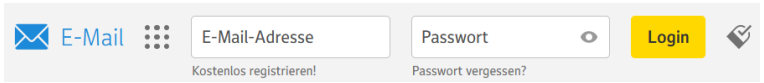
IT-Sicherheit
WiSe 2021/22

Protokolle und Infrastruktur (1)

Motivation

- **Bisher:** Kryptographische Verfahren ohne Anwendungskontext
  - Aber: Wie werden sie in der Praxis eingesetzt und was gibt es zu beachten?

- **Anwendungsbeispiel:** Alice möchte sich bei eMail Provider einloggen
  - Teil 1: Nutzer*innen-seitige Authentifikation
  - Teil 2: Server-seitige Authentifikation
  - Teil 3: Erstellung eines sicheren Kommunikationskanals



Passwort?

Öffentlicher Schlüssel K_E^S?

RSA und AES?

RSA Schlüsselpaar $\left(K_E^S, K_D^S\right)$

Inhalt des Kapitels

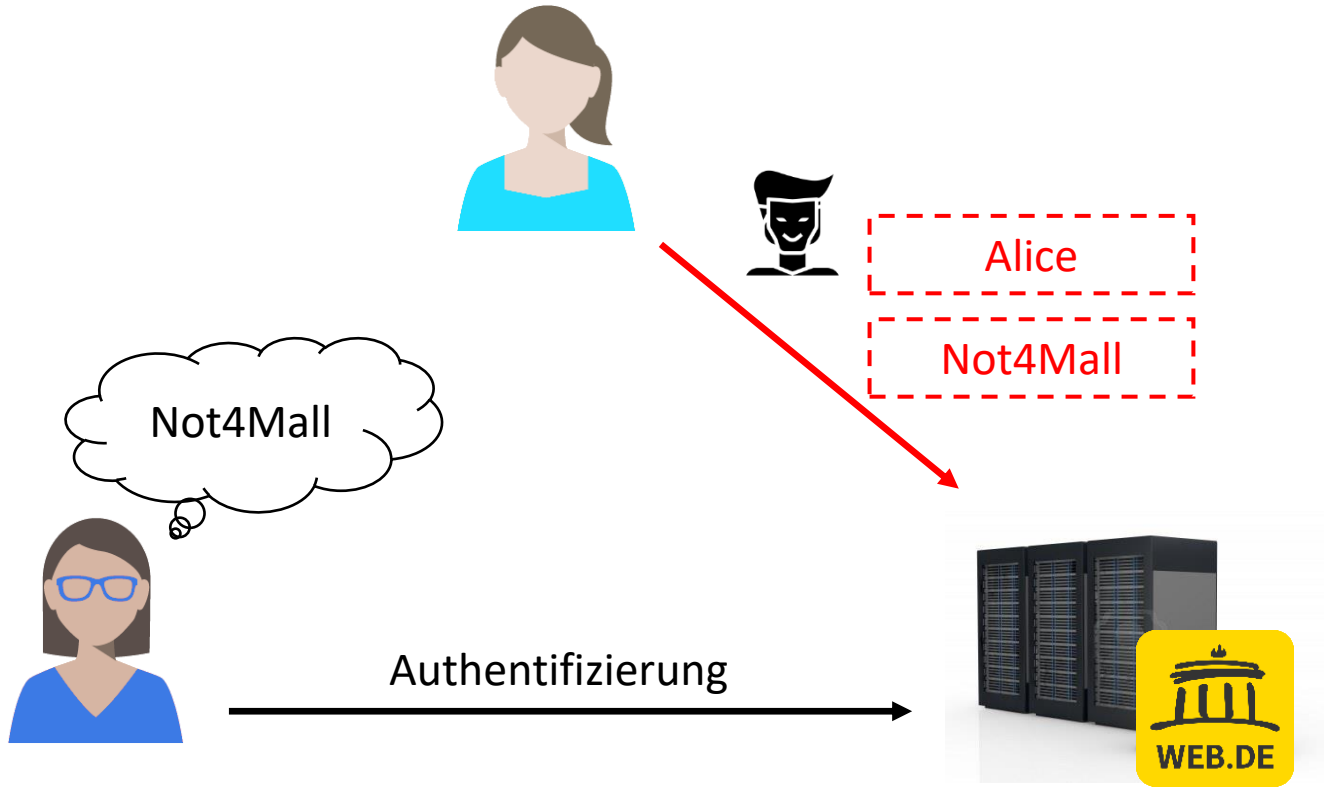
1. **Nutzer*innen-Seitige Authentifikation**
   a) Authentifikationsverfahren (Passwörter, Besitz, Biometrie, Zwei/Mehrfaktor)
   b) Challenge / Response
   c) Single Sign-On (Kerberos) und Identity Provider (OAuth)
2. Server-Seitige Authentifikation
   a) Zertifikate
   b) Public-Key Infrastruktur
3. Protokolle zur sicheren Kommunikation
   a) Transport Layer Security (TLS)
   b) Virtual Private Networks (VPN)
4. Zusammenfassung

Lernziele Protokolle und Infrastruktur

- Verständnis vom Einsatz kryptographischer Verfahren im praktischen Kontext
- Verständnis der IT-Sicherheitsprobleme bei der Absicherung praktischer Anwendungsfälle
- Fähigkeit zur Beurteilung der Sicherheit von Protokollen
- Verständnis des Zusammenspiels der Sicherheitsmaßnahmen, die sich im Anwendungsbeispiel ergeben

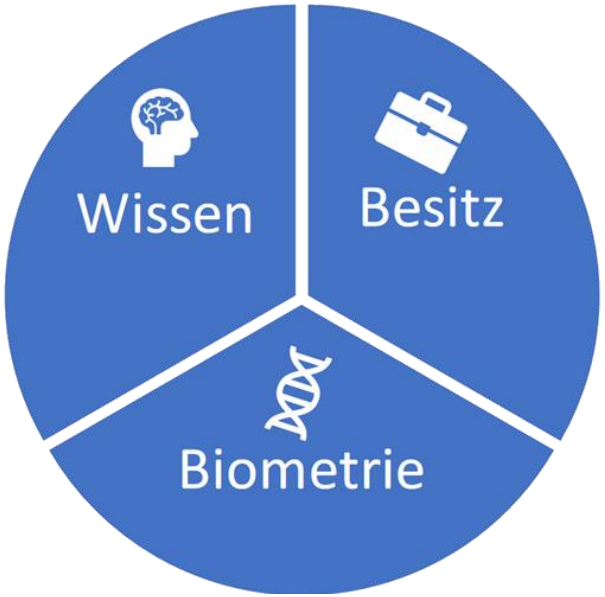
Nutzer*innen-Seitige Authentifikation

- **Bedrohung:** Mallory möchte sich in Alice‘s eMail Account einloggen
- **Ziel:** Nur Alice darf Zugriff auf ihren eMail Account haben



Verfahren für Authentifikation

Merkmal	Sicherheit basiert auf	Beispiele
Wissen	Nur Nutzer*innen bekannt	• **Passwort** • PIN • Sicherheitsfragen
Besitz	Im Besitz von Nutzer*innen. Entwendung wird bemerkt.	• **Chipkarte (z.B. SIM-Karte)** • **Smartphone** • USB Token • TAN Generatoren
Biometrie	Physiologisch oder verhaltenstypisch einzigartige und unkopierbare Merkmale einer Person	• **Fingerabdruck** • Gesicht, Iris, Retina • Gang, Tastaturanschläge

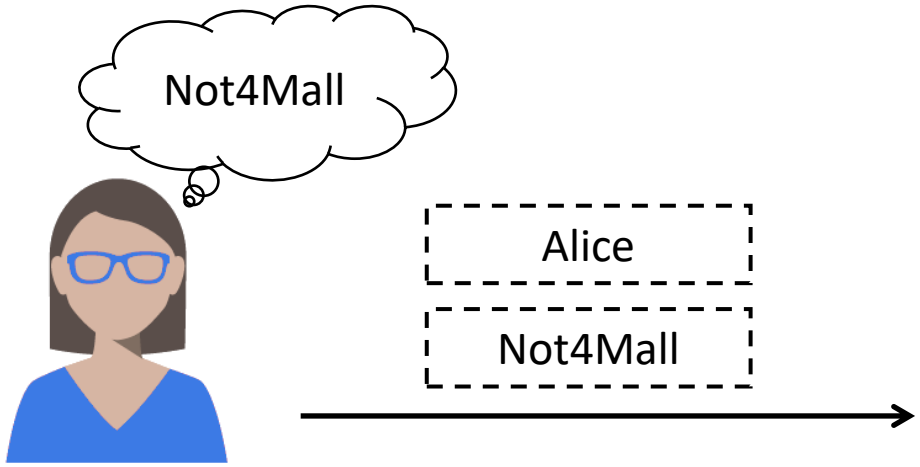


Quelle: Uni Potsdam, Authentifkation WS 17_18

Authentifikation via Wissen
Passwörter

- Passwörter sind das häufigste Authentifizierungsverfahren im Internet

- **Nur Alice kennt** einen geheimen String, der **nicht erraten** werden kann. Probleme:
  1. **Nicht erraten:** Wie sieht ein gutes, nicht-erratbares Passwort aus?
  2. **Nur Alice kennt:** Das Passwort wird eingegeben, übertragen und auf dem Server gespeichert!



Nutzer*in	Passwort
Alice	Not4Mall
Bob	Baumeister
Eve	*Lausch*
.....	....

Erratbarkeit von Passwörtern

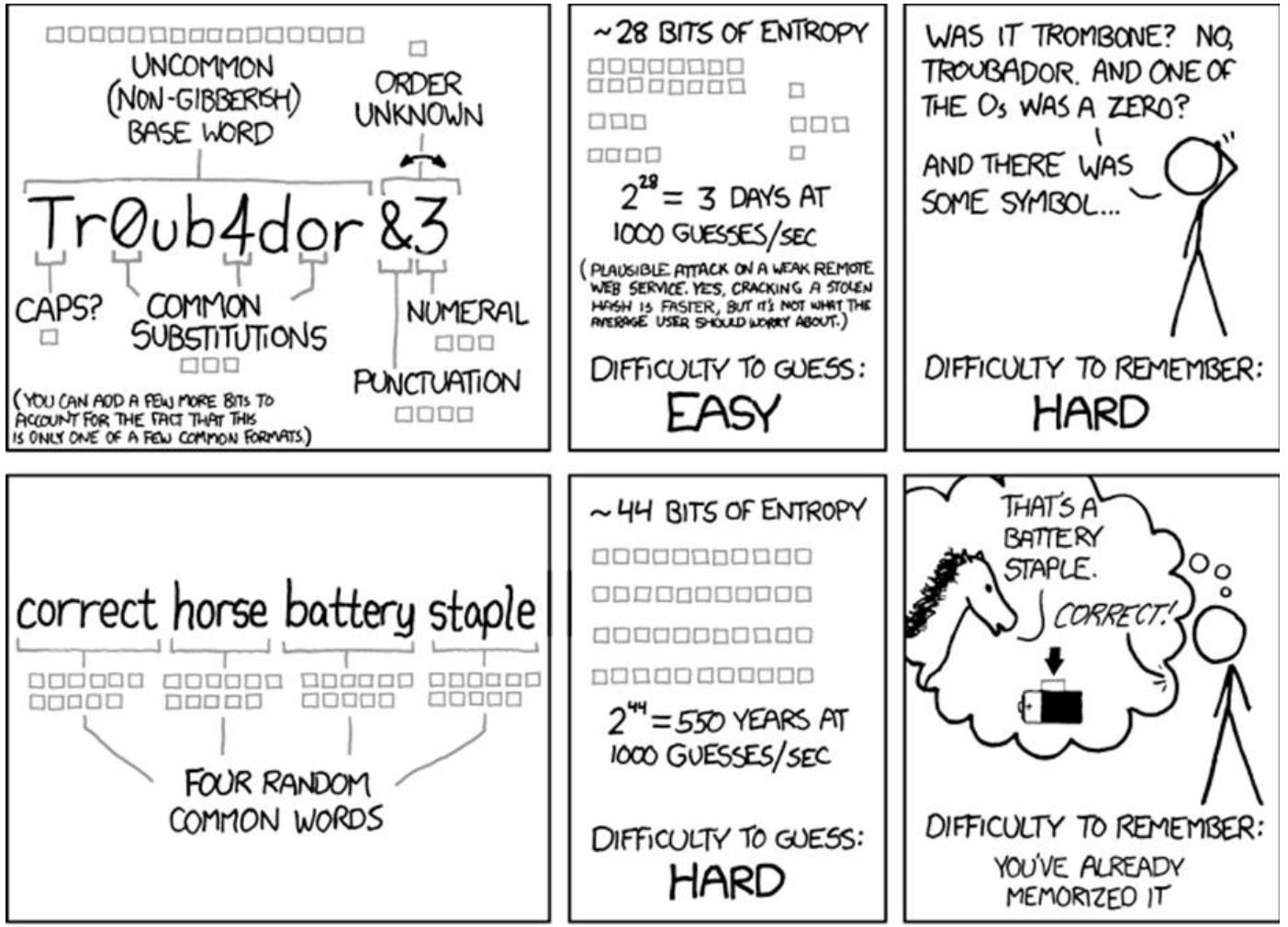
1. passwort
   - Offensichtliches Wort
   ✗ ➔ Häufiges Passwort
2. Marvin81
   - Personalisiertes Wort mit Geburtsdatum
   ✗ ➔ Vorhersagbar falls Person bekannt
3. Tr0ub4dor&3
   ???

- Erratbarkeit von Passwörtern hängt von Anzahl der Kombinationen ab, die ein*e Angreifer*in ausprobieren müsste

- **Entropie:** $\log_2(Anzahl\ Kombinationen)$ (➔ Details siehe Übung)

Erratbarkeit von Passwörtern

- Menschen wählen Passwörter nicht sicher sondern leicht merkbar
- Passwort Policies um Menschen zu sicheren Passwörtern zu „erziehen“
- Jahrelanges Tauziehen zwischen Nutzer*innen und Passwort Policies führte zu vorhersagbaren Passwörtern



Quelle: https://xkcd.com/936/

Erratbarkeit von Passwörtern

Beispiel Passwort	Angriffsstrategie	Passwort Policy Update
asdf	Zufällige Buchstabenkombinationen testen	Mindestens 8 Zeichen
passwort	Wörter aus Wörterbuch testen	Großbuchstaben müssen enthalten sein
PassWort	Buchstabenkombinationen klein und groß	Ziffern hinzufügen
PassW0rt21	• Jahreszahlen anhängen • Gängige Substitutionen (o → 0)	Sonderzeichen hinzufügen
P$ssW0rt21	Gängige Substitutionen (4 → $)	Passwortupdates nach 90 Tagen
P$ssW0rt22	Counter am Ende hochzählen	-

- Heutige Passwort Policy Empfehlungen [NIST17]:
  - Keine Vorgabe von zu enthaltenden Zeichen und kein Passwortupdatezwang
  - Prüfen des Passworts gegen gebrochene Passwörter (https://haveibeenpwned.com)
  - Prüfung der Komplexität des Passworts und Feedback an Nutzer*in
- Passwortkomplexität ermitteln ➔ Siehe Übung!

Subjektive Tipps an Nutzer*innen

- Mindestens 8-16 zufällige Zeichen (je nach Wichtigkeit des Dienstes)
- Verwenden Sie einzigartige und sichere Passwörter für kritische Dienste
  - Z.B., eMail, Online Banking, Unternehmens-IT
  - Nutzen Sie Passwortmanager um sichere Passwörter zu generieren und zu speichern
- Prüfen Sie ob Ihre Passwörter geleakt wurden (haveIbeenpwned.com)
  - Falls ja: Wechseln Sie das Passwort
- Geben Sie Ihren Usernamen und Passwort nie an andere Personen weiter

Prüfen der Passwortrateversuche

- **Problem:** Passwörter können zwar geraten werden, aber nicht geprüft

- Möglichkeiten um ein geratenes Passwort zu verifizieren:
  - **Probehafter Login** auf Webseite
  - Abgreifen **geheimer Daten** auf dem **Server**

- Gegenmaßnahmen **Probehafter Login**, nach X fehlerhaften Versuchen:
  1. Wartezeit zwischen Anmeldeversuchen (exponentiell wachsend)
  2. Blockieren einer IP
  3. Sperrung eines Accounts

Speichern von Passwörtern

- **Alternative zum Passwort raten:** Auslesen der Passwörter auf dem Server

Nutzer*in	Passwort
Alice	Not4Mall
Bob	Baumeister
Eve	*Lausch*

- Mechanismus benötigt um Passwörter sicher auf dem Server zu speichern und trotzdem einen Abgleich zu ermöglichen

https://www.heise.de/security/meldung/Neue-Passwort-Leaks-Insgesamt-2-2-Milliarden-Accounts-betroffen-4287538.html

Sicheres Speichern von Passwörtern
Einfaches Hashing

- **Variante 1:** Speichern des Passwort Hashwertes und Vergleich der Hashes
  - **Einweg:** Vom Hashwert kann nicht auf das Passwort zurückgerechnet werden
  - **Kollisionsresistenz:** Ein falsches Passwort führt zu einem anderen Hashwert

Alice

Not4Mall

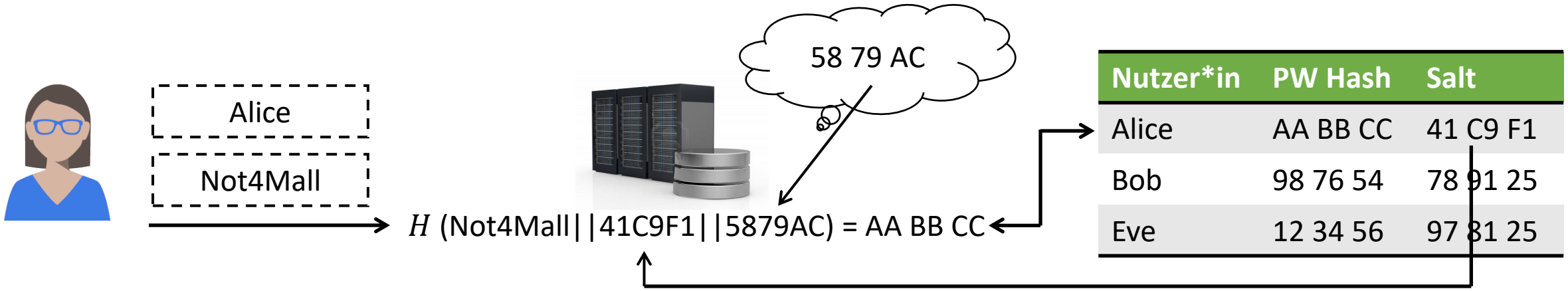
H (Not4Mall) = C0 FF EE

Nutzer*in	Passworthash
Alice	C0 FF EE
Bob	AA 55 99
Eve	11 22 33

- **Problem:** Hashfunktion kann für alle Passwörter einmal vorberechnet werden
  - Sogenannte **Rainbow Table** enthält vorberechnete Hashes
  - Bei neuem Passwort Leak nur noch Abgleich der Hashes mit Rainbow Table

Sicheres Speichern von Passwörtern
Salt und Pepper Hashing

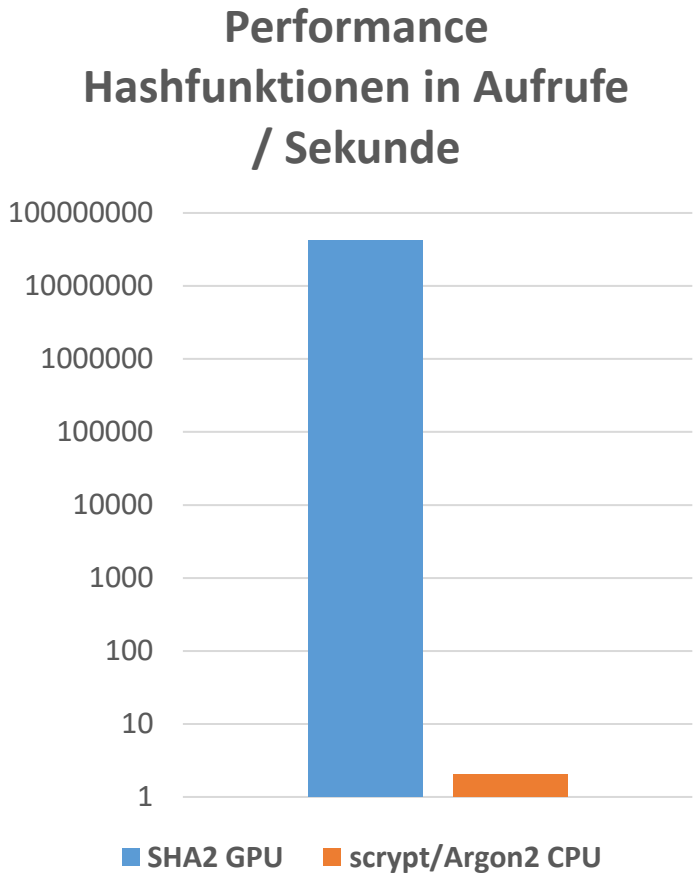
- **Variante 2:** Passwörter werden zusätzlichen und zufälligen Daten gehashed
  - **Salt:** Individueller Zufallswert der mit Passwort gespeichert wird
  - **Pepper:** Zufälliger und geheimer Wert der für alle Passwörter konstant ist
  - Berechnung als H(Passwort || Salt || Pepper)



- **Sicherheitsgewinn:**
  - **Salt:** Rainbow Tables werden erschwert
  - **Pepper:** Brute-force erschwert, da Pepper geraten werden muss (solange Pepper unbekannt)

Sicheres Speichern von Passwörtern
Passworthashfunktionen

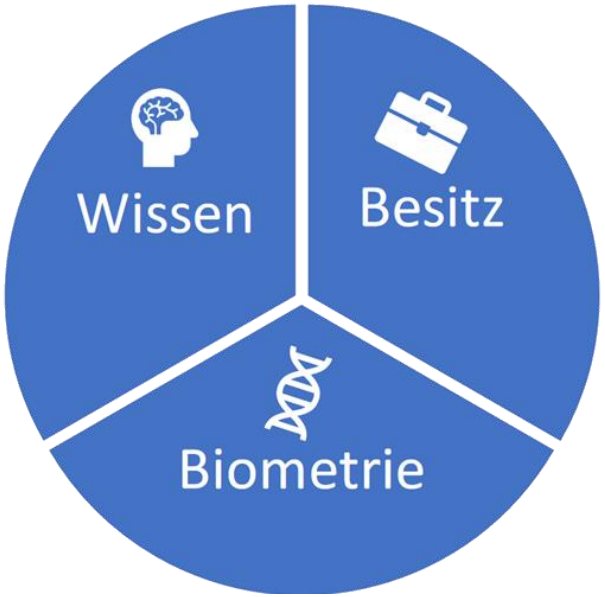
- Designkriterium von Hashfunktionen wie SHA2:
  - Schnelle Ausführung in Software
  - Gute Parallelisierbarkeit in Hardware
- Für Passworthashing aber schlecht, da Brute-Force Angriffe mächtiger werden
- Lösung: Passworthashverfahren mit einstellbarer Effizienz und schlechtem Hardwaredesign:
  - scrypt
  - **Argon2**



Quelle: https://pthree.org/2016/06/28/lets-talk-password-hashing/

Verfahren für Authentifikation

Merkmal	Sicherheit basiert auf	Beispiele
Wissen	Nur Nutzer*innen bekannt	• **Passwort** • PIN • Sicherheitsfragen
Besitz	Im Besitz von Nutzer*innen. Entwendung wird bemerkt.	• **Chipkarte (z.B. SIM-Karte)** • **Smartphone** • USB Token • TAN Generatoren
Biometrie	Physiologisch oder verhaltenstypisch einzigartige und unkopierbare Merkmale einer Person	• **Fingerabdruck** • Gesicht, Iris, Retina • Gang, Tastaturanschläge



Quelle: Uni Potsdam, Authentifkation WS 17_18

Authentifikation via Besitz

- Wissen (z.B. Passwörter) kann kopiert werden ohne, dass Nutzer*in es bemerkt

- **Lösung:** Physisches Objekt, das Nutzer*in ständig bei sich tragen kann und dessen Diebstahl bemerkt wird (analog zum Haustürschlüssel)

- Verschiedene Möglichkeiten zur Authentifikation via Besitz:
  - **Besitz als Medium**, das Geheimnis an Nutzer*in kommuniziert (z.B. Einmalpasswort)
  - **Besitz als Authentifikator**, welches Login für Nutzer*in übernimmt (z.B. Smartcard)

Authentifikation via Besitz Einmalpasswort (OTP)

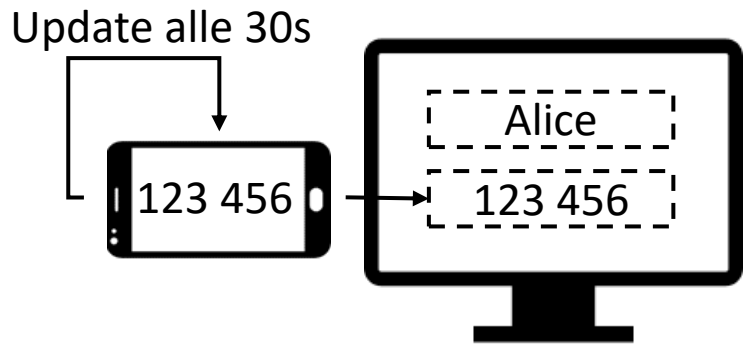
- **Einmalpasswort (OTP):** kurzzeitiges gültiges und einmal nutzbares Passwort:
  - SMS mit TAN, die für bestimmte Zeit gültig ist
  - **Gerät, welches einen Sicherheitsschlüssel einprogrammiert hat**

1. Initialisierung (bei Registrierung)

Funktionsweise:

1. Backend generiert Schlüssel K für Nutzer*in
2. Gerät liest und speichert Schlüssel K

2. Login



Funktionsweise:

1. Gerät berechnet alle 30s Hashfunktion $H(K\ , Zeit)$
2. Backend verifiziert Hashwert und prüft Zeittoleranz

Authentifikation via Besitz
Smart Card

- **Smart Card:** Karte mit eingebautem Chip, der Hardware Logik und Speicher enthält. Beispiele:
  - Bankkarte oder Kreditkarte
  - Personalausweis

- Smart Card enthält einen kryptographischen Schlüssel, der speziell gegen auslesen gesichert ist

- Smart Card authentifiziert sich via einem „**Challenge-Response Protokoll**“ direkt gegenüber einem Server (Smart Card oder NFC Reader benötigt)

Authentifikation via Besitz
Challenge-Response Protokoll

1. Server sendet zufällig generierte Challenge C
2. Alice‘s Smart Card berechnet Response R basierend mittels einer kryptographischen Funktion f unter Eingabe der Challenge und eines Geheimnisses G
3. Alice‘s Smard Card sendet Response R zurück an Server
4. Server verifiziert Response R

Challenge C ←

Response
$R = f(C, G)$

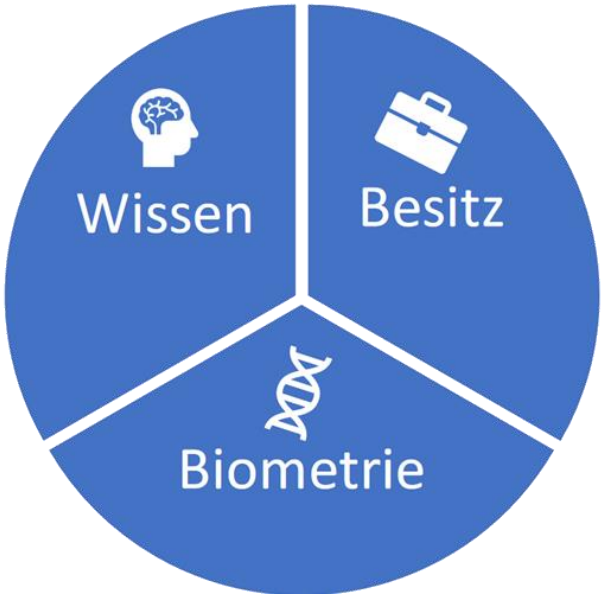
Response R →

Prüfe Response

Konkrete Umsetzungen für Challenge / Response → Siehe Übung

Verfahren für Authentifikation

Merkmal	Sicherheit basiert auf	Beispiele
Wissen	Nur Nutzer*innen bekannt	• **Passwort** • PIN • Sicherheitsfragen
Besitz	Im Besitz von Nutzer*innen. Entwendung wird bemerkt.	• **Chipkarte (z.B. SIM-Karte)** • **Smartphone** • USB Token • TAN Generatoren
Biometrie	Physiologisch oder verhaltenstypisch einzigartige und unkopierbare Merkmale einer Person	• **Fingerabdruck** • Gesicht, Iris, Retina • Gang, Tastaturanschläge



Quelle: Uni Potsdam, Authentifkation WS 17_18

Biometrische Identifikation

- Physiologische oder verhaltenstechnische eindeutige Merkmale
  - Fingerabdrücke
  - Gesicht, Iris, Retina
  - Tippverhalten auf der Tastatur
  - Gang

Quelle: [JNR16]

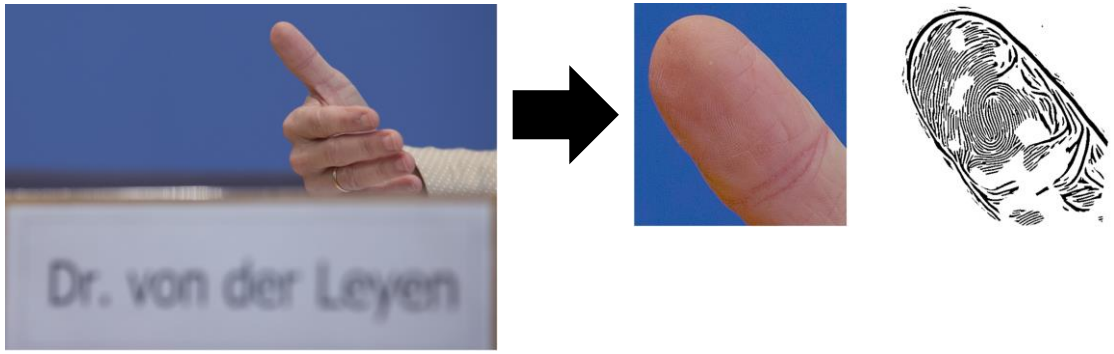
- Scanner legt Modell des Merkmals an (z.B. via maschinellem Lernen)
  - Unpräzise Messungen müssen trotzdem zur richtigen Entscheidung führen

- Merkmale sind eindeutig und schwer von Menschen zu kopieren

Biometrische Identifikation
Sicherheit und Nutzbarkeit

- Biometrische Merkmale sind zwar schwer von Menschen zu kopieren, aber:
  - Öffentlich einsehbar und nicht sonderlich geschützt
  - Maschinell nachstellbar wenn das Modell bekannt ist
  - Unsicher wenn Daten einmal veröffentlicht wurden

- Nutzbarkeit ist sehr gut, da:
  - Biometrische Merkmale immer dabei
  - Weite mediale Verbreitung



https://fahrplan.events.ccc.de/congress/2014/Fahrplan/system/attachments/2573/original/congress2014.pdf

- Erkennung häufig fehlerhaft, da:
  - Externe Einflüsse störend wirken (Licht, Kälte)
  - Merkmale sich über die Zeit ändern (Gesicht, Deutlichkeit Fingerabdruck)

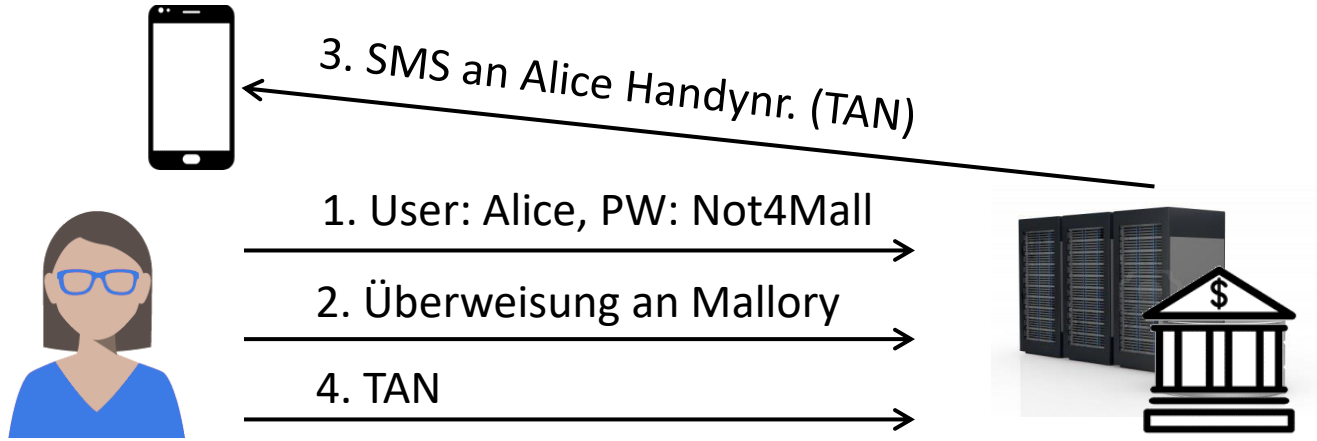
Vor- und Nachteile der Authentifizierungstechniken

Merkmal	Vorteile	Nachteile
Wissen	• Einfach zu implementieren • Theoretisch sicher • Keine zusätzliche Technik	• Sicherheit abhängig vom gewählten Passwort • Schwer zu merken bei vielen Zugängen • Kopieren nicht bemerkbar
Besitz	• Kein Merken notwendig • Entwendung ist bemerkbar • Standardmäßig hohe Sicherheit	• Ggf. Physikalische Schnittstelle benötigt (Smart Card Reader) • Aufwändig in der Umsetzung • Muss von Nutzer*in mittransportiert werden
Biometrie	• Kein Transport oder Merken notwendig • Eindeutig pro Mensch	• Physikalischer Scanner benötigt • Externe Faktoren können zu Fehlern führen • Kopieren manchmal nicht bemerkbar • Merkmal nicht wechselbar ➔ Ein Leak reicht um Sicherheit des Merkmals zu korrumpieren • Kann auch gegen Nutzer*in verwendet werden

Zwei- und Mehrfaktorauthentifizierung

- Mechanismen aus Wissen, Besitz und Biometrie können kombiniert werden um mehr Sicherheit zu erreichen:
  - **Zweifaktor Authentifizierung:** Kombination von zwei Mechanismen aus verschiedenen Kategorien
  - **Multifaktor Authentifizierung:** Kombination von mehr als zwei Mechanismen aus verschiedenen Kategorien

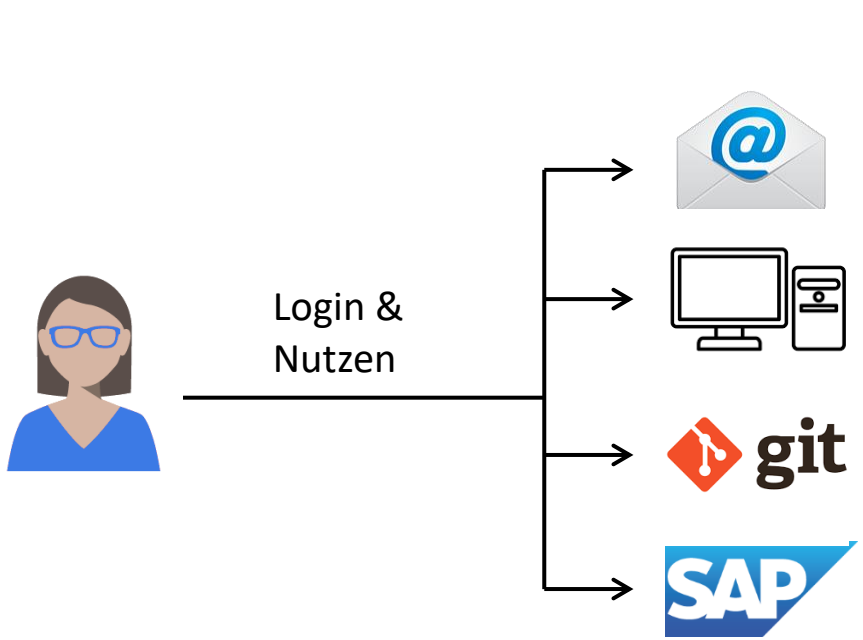
- **Beispiel: Online Überweisung**
  - Wissen (Passwort)
  - Besitz (SIM Karte / Smartphone)

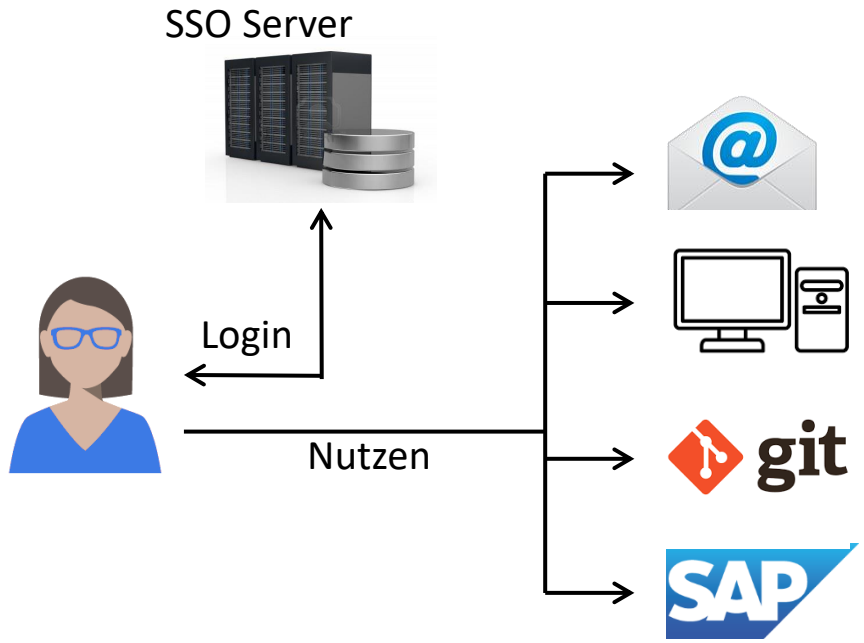


Single Sign-On (SSO) Systeme

- Viele Systeme verlangen eine individuelle Authentifizierung

- **Lösung: Single Sign-On (SSO)** Systeme, die zentrale Authentifizierung ermöglichen
  - Unternehmensnetzwerke: Kerberos
  - Privat Online: OAuth/OpenID Systeme (Google, Facebook, …)

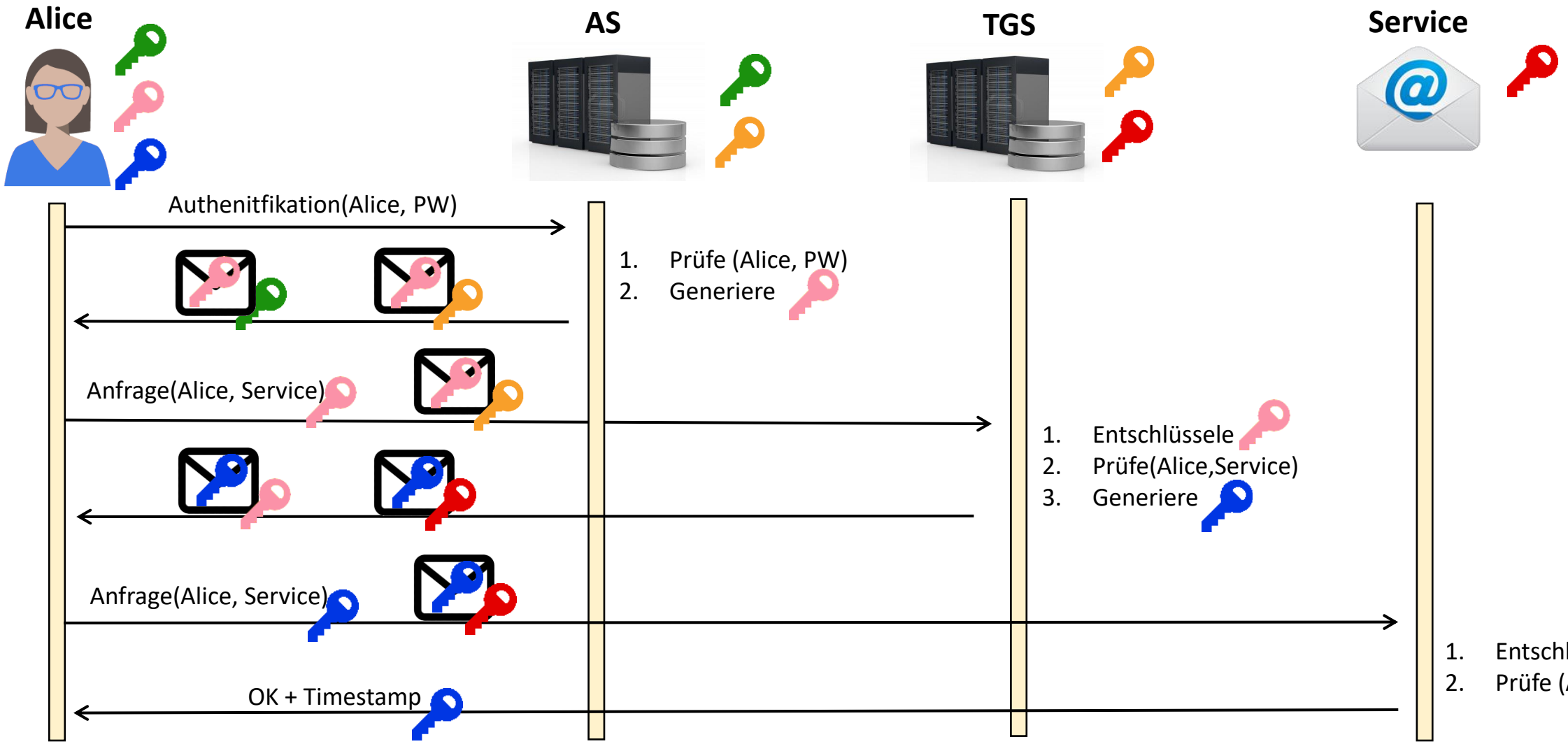




Kerberos Single Sign-On

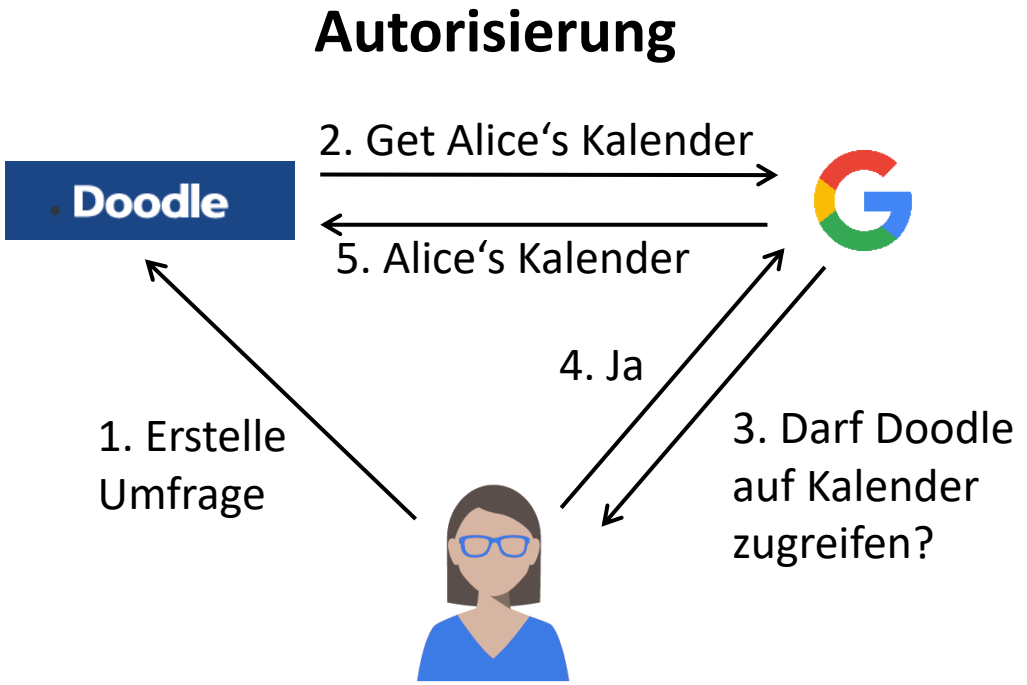
- Kerberos ist der de-facto Standard für Single Sign-On im Unternehmen
- Nutzer*in authentifiziert sich nur gegenüber Domain Controller und bekommt von diesem Tickets zur Nutzung von Services ausgestellt
- Domain Controller besteht aus:
  - **Authentifizierungsserver (AS):** Prüft Identität und stellt ein Ticket für eine Sitzung über eine gewisse Dauer aus
  - **Ticketgenehmigungsserver (TGS):** Prüft ob Sitzung aktiv und stellt Ticket zur Nutzung eines Services aus.

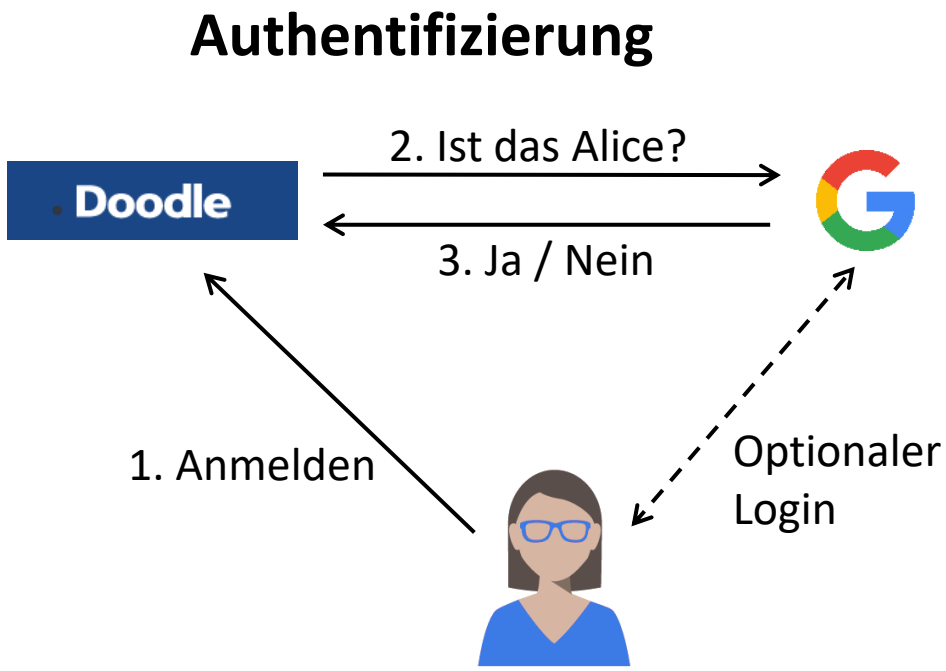
Kerberos Protokoll

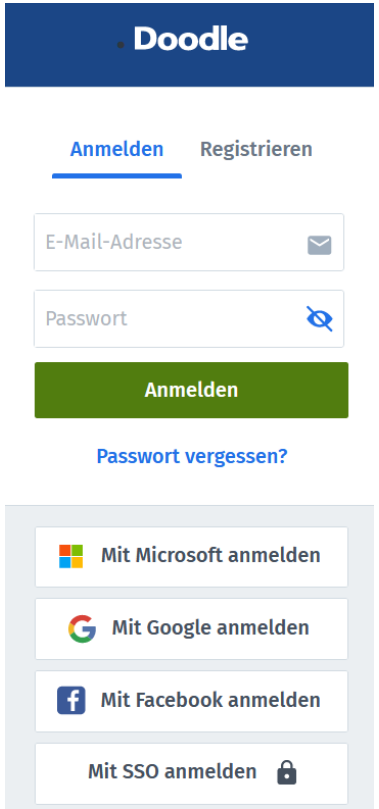


Open-Authorization [OAuth]

- OAuth 2.0 ermöglicht **Autorisierung für Webseiten,** die auf Accounts anderer Webseiten zugreifen wollen, wird aber oft zur **Authentifizierung** als Single Sign-On verwendet







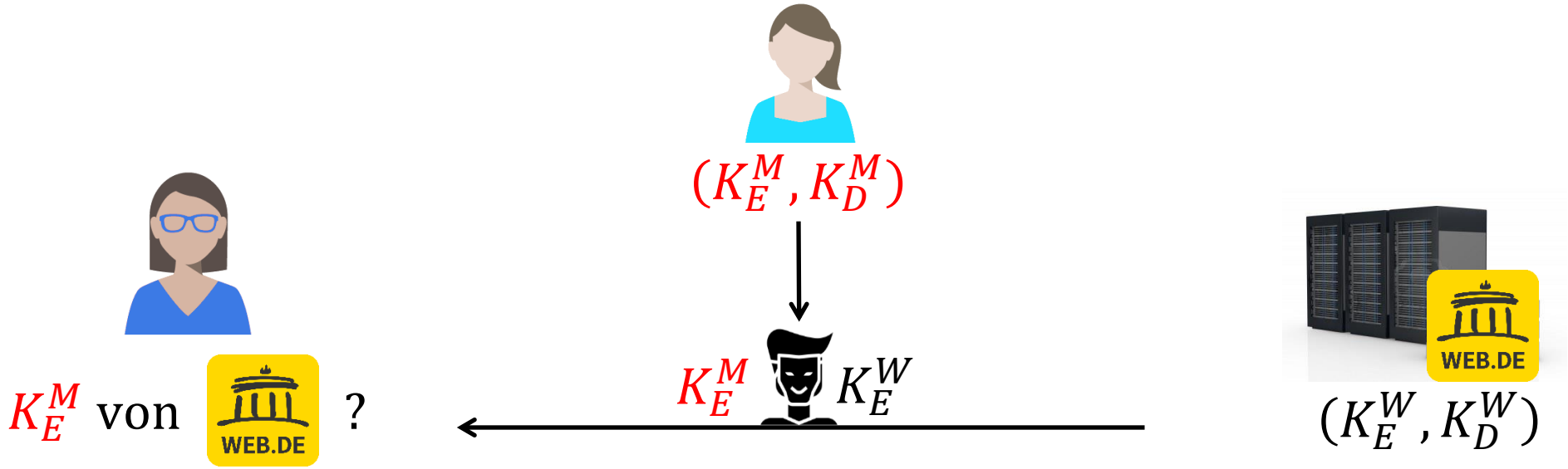
Inhalt des Kapitels

1. Nutzer*innen-Seitige Authentifikation
   a) Authentifikationsverfahren (Passwörter, Besitz, Biometrie, Zwei/Mehrfaktor)
   b) Challenge / Response
   c) Single Sign-On (Kerberos) und Identity Provider (OAuth)
2. **Server-Seitige Authentifikation**
   a) Zertifikate
   b) Public-Key Infrastruktur
3. Protokolle zur sicheren Kommunikation
   a) Transport Layer Security (TLS)
   b) Virtual Private Networks (VPN)
4. Zusammenfassung

Server-Seitige Authentifkation

- **Bedrohung:** Mallory tauscht öffentlichen Schlüssel aus um Kommunikation abzufangen

- **Ziel:** Alice kann prüfen ob der öffentliche Schlüssel zum korrekten Ziel gehört

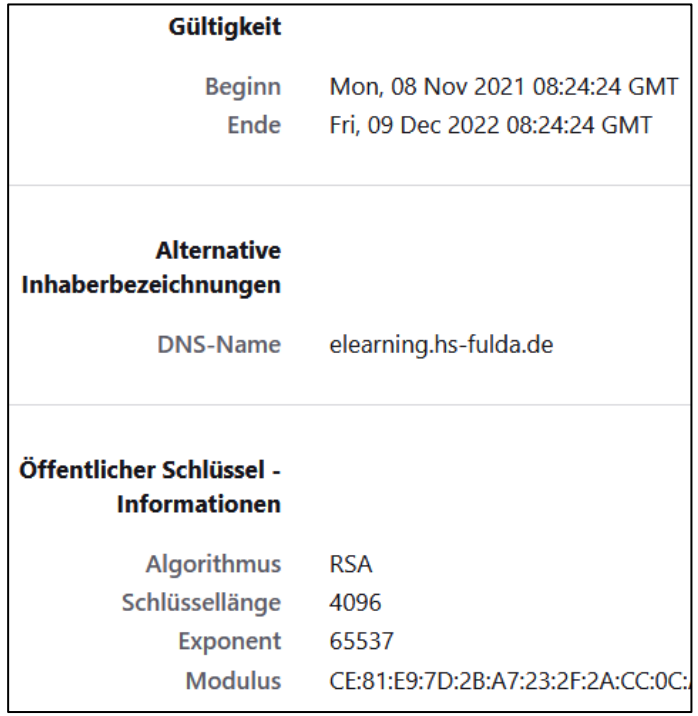


Zertifikate (1/3)

- **Idee:** Verknüpfen des öffentlichen Schlüssels des Webservers mit Attributen, die nachgeprüft werden können (z.B. DNS-Name oder IP).

- Zertifikate können weitere Informationen enthalten:
  - Gültigkeitsdauer (von / bis)
  - Verwendungszweck des Schlüssels (Signieren/Verschlüsseln)
  - Anwendungsspezifische Informationen (z.B. Impfstoff für Impfzertifikate)
  - …..

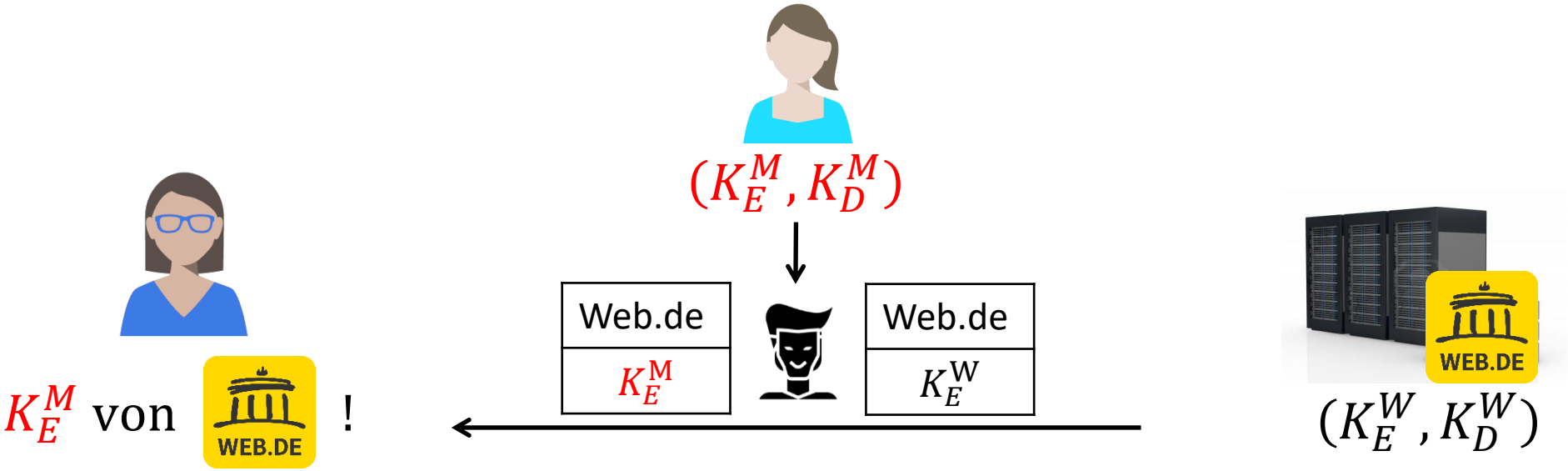
- Zertifikat wird an Nutzer*in übertragen



Gültigkeit	
Beginn	Mon, 08 Nov 2021 08:24:24 GMT
Ende	Fri, 09 Dec 2022 08:24:24 GMT
Alternative Inhaberbezeichnungen	
DNS-Name	elearning.hs-fulda.de
Öffentlicher Schlüssel - Informationen	
Algorithmus	RSA
Schlüssellänge	4096
Exponent	65537
Modulus	CE:81:E9:7D:2B:A7:23:2F:2A:CC:0C:

Ausschnitt Zertifikat Moodle HS Fulda

Zertifikate (2/3)

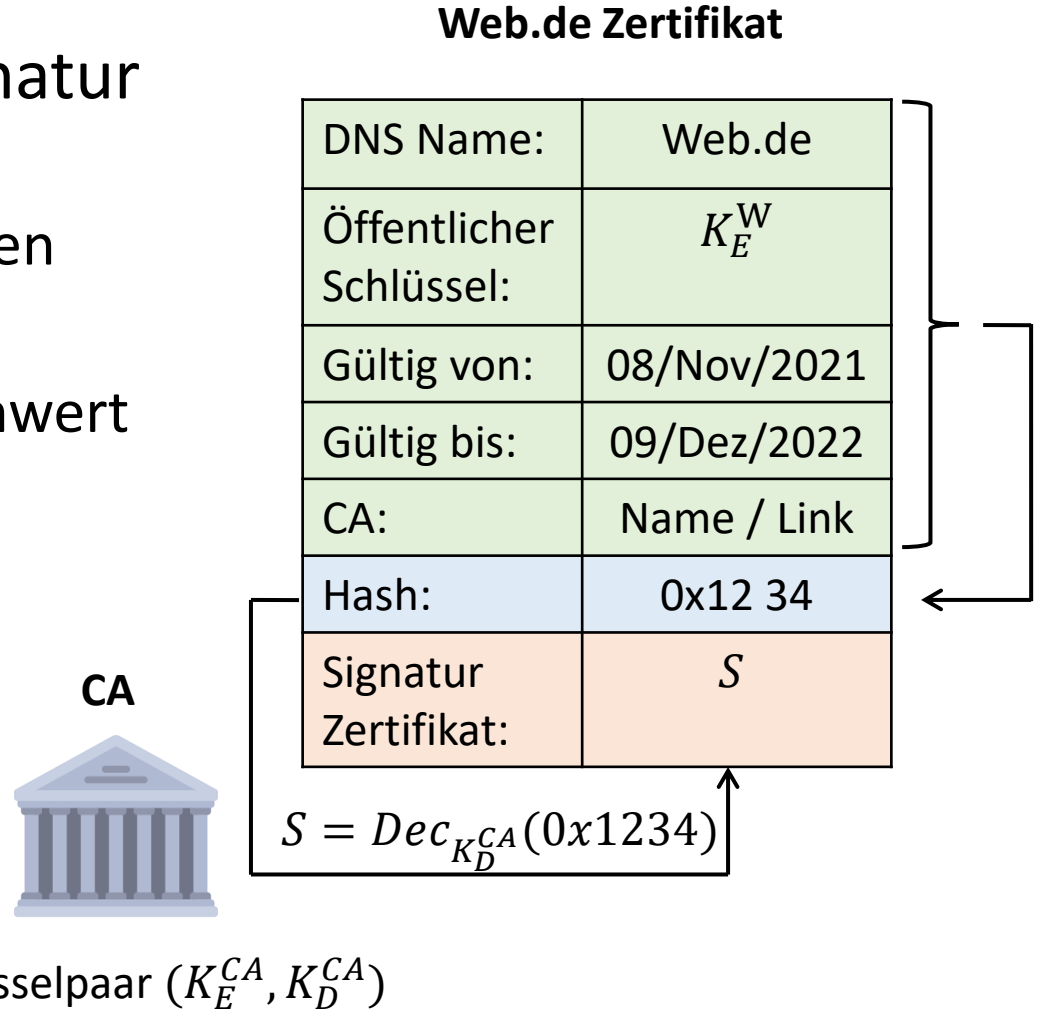


- **Problem:** Authentizität des Zertifikates ist nicht sichergestellt.

- Wenn digitale Signaturen nicht reichen → Mehr digitale Signaturen!

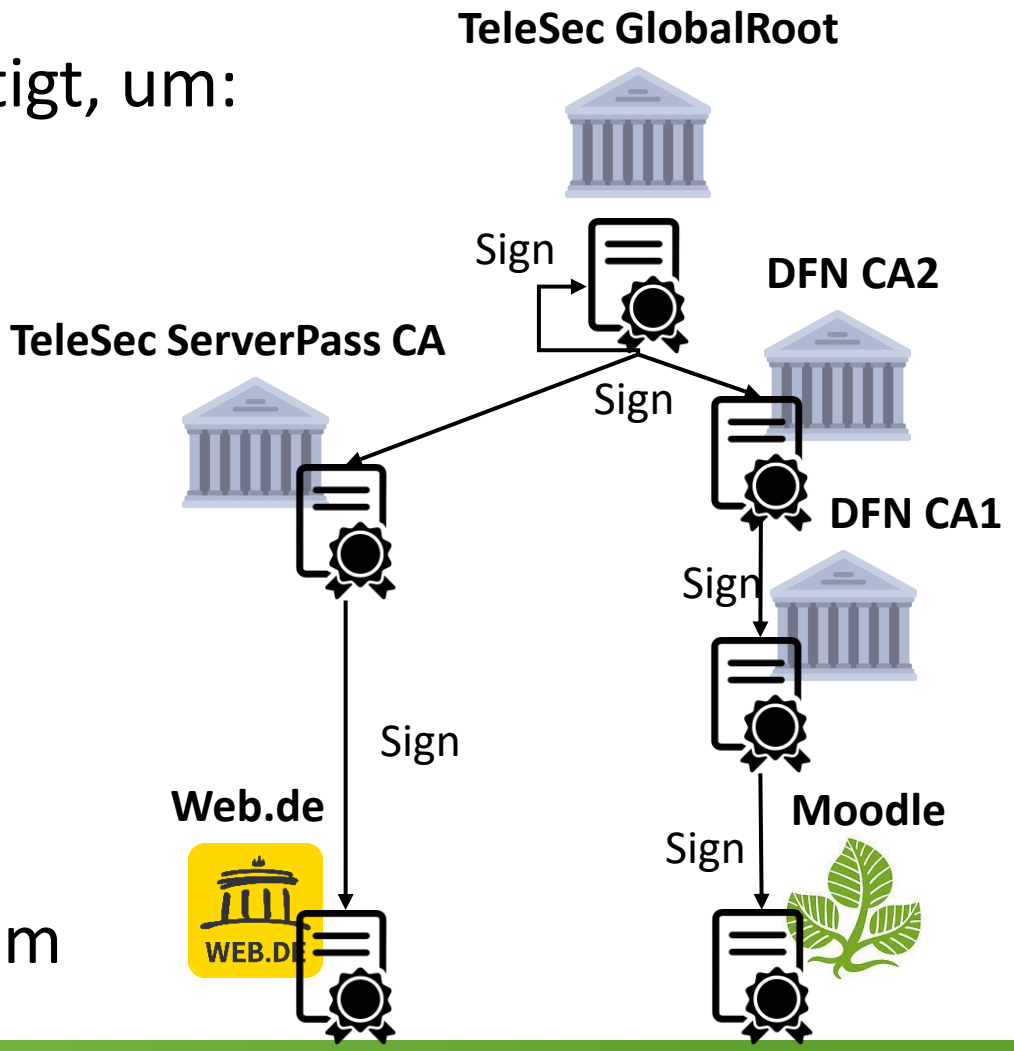
Zertifikate (3/3)

- Zertifizierungsstelle (CA) erstellt digitale Signatur eines Zertifikates:
  1. Identität der CA wird in Zertifikat aufgenommen
  2. Bildung Hashwert über Zertifikatsinhalt
  3. CA mit Schlüsselpaar (K_E^{CA}, K_D^{CA}) signiert Hashwert des Zertifikates
  4. Signatur wird an Zertifikat angehangen
- Nutzer*in kann Zertifikat mit Schlüssel K_E^{CA} der CA prüfen



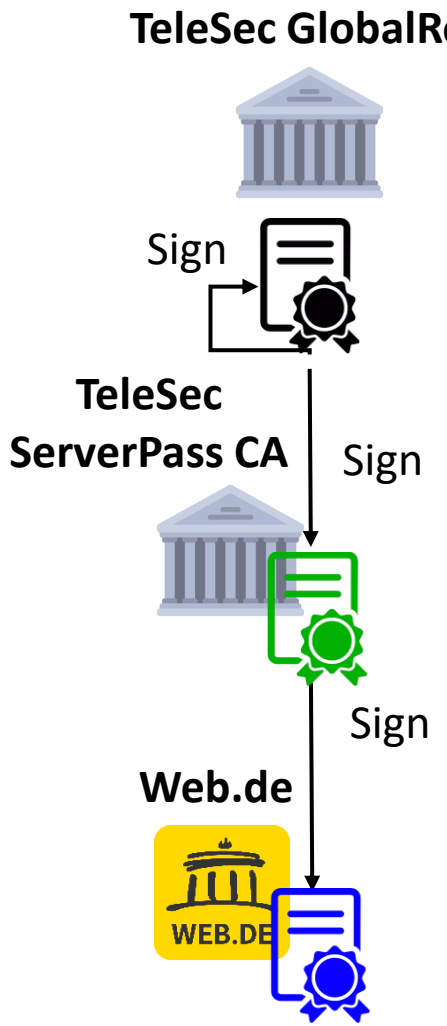
Public Key Infrastruktur (PKI)

- Eine **Public Key Infrastruktur (PKI)** wird benötigt, um:
  - Zertifikate zu erstellen und zu signieren
  - Informationen über Zertifikate bereitzustellen
  - Unsichere Zertifikate zu löschen
- Teilnehmende in einer PKI
  - **Wurzel-CA:** Oberste Zertifizierungsstelle
  - **CA:** Stellt Zertifikate aus
  - **Webseiteninhaber*innen:** Beantragen Zertifikate unter Nachweis der Identität
- Zertifikat der Wurzel-CA liegt im Betriebssystem



Public Key Infrastruktur (PKI)
Zertifikatsprüfung

1. Web.de Zertifikat unbekannt
2. Prüfe Gültigkeit und Identität
3. Prüfe Signatur
   a. CA Zertifikat unbekannt
   b. Hole CA Zertifikat
   c. Prüfe Gültigkeit und Identität
   d. Prüfe Signatur
      i. Wurzel-CA Zertifikat bekannt und verifiziert
      ii. Signatur CA Zertifikat OK
   e. CA Zertifikat Authentisch
   f. Web.de Zertifikat Signatur OK
4. <u>Web.de Zertifikat Authentisch</u>



Zertifikate und PKI Sicherheit

- Probleme mit Zertifikaten:
  - Steuerzeichen im DNS Namen, die nicht korrekt verifiziert werden (web.de\0.foo.com)
  - Implementierungen verifizieren Zertifikate nicht komplett oder gar nicht
  - Neubeantragung von Zertifikaten nach Ablaufdatum vergessen

- Probleme mit PKI:
  - Vertrauensstruktur nicht transparent, da viele Root CAs (400 Root CAs) [PFS14]
  - Eine bösartige CA reicht aus um System zu kompromittieren
  - Fake CAs können beliebig korrekte Zertifikate ausstellen [WoSign]

- Falls Zertifikate „unsicher“ werden (z.B. privater Schlüssel öffentlich geworden), werden Sie auf Zertifikatsrückruflisten (CRLs) der CA veröffentlicht
  - Um Länge der CRLs zu reduzieren ➔ Kurze Lebenszeit ausgestellter Zertifikate

Referenzen

[JNR16]: A. Jain, K. Nandakumar, A. Ross: 50 Years of Biometric Research Accomplishments: https://www.researchgate.net/publication/290509735_50_Years_of_Biometric_Research_Accomplishments_Challenges_and_Opportunities

[ARG]: https://netzpolitik.org/2021/datenleck-argentinische-ausweisdaten-im-netz/

[OAuth]: OAuth Standard for Authorization https://oauth.net/2/

[WoSign]: https://wiki.mozilla.org/CA:WoSign_Issues

[PFS14]: https://www.ifca.ai/pub/fc14/paper_100.pdf