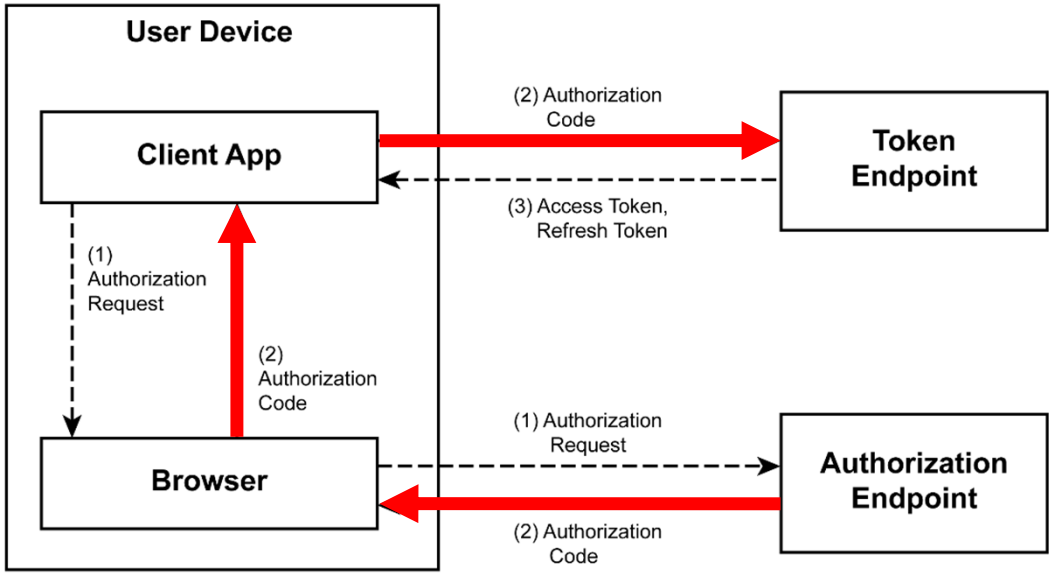
Moderne Auth-Services
→ OAuth 2.0 (2/4)

- **(1) Authorization Request:**
  - Die Anwendung auf dem Endgerät öffnet eine Browsersession zwischen der Anwendung und dem „Authorization Endpoint“. („Best Current Practice“ nach RFC 8252)
  - Innerhalb dieser Session wird die Autorisierung realisiert. (Es sollte eine zusätzliches Authentifizierung verwendet werden)



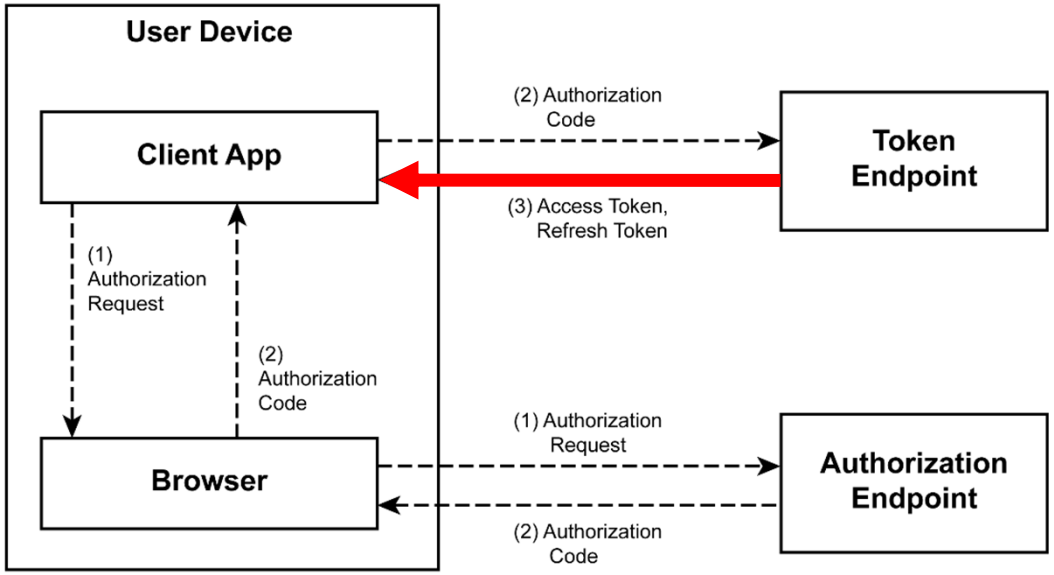
Moderne Auth-Services
→ OAuth 2.0 (3/4)

- **(2) Authorization Code:**
  - Nach erfolgreicher Autorisierung und ggf. Authentifizierung wird dem Browser ein Autorisierungscode zur Verfügung gestellt.
  - Mit Hilfe dieses Codes kann nachgewiesen werden, dass eine Autorisierung erfolgreich durchgeführt wurde.
  - Mit dem Autorisierungscode muss ein Zugriffscode angefordert werden.



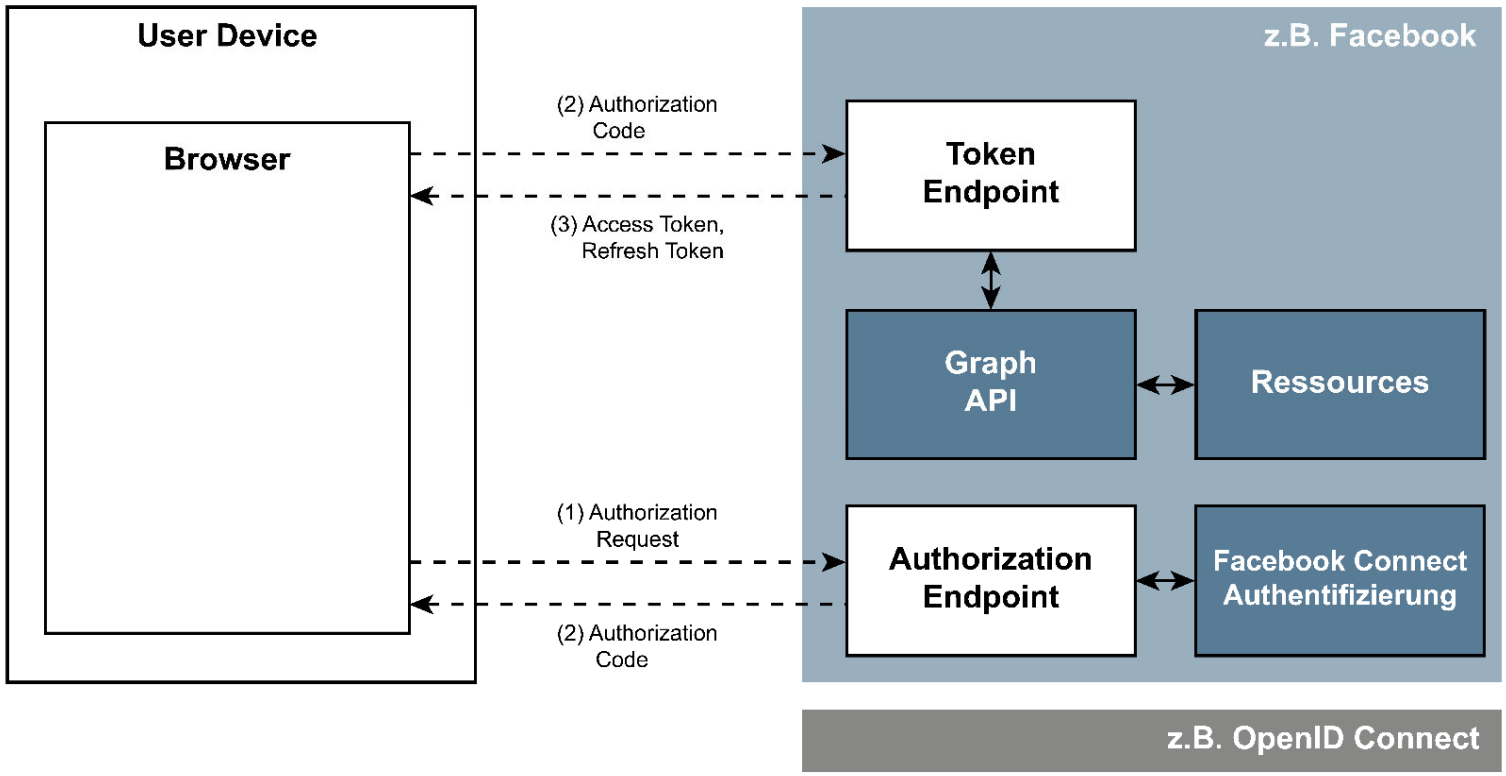
Moderne Auth-Services
→ OAuth 2.0 (4/4)

- **(3) Access Token, Refresh Token:**
  - Der Autorisierungscode wird zuerst auf Gültigkeit überprüft.
  - Anschließend wird der Anwendung ein Zugriffscode ausgestellt, mit dem der Zugriff auf die Ressource erfolgen kann.
  - Ein „Refresh Token“ wird auf ähnliche Weise immer dann angefordert, wenn der Zugriffscode abgelaufen ist.



Moderne Auth-Services
→ Facebook Connect (1/6)

- Anwendungsbeispiel: Anmeldung bei Stack Overflow mittels Facebook Connect
  - Basiert auf Oauth 2.0.
  - Proprietäre Alternative zu dem offenen Standard OpenID Connect.



Moderne Auth-Services
→ Facebook Connect (2/6)

- Für die Anmeldung bei Stack Overflow wird der Nutzer auf die Webanwendung von Facebook umgeleitet.

- Dem HTTP-Request wird folgendes angehängt:
  - Eindeutige ID der Webanwendung von Stack Overflow.
  - Benötigte Berechtigung (E-Mail-Adresse des Nutzers)
  - Zieladresse für die Weiterleitung nach der Authentifizierung.
  - Zustandsobjekt gegen CSRF-Angriffe und für die Wiederherstellung eines Kommunikationszustandes.

Moderne Auth-Services
→ Facebook Connect (3/6)

```
HTTP/1.1
GET https://www.facebook.com/v2.0/dialog/oauth?
    client_id=145044622175352
    &scope=email
    &redirect_uri=https://stackauth.com/auth/oauth2/facebook
    &state=
        {
        "sid":1,
        "st":"a7b1972f33fdea4620e3276001a2b33ce9c1d5f9bc63227ecab767c1bd2617bd",
        "ses":"1c2bdda2ed7f4322952baf4ef2cbbd66"
        }
```

- Nach der erfolgreichen Authentifizierung und Autorisierung wird der Nutzer auf die Webanwendung von Stack Overflow zurückgeleitet.
- An den HTTP-Request wird der Autorisierungscode und ein aktualisiertes Zustandsobjekt angehängt.

Moderne Auth-Services
→ Facebook Connect (4/6)

```
HTTP/1.1
GET https://stackauth.com/auth/oauth2/facebook?
    code=AQBxewj3LeZBc4NzUJv7FFd0fDfB2UM5jyfeX5Cac2NaaxQMRxsfY03TDCcdaj...
    &state=
        {
        "sid":1,
        "st":"2458ff31cc4291023c2c301aa58dd6a4938a0bcfa5066d229a77c1f8abc09e51",
        "ses":"d4bda5e797a94c858d8a40edafc431f5"
        }
```

- Mit dem Autorisierungscode kann ein Zugriffscode bei der Graph API von Facebook angefragt werden.
- Hierfür muss wieder die ID der Webanwendung von Stack Overflow an den HTTP-Request angehängt werden.
- Der Zugriffscode wird serverseitig von Stack Overflow abgerufen. Aus diesem Grund muss Stack Overflow das eigene Passwort angeben.

Moderne Auth-Services
→ Facebook Connect (5/6)

```
HTTP/1.1
GET https://graph.facebook.com/v3.1/oauth/access_token?
    client_id=145044622175352
    &redirect_uri=https://stackauth.com/auth/oauth2/facebook
    &client_secret=<PASSWORD>
    &code=AQBxewj3LeZBc4NzUJv7FFd0fDfB2UM5jyfeX5Cac2NaaxQMRxsfY03TDCcdaj...
```

- Nach der erfolgreichen Validierung des Autorisierungscodes, wird der Webanwendung ein Zugriffscode in dem folgenden Format von der Graph API zur Verfügung gestellt:

```
{
"access_token": "<ACCESS-TOKEN>",
"token_type": "<TYPE>",
"expires_in": <SECONDS-TIL-EXPIRATION>
}
```

Moderne Auth-Services
→ Facebook Connect (6/6)

- Mit dem Zugriffscode könnte die Webanwendung nun auf die freigegebenen Ressourcen der Graph API zu dem Nutzer zugreifen.
- In diesem Anwendungsbeispiel ist das lediglich die bei Facebook hinterlegte E-Mail-Adresse des Nutzers.
- Diese kann anschließend verwendet werden, um beispielsweise erstmalig einen neuen Account bei Stack Overflow zu dieser E-Mail-Adresse zu erstellen oder um den Login durchzuführen.

Moderne Auth-Services
→ OpenID Connect (1/3)

- Bei OpenID Connect handelt es sich um die dritte und aktuelle Generation des OpenID Protokolls.
  - Es basiert im Wesentlichen auf dem verbreiteten OAuth 2.0 Protokoll und erweitert dieses um fehlende Identity Services, speziell um Protokollabläufe für die Authentifikation.

- Aufbauend auf dem OAuth 2.0 Protokoll wird über das Attribut „scope=openid“ die Authentifikation gestartet.

```
HTTP/1.1
GET https://accounts.login.idm.telekom.com/oauth2/auth?
    client_id=10LIVESAM30000004901VESPAPICOTELEKOM0000
    &scope=openid
    &redirect_uri=https://www.telekom.de/tech/sam/ess/callback
    &state=f3a34ae4-80eb-44d9-84f1-a942aafb67f8
    &response_type=code
```

Moderne Auth-Services
→ OpenID Connect (2/3)

- Die weiteren Protokollabläufe erfolgen analog zu den beschriebenen Abläufen von Facebook Connect.

- Der abschließend erhaltene Access Token enthält jedoch bei der Verwendung von OpenID Connect ein weiteres Attribut.

```
{
"access_token": "<ACCESS-TOKEN>",
"token_type": "<TYPE>",
"expires_in": <SECONDS-TIL-EXPIRATION>
"id_token": "<JSON-WEB-TOKEN>"
}
```

- Hierbei handelt es sich um einen JSON Web Token (JWT), der wichtige Informationen zu der durchgeführten Authentifikation des Benutzers enthält.

Moderne Auth-Services
→ OpenID Connect (3/3)

```
{
"sub": "<USER-ID>",
"iss": "<ISSUING-AUTHORITY>",
"aud": "<AUDIENCE-RESTRICTION>",
"nonce": "<ANTI-REPLAY-VALUE>",
"auth_time": <AUTHENTICATION-TIME>,
"acr": <AUTHENTICATION-CONTEXT>,
"iat": <ISSUING-TIME>,
"exp": <EXPIRATION-TIME>
}
```

- Basierend auf den enthaltenen Informationen kann eine Anwendung
  - den zuvor authentifizierten Benutzer identifizieren,
  - einen neuen Account für ihn erstellen oder
  - ihn mit einem bereits bestehenden Account einloggen.

Identifikation und Authentifikation
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Identifikation und Authentifikation
- Generelle Authentifikationsverfahren
- Passwort-Verfahren
- Einmal-Passwort-Verfahren
- Challenge-Response-Verfahren
- Biometrische Verfahren
- Mehrfaktor-Authentifizierung
- Moderne Authentifizierungssysteme
- **FIDO**
- Zusammenfassung

FIDO
→ Wer oder was ist die FIDO Alliance?

- FIDO steht für Fast Identity Online

- FIDO Alliance besteht aus mehreren Mitglieder unter anderem:
  - Google
  - Microsoft
  - Lenovo
  - PayPal,
  - Visa
  - MasterCard
  - NXP
  - Nok Nok Lab
  - …

FIDO
→ Ziele der FIDO Alliance

- Bereitstellung einer starken multifaktor Authentikation
- Wahlmöglichkeiten zwischen verschiedenen Authentikationsmechanismen
- Vereinfachung der Integration neuer Authentikationsmechanismen
- Erweiterbarkeit
- Verwendung offener Standards (wenn möglich)
- Entwicklung neuer offener Standards (wenn notwendig)
- Datenschutz
- Benutzerkomfort

FIDO
→ UAF vs. U2F (1/2)

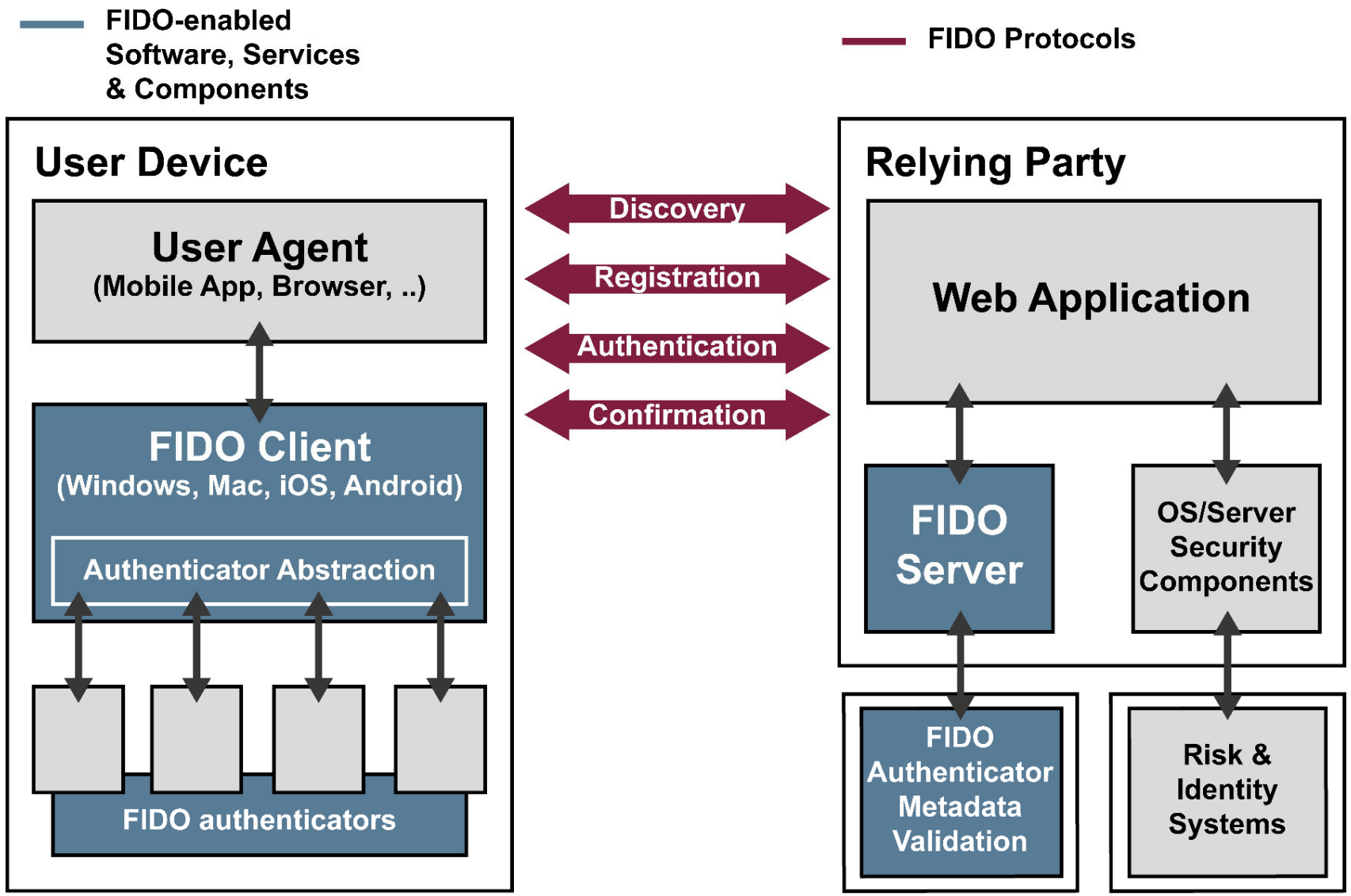
- FIDO Alliance stellt zwei Spezifikationen bereit:
  - Universal Authentication Framework (UAF)
  - Universal 2nd Factor (U2F)

- **UAF:**
  - Ziel: Bereitstellung passwortloser und multifaktor Sicherheit für Online-Dienste
  - Der User kann einen vorhandenen Auth-Mechanismus wählen und mit einem Online-Dienst registrieren
  - Nach der Registrierung kann der entsprechende Auth-Mechanismus für die Anmeldung beim Dienst verwendet werden
  - UAF erlaubt eine Filterung der verwendbaren Auth-Mechanismen (=> Vertrauen in bestimmte Mechanismen)

FIDO
→ UAF vs. U2F (2/2)

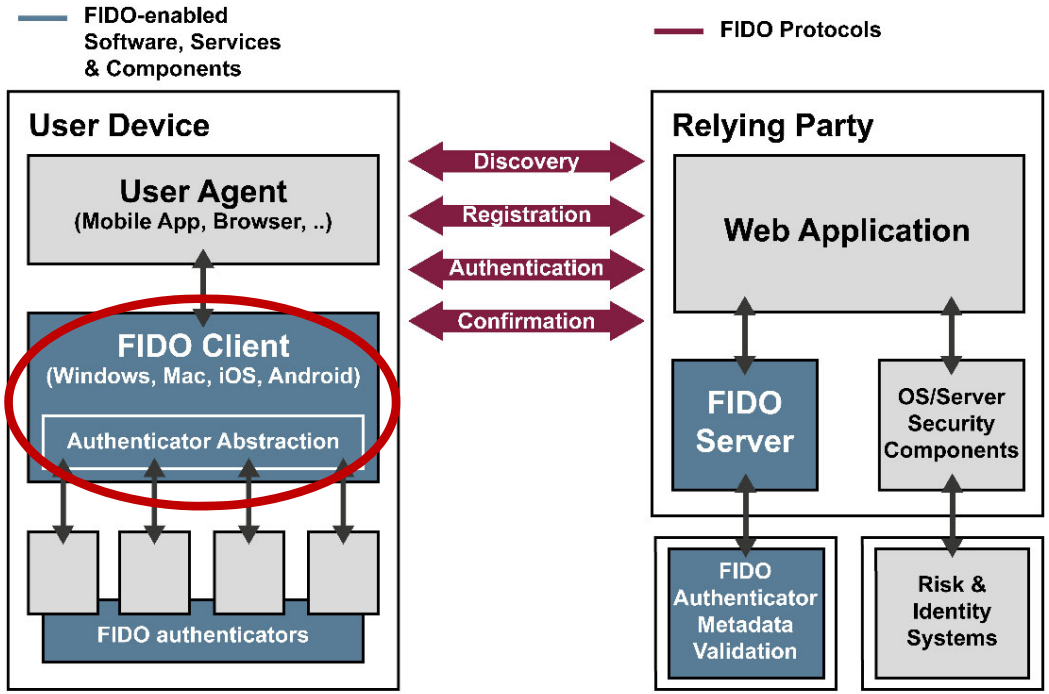
- **U2F**
  - Ziel: Verbesserung der Sicherheit eines Online-Dienstes durch zusätzliche Zwei-Faktor-Authentikation
  - Der User kann sich normal mit seinem gewohnten Mechanismus (Benutzername/Passwort) einloggen
  - Der Online-Dienst kann zu jeder Zeit einen Token (2nd Factor Device; NFC oder USB) vom Benutzer verlangen für weitere Authentikation verlangen
    - Der 2nd Factor muss dementsprechend registriert werden

FIDO
→ Architektur



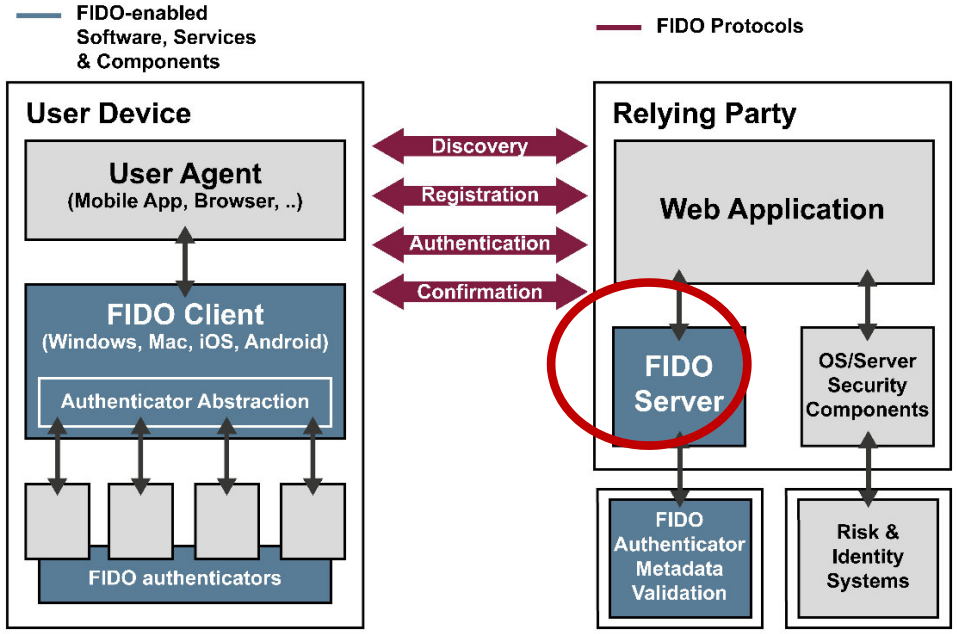
FIDO
→ FIDO Client

- Realisiert Client-Seite der FIDO Protokolle auf dem Gerät des Benutzers
- Interagiert mit Authenticator und User-Agent auf dem Gerät
- empfängt UAF-Protokoll-Nachrichten vom FIDO Server
- Browser-Plugin für Desktop-PCs
- Android Service
- …



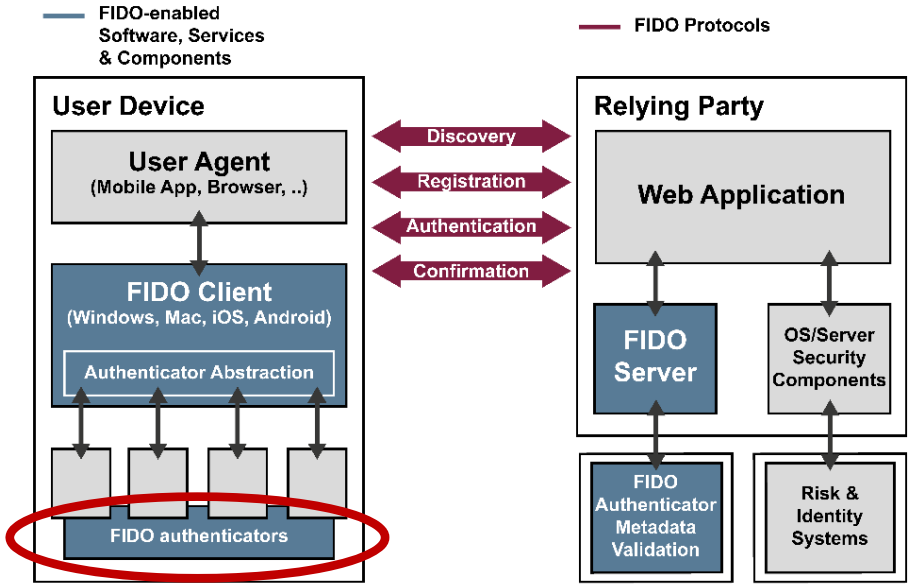
FIDO
→ FIDO Server

- Interagiert mit Online-Diensten
- Sendet UAF-Protokoll-Nachrichten an den FIDO-Client
- Valdiert UAF-Protokoll-Antworten
- Verwaltet FIDO-Benutzerdaten und kennt UserID des Benutzers im Online-Dienst
- Steuert die Filterung der Authentikatoren
- Als Service oder Standalone aufgestellt



FIDO
→ FIDO Authenticator (1/2)

- Sichere Entität, die auf dem Gerät des Benutzers vorhanden oder dazu verbunden ist
- Führt die Authentifizierung des Benutzers durch
- Kommuniziert mit Peripherie des Geräts (WebCam, NFC-Reader, Fingerabdruck-sensor) um den User zu authentifizieren
- Kann mit externen Services kommunizieren, um den User zu authentifizieren

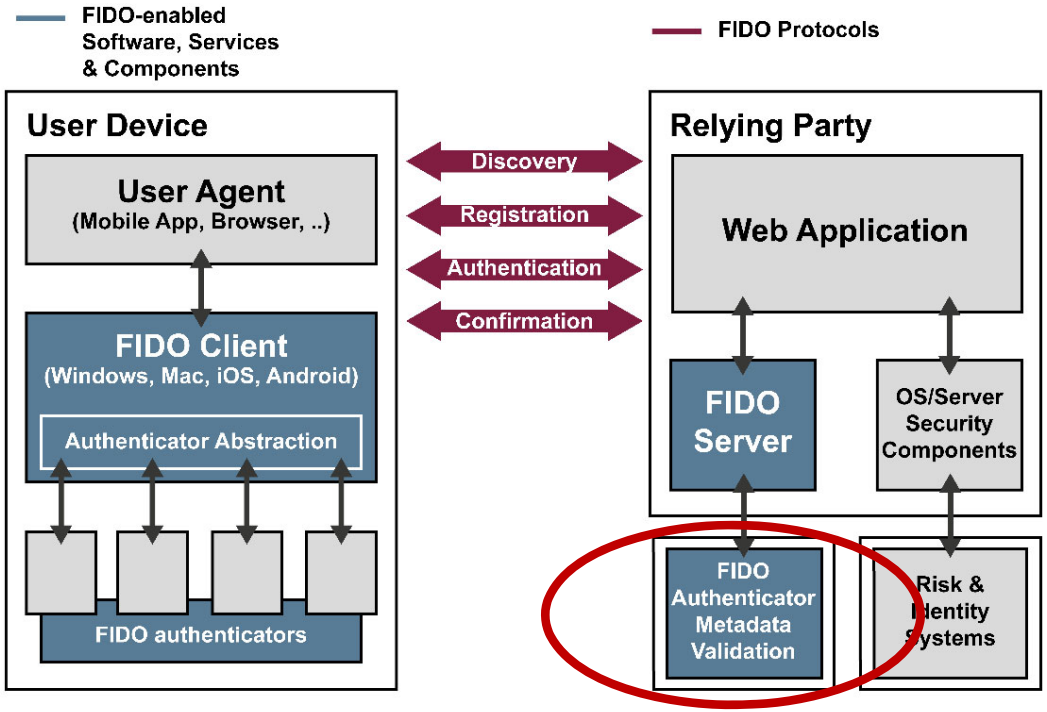


FIDO
→ FIDO Authenticator (2/2)

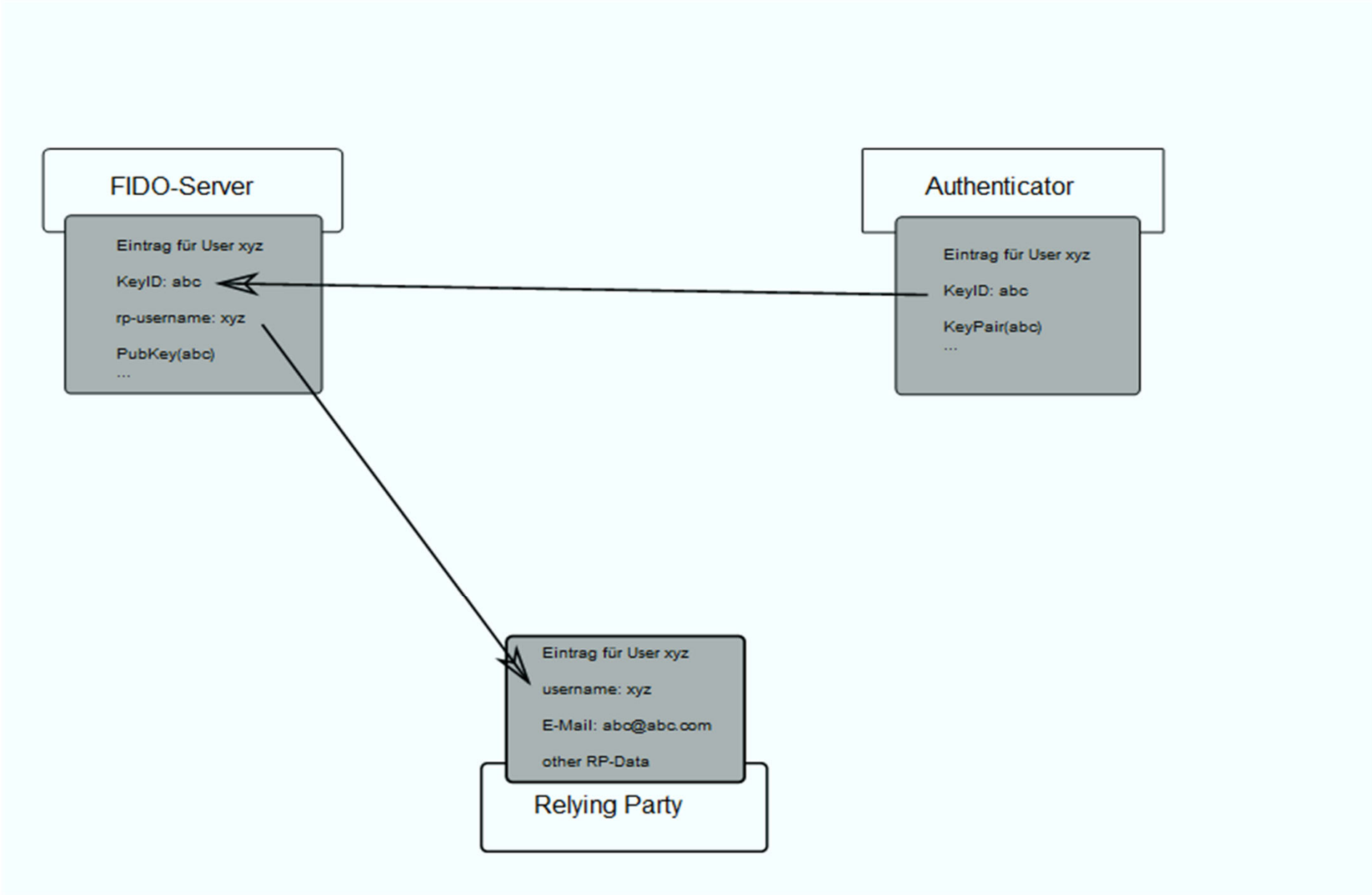
- Generiert Schlüsselmaterial für Nutzer
- Signiert vom FIDO-Server übermittelte Challenges
- Bereitgestellt als:
  - Dynamic Link Library (DLL; Windows)
  - Share Object (.so; Linux)
  - Service (Smartphone)

FIDO
→ Meta-Daten

- Informationen über die bekannten und vertrauten Authenticatoren (IDs, Fähigkeiten etc.)
- IDs der Authentikatoren werden von der FIDO Alliance vergeben (=> nur vertraute Authentikatoren können verwendet werden)
- Bilden die Grundlage für die Filterung der Auswahlmöglichkeiten des Benutzers



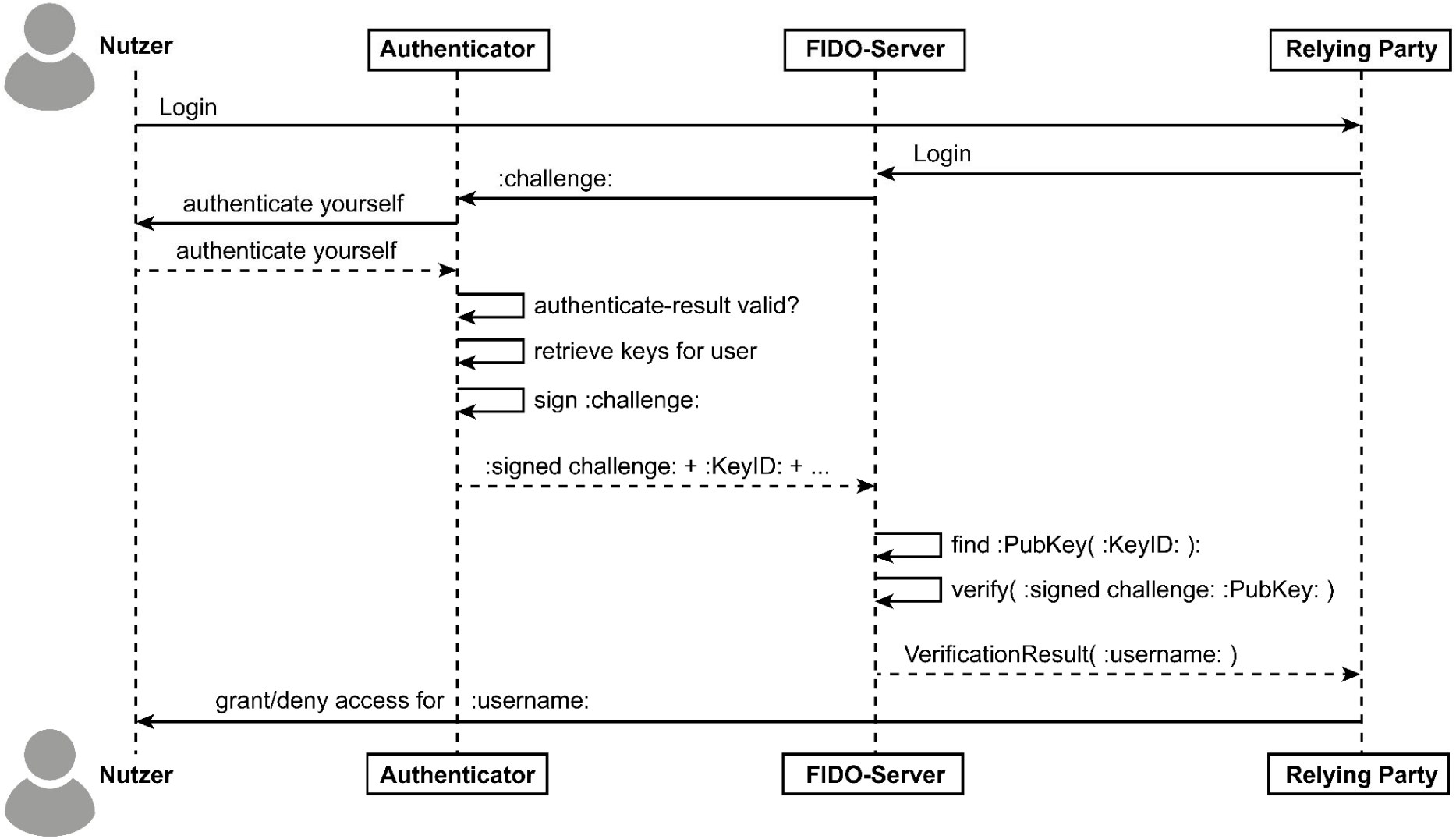
FIDO
→ Identifikation des Benutzers (1/2)



FIDO
→ Identifikation des Benutzers (2/2)

- Während der Registrierung speichert der Authentikator nur die benutzer-bezogenen Daten, die wichtig für die spätere Authentifizierung sind
  - KeyID (vom Authentikator generiert)
  - Schlüsselmaterial des Benutzers
  - Je nach Implementierung authentikator-spezifische Daten
- KeyID ist in der Datenbank des FIDO Servers mit weiteren Benutzerdaten assoziiert und wird für das Auffinden des Users verwendet
- Benutzerdaten enthalten unter anderem die BenutzerID des Users im Online-Dienst
  - der FIDO Server kann dem Online-Dienst das Authentkations-Ergebnis für den entsprechenden User mitteilen

FIDO
→ Authentifizierung des Benutzers (1/3)



FIDO
→ Authentifizierung des Benutzers (2/3)

- der Benutzer wird zweimal authentifiziert:
  - Lokal durch den Authenticator
  - Über Challenge-/Response-Verfahren durch FIDO Server
- Beim Login via FIDO UAF übermittelt der FIDO Server eine Challenge an den FIDO Client
- Der User authentifiziert sich lokal gegen den Authentikator
- Bei erfolgreicher Authentifizierung schaltet der Authentikator das Schlüsselmaterial des jeweiligen Benutzers frei und bildet die Signatur zur übermittelten Challenge

FIDO
→ Authentifizierung des Benutzers (3/3)

- Die generierte Signatur, die verwendete KeyID und Challenge werden an den Server übertragen

- Der Server lokalisiert über die KeyID den entsprechenden PubKey, verifiziert die Signatur (valide => Auth-Erfolg) und übermittelt das Ergebnis zusammen mit der UserID des Users an den Online-Dienst

FIDO
→ Besonderheiten

- Authentifizierung gegenüber dem FIDO Server und somit des Online-Dienstes ist standardisiert über ein Challenge-/Response-Verfahren

- FIDO-Client-/Authenticator-Specific-Module-Fuktionalität ist standardisiert
  - nur in Spezialfällen ist es wirklich notwendig spezielle Komponenten mit **erweiteter** Funktionalität zu implementieren

- Authentifizierung gegenüber dem Authentikator ist vom Hersteller abhängig
  - Auf welche Art und Weise der User vom Authentikator authentifiziert wird geht über die Spezifikation hinaus

FIDO
→ UAF Protokolle

- FIDO Protokolle dienen dem Transport der Informationen zwischen den einzelnen Beteiligten

- Insgesamt gibt es 4 Arten:
  - Registration (Auffinden und Registrierung von Authentikatoren bei Online-Diensten)
  - Authentication (Authentifizierung eines Benutzers)
  - Confirmation (neben Authentifizierung zusätzliche Bestätigung einer bestimmten Transaktion)
  - Deregistration (De-Registrierung)

Identifikation und Authentifikation
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Identifikation und Authentifikation
- Generelle Authentifikationsverfahren
- Passwort-Verfahren
- Einmal-Passwort-Verfahren
- Challenge-Response-Verfahren
- Biometrische Verfahren
- Mehrfaktor-Authentifizierung
- Moderne Authentifizierungssysteme
- FIDO
- **Zusammenfassung**

Identifikation und Authentifikation
→ Zusammenfassung (1/2)

- Authentifikationsverfahren sind die Grundlage für die Identifikation und Authentifikation von Nutzern.
- Zunehmend wird es wichtiger, Authentifikationsverfahren zu verwenden, die in der globalen handelnden Gesellschaft über staatliche Grenzen und Verantwortungsbereiche hinaus verwendet werden können.
- Aktuell sorgen viele Dienstanbieter selber dafür, auf welche Weise ein Nutzer identifiziert und authentifiziert wird (z.B. Identifizierung per E-Mail oder Nutzung von Onlinefunktionalitäten bestimmter Ausweisdokumente, wie zum Beispiel dem neuen deutschen Personalausweis).
- Die Identifikation und Authentifikation wird zunehmend von zentralen Identity Providern gekapselt.

Identifikation und Authentifikation
→ Zusammenfassung (2/2)

- 2FA wird zunehmend als Sicherheitsmechanismus verwendet.

- In der Zukunft werden adaptive und risikobasierte Authentifikationsverfahren genutzt, die gleichzeitig ein Höchstmaß an Sicherheit und Nutzerfreundlichkeit bereitstellen.

Westfälische Hochschule
Gelsenkirchen Bocholt Recklinghausen
University of Applied Sciences

Identifikation und Authentifikation

- Vorlesung Cyber-Sicherheit -

Prof. Dr. (TU NN)
Norbert Pohlmann

Institut für Internet-Sicherheit – if(is)
Westfälische Hochschule, Gelsenkirchen
http://www.internet-sicherheit.de

if(is) internet-sicherheit.

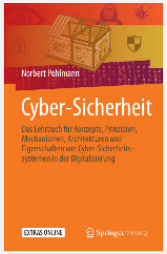
Anhang / Credits

Wir empfehlen

- **Kostenlose App securityNews**

securityNews

- **7. Sinn im Internet (Cyberschutzraum)**
  https://www.youtube.com/cyberschutzraum
- **Master Internet-Sicherheit**
  https://it-sicherheit.de/master-studieren/
- **Cyber-Sicherheit**
  Das **Lehrbuch** für Konzepte, Mechanismen, Architekturen und Eigenschaften von Cyber-Sicherheitssystemen in der Digitalisierung“, Springer Vieweg Verlag, Wiesbaden 2019
- https://norbert-pohlmann.com/cyber-sicherheit/



Besuchen und abonnieren Sie uns :-)

WWW
https://www.internet-sicherheit.de

Facebook
https://www.facebook.com/Internet.Sicherheit.ifis

Twitter
https://twitter.com/_ifis

YouTube
https://www.youtube.com/user/InternetSicherheitDE/

Prof. Norbert Pohlmann
https://norbert-pohlmann.com/

Quellen Bildmaterial

Eingebettete Piktogramme:
- Institut für Internet-Sicherheit – if(is)

Der Marktplatz IT-Sicherheit

(IT-Sicherheits-) Anbieter, Lösungen, Jobs, Veranstaltungen und Hilfestellungen (Ratgeber, IT-Sicherheitstipps, Glossar, u.v.m.) leicht & einfach finden.
https://www.it-sicherheit.de/

Literatur
→ Artikel / Bücher

M. Hertlein, P. Manaras, N. Pohlmann: „Die Zeit nach dem Passwort - Handhabbare Multifaktor-Authentifizierung für ein gesundes Eco-System", DuD Datenschutz und Datensicherheit – Recht und Sicherheit in Informationsverarbeitung und Kommunikation, Vieweg Verlag, 4/2016
https://norbert-pohlmann.com/wp-content/uploads/2016/05/345-Die-Zeit-nach-dem-Passwort-Handhabbare-Multifaktor-Authentifizierung-für-ein-gesundes-Eco-System-Prof.-Norbert-Pohlmann.pdf

J.-H. Frintrop, N. Pohlmann, R. Widdermann, T. Ziegler: „Wenn der Softbot menschliche Identität bestätigt – VideoIdent-Verfahren: Die Technik“, Die Bank – Zeitschrift für Bankpolitik und Praxis, Bank-Verlag, Köln 06/2017
https://norbert-pohlmann.com/wp-content/uploads/2017/07/358-Wenn-der-Softbot-menschliche-Identität-bestätigt-–-VideoIdent-Verfahren-Die-Technik-Prof.-Norbert-Pohlmann.pdf

M. Hertlein, P. Manaras, N. Pohlmann: „Smart Authentication, Identification and Digital Signatures as Foundation for the Next Generation of Eco Systems”, In the Book ”Digital Marketplaces Unleashed”, Editors: Claudia Linnhoff-Popien, Ralf Schneider and Michael Zaddach, Springer-Verlag GmbH Germany, 2017
https://norbert-pohlmann.com/wp-content/uploads/2019/07/362-Smart-Authentication-Identification-and-Digital-Signatures-as-Foundation-for-the-Next-Generation-of-Eco-Systems-Prof.-Norbert-Pohlmann.pdf

N. Pohlmann, A. Stöhr: „Smartphone Bürger-ID – IT-Sicherheit als Wegbereiter für die Digitalisierung“, IT-Sicherheit – Fachmagazin für Informationssicherheit und Compliance, DATAKONTEXT-Fachverlag, 5/2019
https://norbert-pohlmann.com/wp-content/uploads/2019/08/403-Smartphone-Bürger-ID-IT-Sicherheit-als-Wegbereiter-für-die-Digitalisierung-Prof-Norbert-Pohlmann.pdf

N. Pohlmann: "Cyber-Sicherheit – Das Lehrbuch für Konzepte, Mechanismen, Architekturen und Eigenschaften von Cyber-Sicherheitssystemen in der Digitalisierung“, ISBN 978-3-658-25397-4; 594 Seiten, Springer-Vieweg Verlag, Wiesbaden 2019
https://norbert-pohlmann.com/cyber-sicherheit/