



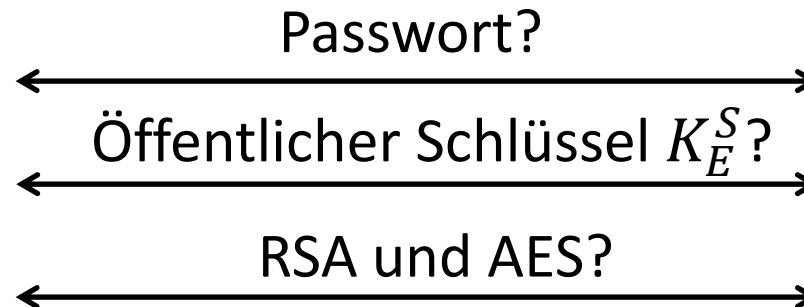
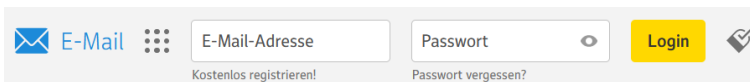
# IT-Sicherheit

## WiSe 2021/22

Protokolle und Infrastruktur (1)



- **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



RSA Schlüsselpaar  $(K_E^S, K_D^S)$



## 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)

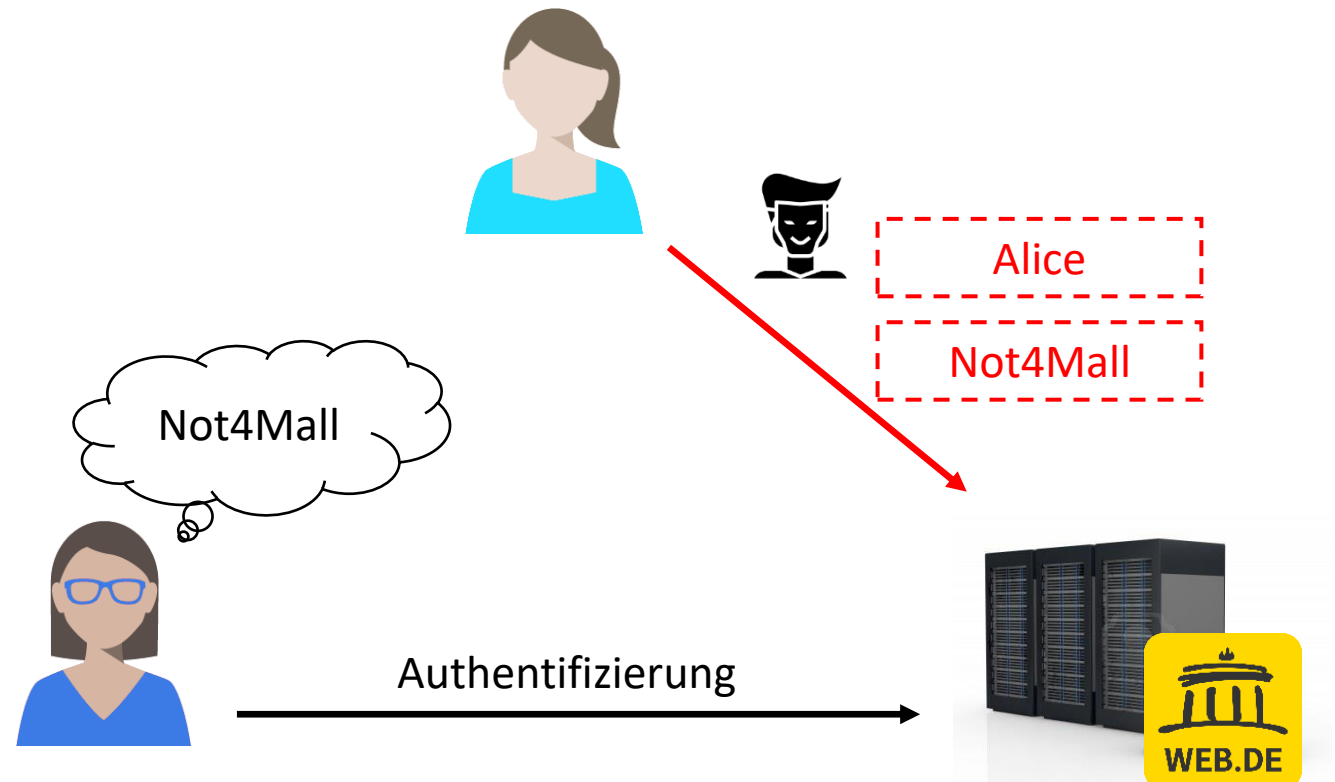
## 4. Zusammenfassung



- 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

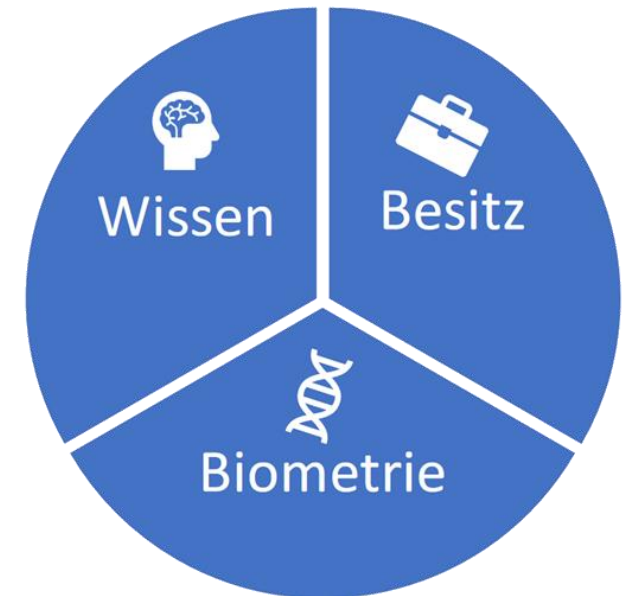


- **Bedrohung:** Mallory möchte sich in Alice's eMail Account einloggen
- **Ziel:** Nur Alice darf Zugriff auf ihren eMail Account haben





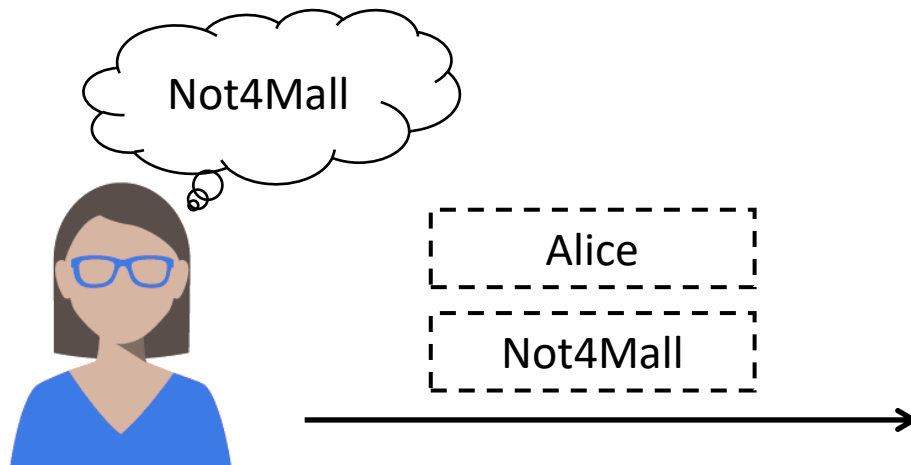
Merkmal	Sicherheit basiert auf	Beispiele
Wissen	Nur Nutzer*innen bekannt	<ul style="list-style-type: none"><li>• <b>Passwort</b></li><li>• PIN</li><li>• Sicherheitsfragen</li></ul>
Besitz	Im Besitz von Nutzer*innen. Entwendung wird bemerkt.	<ul style="list-style-type: none"><li>• <b>Chipkarte (z.B. SIM-Karte)</b></li><li>• <b>Smartphone</b></li><li>• USB Token</li><li>• TAN Generatoren</li></ul>
Biometrie	Physiologisch oder verhaltenstypisch einzigartige und unkopierbare Merkmale einer Person	<ul style="list-style-type: none"><li>• <b>Fingerabdruck</b></li><li>• Gesicht, <u>Iris</u>, Retina</li><li>• Gang, Tastaturanschläge</li></ul>



Quelle: Uni Potsdam, Authentifikation WS 17\_18



- 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*
.....	....



1. **password**
  - Offensichtliches Wort

✗

➔ Häufiges Passwort
  
2. **Marvin81**
  - Personalisiertes Wort mit Geburtsdatum

✗

➔ Vorhersagbar falls Person bekannt
  
3. **Tr0ub4dor&3**

???

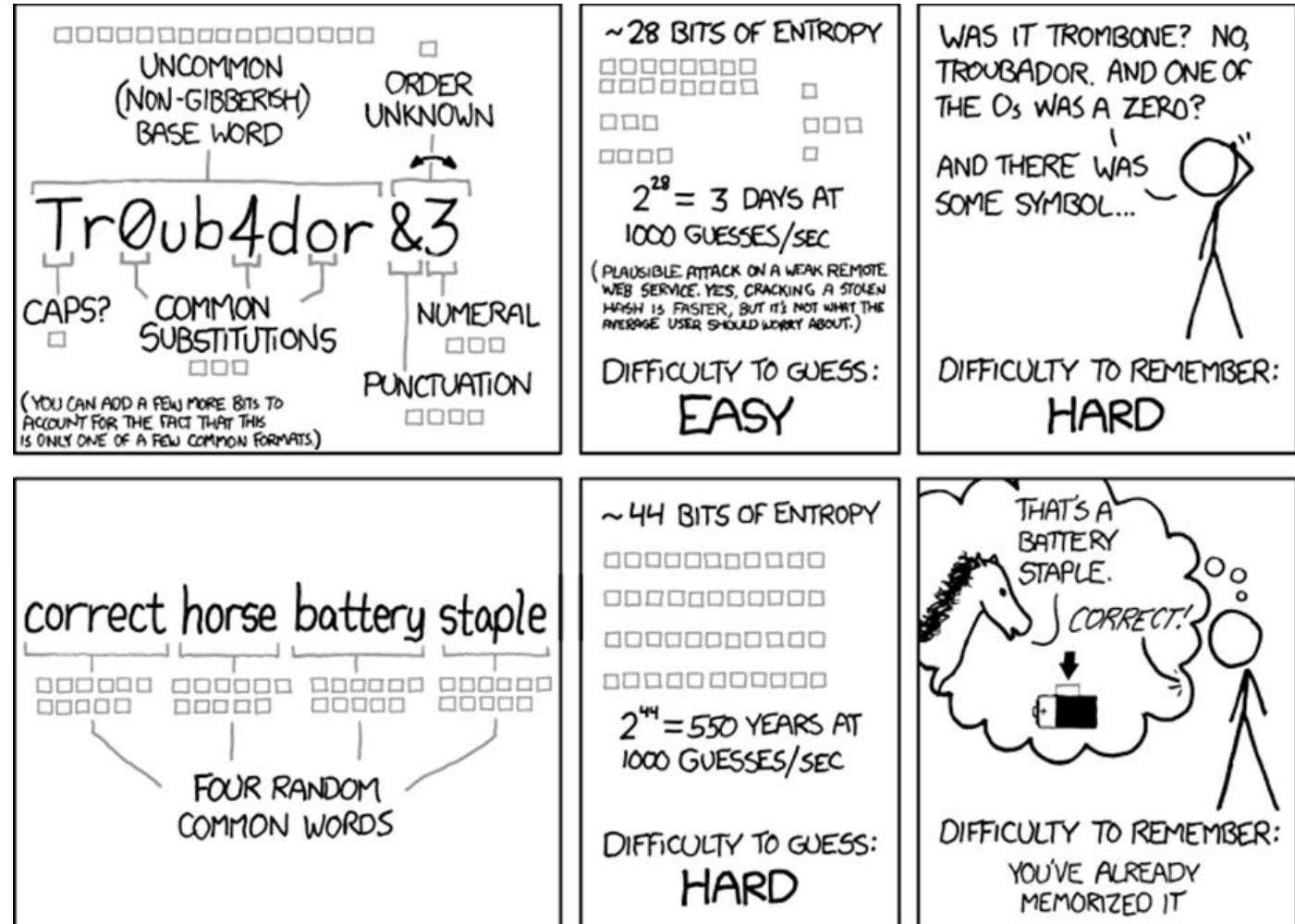
- Erratbarkeit von Passwörtern hängt von Anzahl der Kombinationen ab, die ein\*e Angreifer\*in ausprobieren müsste

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





- 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/>



Beispiel Passwort	Angriffsstrategie	Passwort Policy Update
asdf	Zufällige Buchstabenkombinationen testen	Mindestens 8 Zeichen
password	Wörter aus Wörterbuch testen	Großbuchstaben müssen enthalten sein
PassWort	Buchstabenkombinationen klein und groß	Ziffern hinzufügen
PassW0rt21	<ul style="list-style-type: none"><li>Jahreszahlen anhängen</li><li>Gängige Substitutionen (o → 0)</li></ul>	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!



- 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](https://haveibeenpwned.com))
  - Falls ja: Wechseln Sie das Passwort
- Geben Sie Ihren Usernamen und Passwort nie an andere Personen weiter



- **Problem:** Passwörter können zwar geraten werden, aber nicht geprüft
- Möglichkeiten um ein geratenes Passwort zu verifizieren:
  - **Probhafter Login** auf Webseite
  - Abgreifen **geheimer Daten** auf dem **Server**
- Gegenmaßnahmen **Probhafter Login**, nach X fehlerhaften Versuchen:
  1. Wartezeit zwischen Anmeldeversuchen (exponentiell wachsend)
  2. Blockieren einer IP
  3. Sperrung eines Accounts



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



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

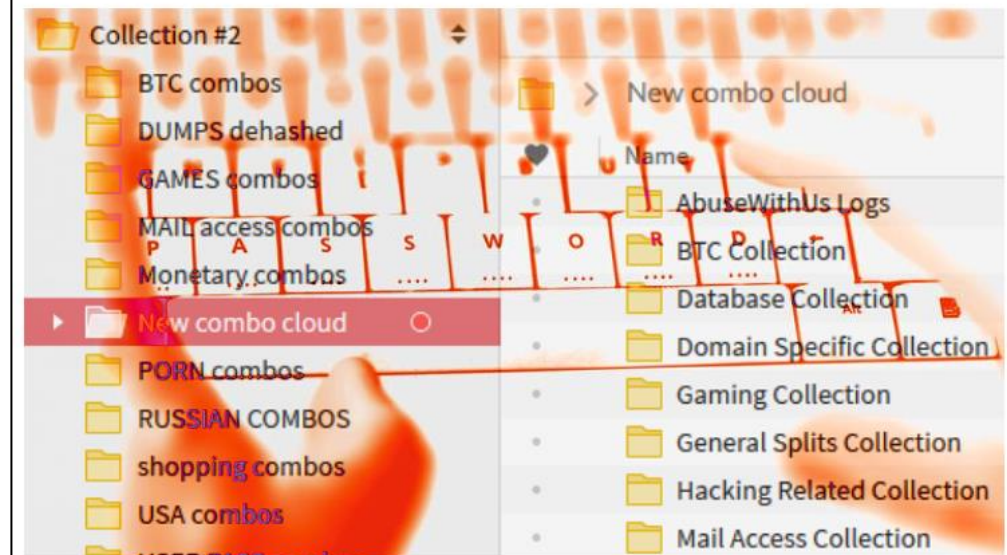
- Mechanismus benötigt um  
Passwörter sicher auf dem Server zu  
speichern und trotzdem einen  
Abgleich zu ermöglichen

## Neue Passwort-Leaks: Insgesamt 2,2 Milliarden Accounts betroffen

Nach der Passwort-Sammlung Collection #1 kursieren nun auch die riesigen Collections #2-5 im Netz. So überprüfen Sie, ob Ihre Accounts betroffen sind.

Lesezeit: 3 Min. In Pocket speichern

577



(Bild: plantic/Shutterstock.com, Montage: heise Online)

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





- **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



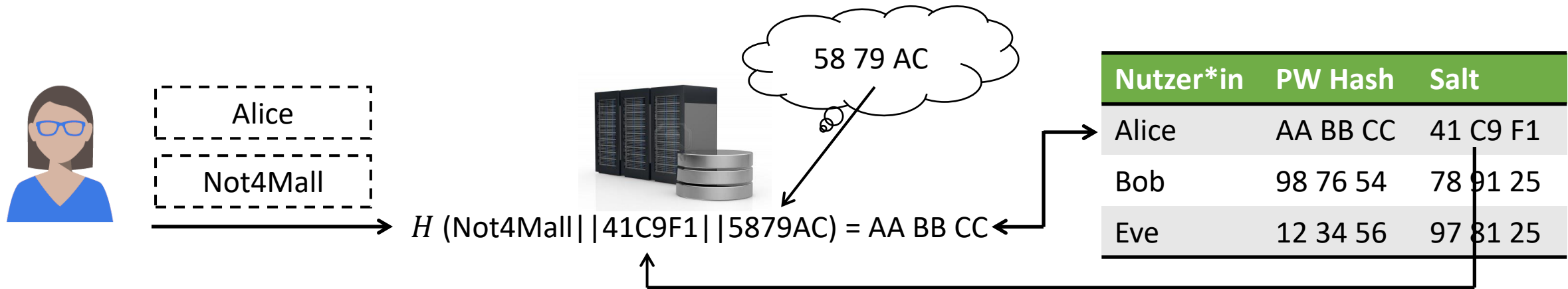
- **Problem:** Hashfunktion kann für alle Passwörter einmal vorberechnet werden
  - Sogenannte **Rainbow Table** enthält vorberechnete Hashes
  - Bei neuem Passwort Leak nur noch Abgleich der Hashes mit Rainbow Table

# Sicheres Speichern von Passwörtern

## Salt und Pepper Hashing



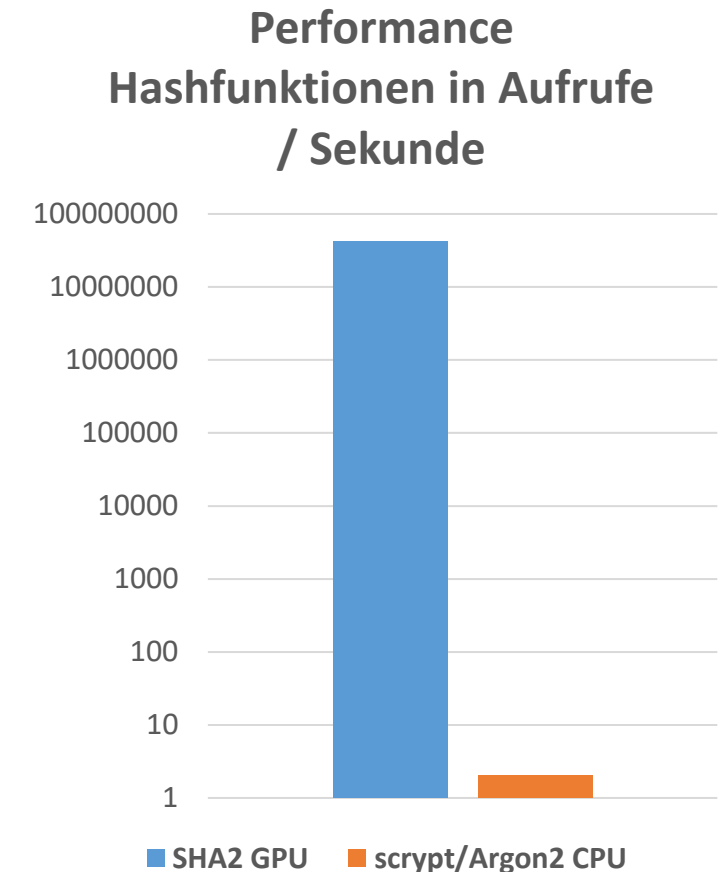
- **Variante 2:** Passwörter werden zusätzlichen und zufälligen Daten gehashed
  - **Salt:** Individueller Zufallswert der mit Passwort gespeichert wird
  - **Pepper:** Zufälliger und geheimer Wert der für alle Passwörter konstant ist
  - Berechnung als  $H(\text{Passwort} || \text{Salt} || \text{Pepper})$



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



- 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**

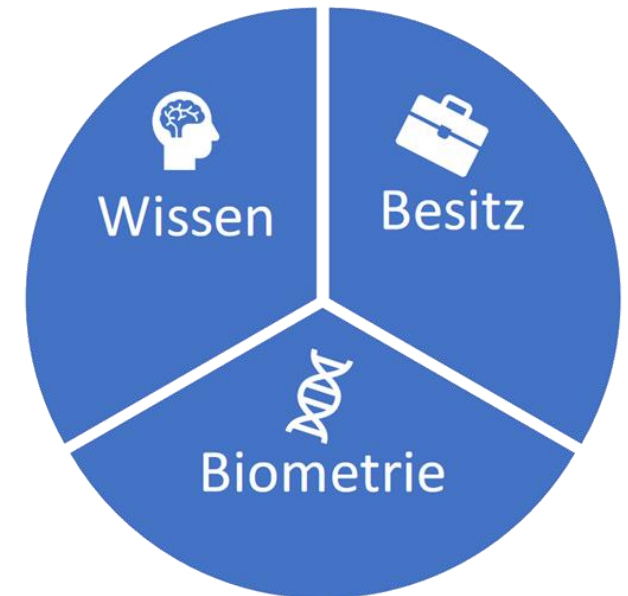


Quelle: <https://pthree.org/2016/06/28/lets-talk-password-hashing/>





Merkmal	Sicherheit basiert auf	Beispiele
Wissen	Nur Nutzer*innen bekannt	<ul style="list-style-type: none"><li>• <b>Passwort</b></li><li>• PIN</li><li>• Sicherheitsfragen</li></ul>
Besitz	Im Besitz von Nutzer*innen. Entwendung wird bemerkt.	<ul style="list-style-type: none"><li>• <b>Chipkarte (z.B. SIM-Karte)</b></li><li>• <b>Smartphone</b></li><li>• USB Token</li><li>• TAN Generatoren</li></ul>
Biometrie	Physiologisch oder verhaltenstypisch einzigartige und unkopierbare Merkmale einer Person	<ul style="list-style-type: none"><li>• <b>Fingerabdruck</b></li><li>• Gesicht, Iris, Retina</li><li>• Gang, Tastaturanschläge</li></ul>



Quelle: Uni Potsdam, Authentifikation WS 17\_18



- 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)



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

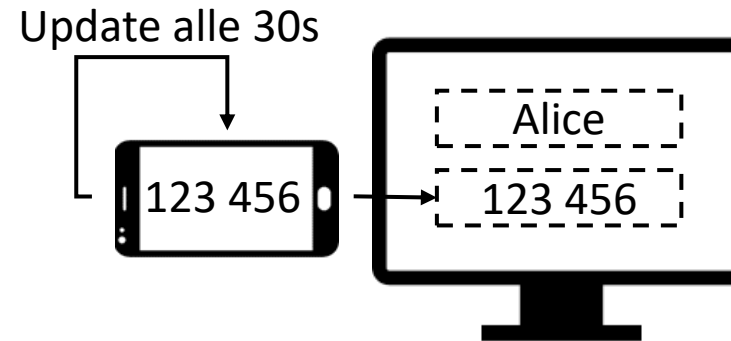
## 1. Initialisierung (bei Registrierung)



### 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



- **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 Smart Card sendet Response  $R$  zurück an Server
4. Server verifiziert Response  $R$



Challenge  $C$



Response  
 $R = f(C, G)$

Response  $R$

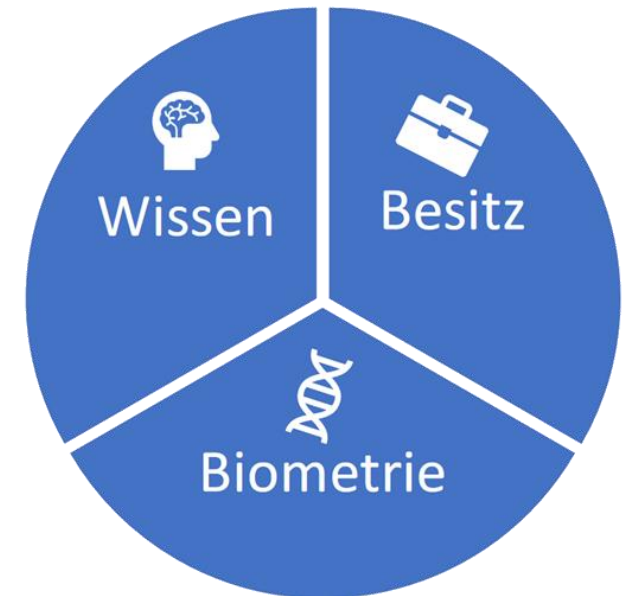


Prüfe Response

Konkrete Umsetzungen für Challenge  
/ Response → Siehe Übung



Merkmal	Sicherheit basiert auf	Beispiele
Wissen	Nur Nutzer*innen bekannt	<ul style="list-style-type: none"><li>• <b>Passwort</b></li><li>• PIN</li><li>• Sicherheitsfragen</li></ul>
Besitz	Im Besitz von Nutzer*innen. Entwendung wird bemerkt.	<ul style="list-style-type: none"><li>• <b>Chipkarte (z.B. SIM-Karte)</b></li><li>• <b>Smartphone</b></li><li>• USB Token</li><li>• TAN Generatoren</li></ul>
Biometrie	Physiologisch oder verhaltenstypisch einzigartige und unkopierbare Merkmale einer Person	<ul style="list-style-type: none"><li>• <b>Fingerabdruck</b></li><li>• Gesicht, Iris, Retina</li><li>• Gang, Tastaturanschläge</li></ul>

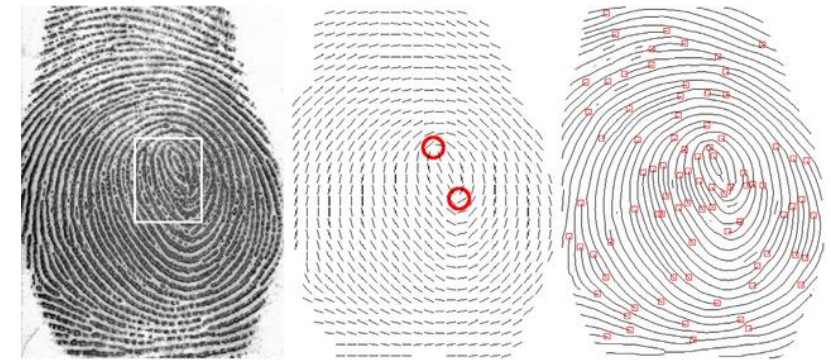


Quelle: Uni Potsdam, Authentifikation WS 17\_18



- Physiologische oder verhaltenstechnische eindeutige Merkmale

- Fingerabdrücke
- Gesicht, Iris, Retina
- Tippverhalten auf der Tastatur
- Gang



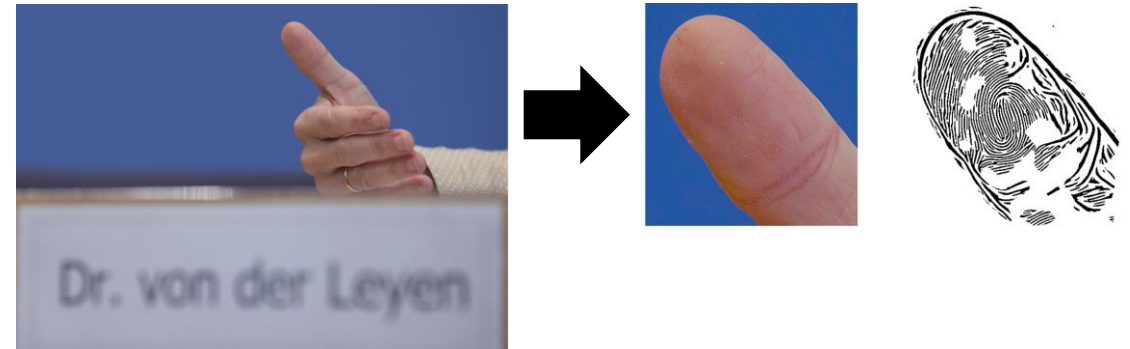
Quelle: [JNR16]

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



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



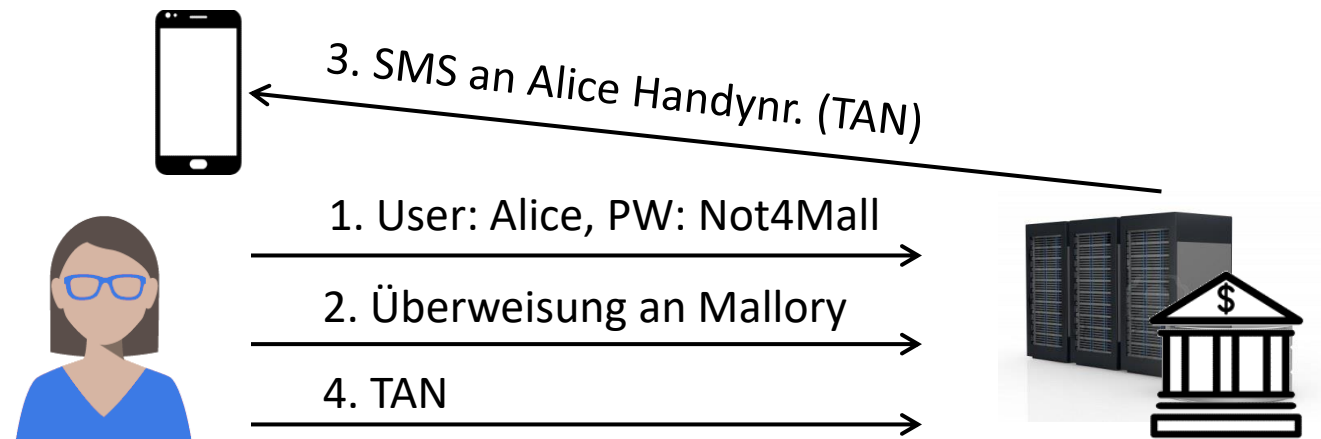
# Vor- und Nachteile der Authentifizierungstechniken



Merkmal	Vorteile	Nachteile
Wissen	<ul style="list-style-type: none"><li>• Einfach zu implementieren</li><li>• Theoretisch sicher</li><li>• Keine zusätzliche Technik</li></ul>	<ul style="list-style-type: none"><li>• Sicherheit abhängig vom gewählten Passwort</li><li>• Schwer zu merken bei vielen Zugängen</li><li>• Kopieren nicht bemerkbar</li></ul>
Besitz	<ul style="list-style-type: none"><li>• Kein Merken notwendig</li><li>• Entwendung ist bemerkbar</li><li>• Standardmäßig hohe Sicherheit</li></ul>	<ul style="list-style-type: none"><li>• Ggf. Physikalische Schnittstelle benötigt (Smart Card Reader)</li><li>• Aufwändig in der Umsetzung</li><li>• Muss von Nutzer*in mittransportiert werden</li></ul>
Biometrie	<ul style="list-style-type: none"><li>• Kein Transport oder Merken notwendig</li><li>• Eindeutig pro Mensch</li></ul>	<ul style="list-style-type: none"><li>• Physikalischer Scanner benötigt</li><li>• Externe Faktoren können zu Fehlern führen</li><li>• Kopieren manchmal nicht bemerkbar</li><li>• Merkmal nicht wechselbar → Ein Leak reicht um Sicherheit des Merkmals zu korrumpieren</li><li>• Kann auch gegen Nutzer*in verwendet werden</li></ul>

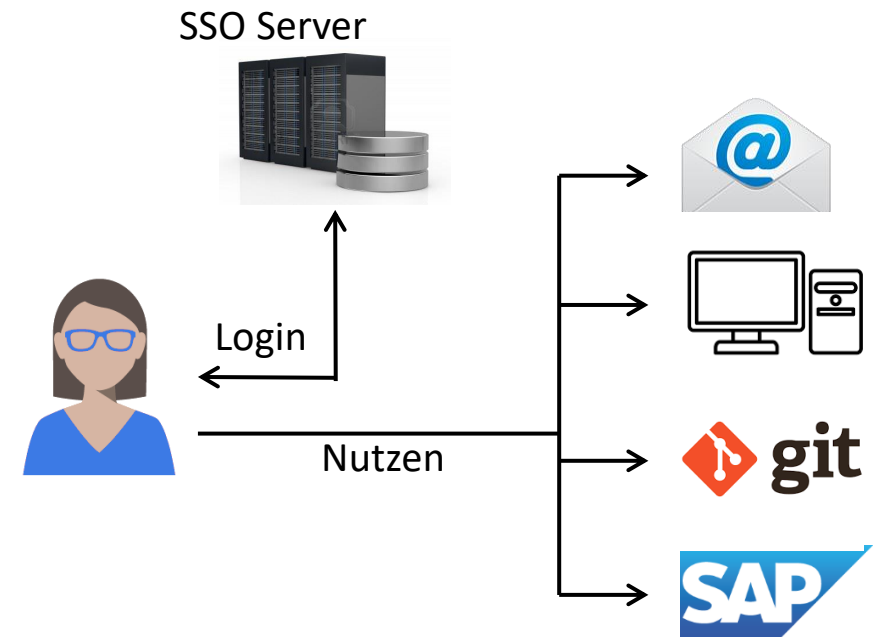
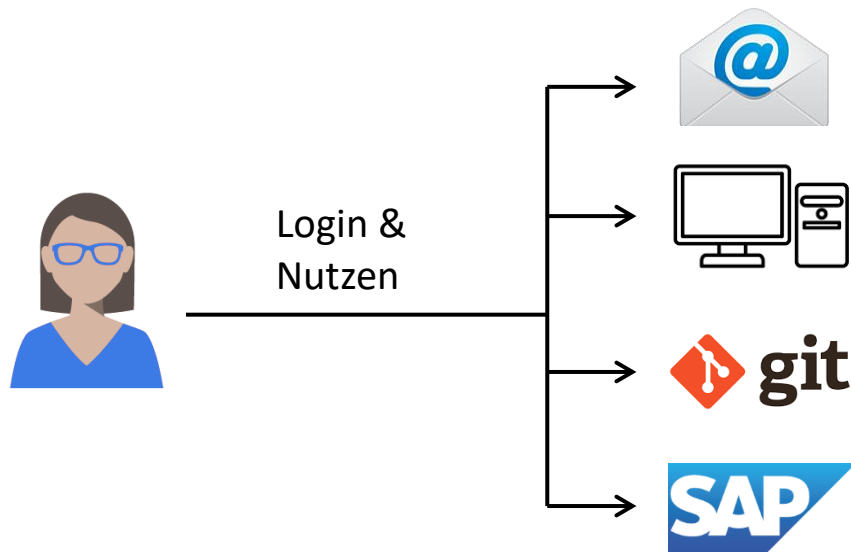


- Mechanismen aus Wissen, Besitz und Biometrie können kombiniert werden um mehr Sicherheit zu erreichen:
  - **Zweifaktor Authentifizierung:** Kombination von zwei Mechanismen aus verschiedenen Kategorien
  - **Multifaktor Authentifizierung:** Kombination von mehr als zwei Mechanismen aus verschiedenen Kategorien
- **Beispiel: Online Überweisung**
  - Wissen (Passwort)
  - Besitz (SIM Karte / Smartphone)





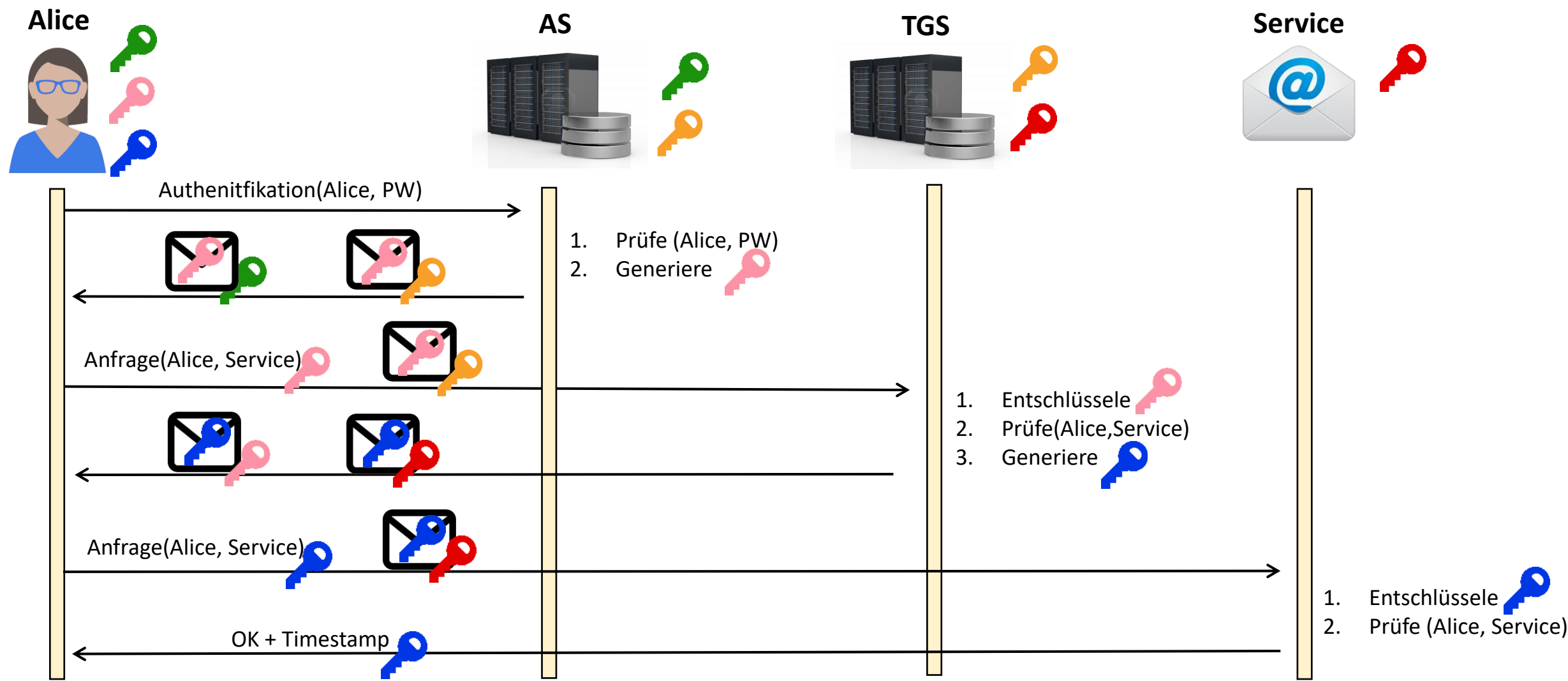
- Viele Systeme verlangen eine individuelle Authentifizierung
- **Lösung: Single Sign-On (SSO) Systeme**, die zentrale Authentifizierung ermöglichen
  - Unternehmensnetzwerke: Kerberos
  - Privat Online: OAuth/OpenID Systeme (Google, Facebook, ...)





- Kerberos 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.

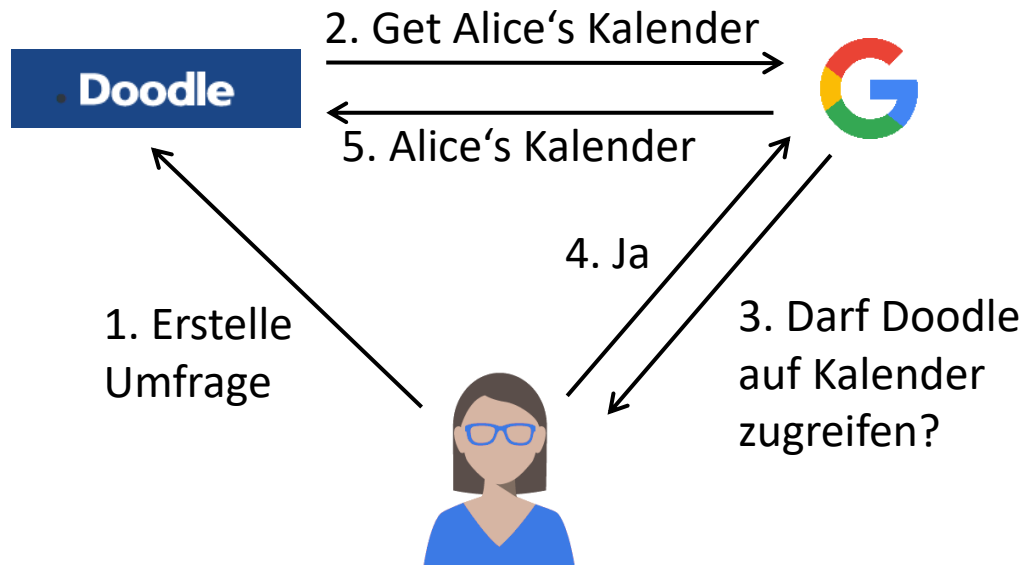
# Kerberos Protokoll



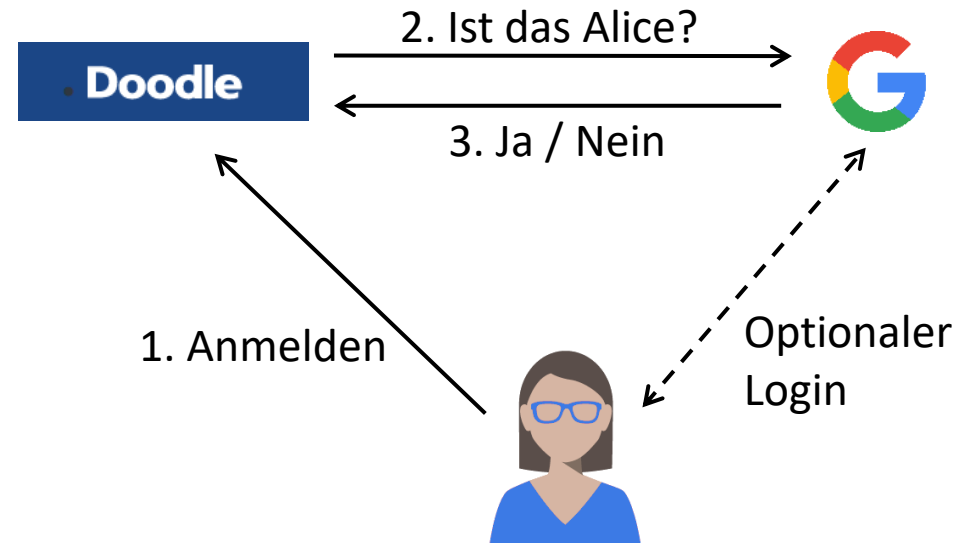


- OAuth 2.0 ermöglicht **Autorisierung für Webseiten**, die auf Accounts anderer Webseiten zugreifen wollen, wird aber oft zur **Authentifizierung** als Single Sign-On verwendet

## Autorisierung



## Authentifizierung



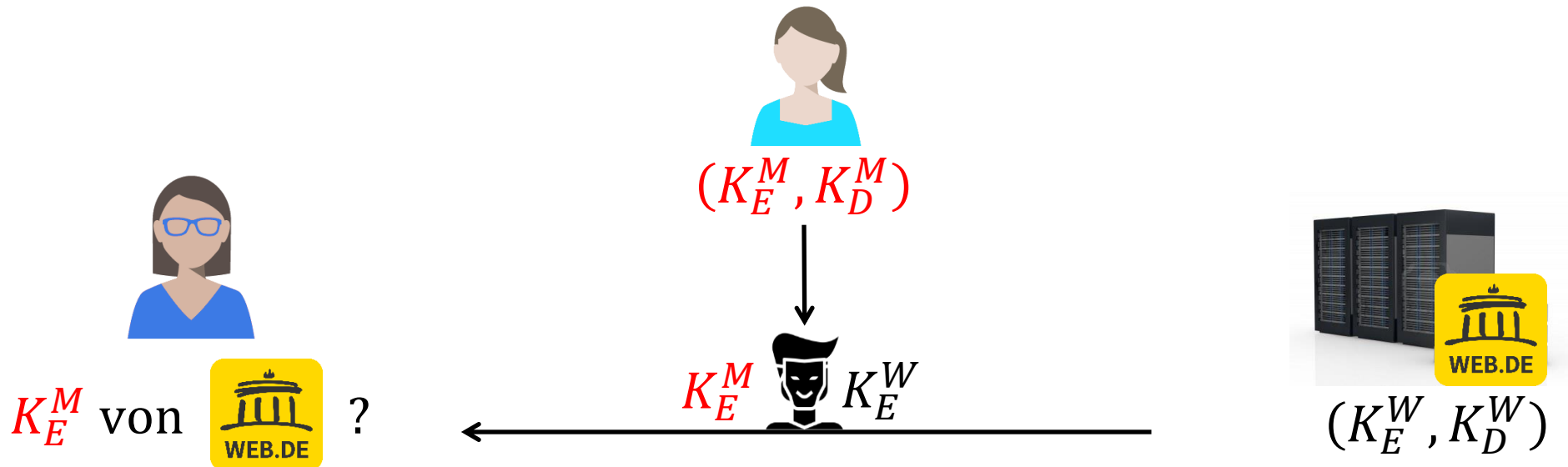
The screenshot shows the Doodle login page. It includes a blue header with the Doodle logo, links for 'Anmelden' (Login) and 'Registrieren' (Register), input fields for 'E-Mail-Adresse' and 'Passwort', and a green 'Anmelden' button. Below the button is a link for 'Passwort vergessen?' (Forgot password?). At the bottom, there are four buttons for social login: 'Mit Microsoft anmelden', 'Mit Google anmelden', 'Mit Facebook anmelden', and 'Mit SSO anmelden'.



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



- **Bedrohung:** Mallory tauscht öffentlichen Schlüssel aus um Kommunikation abzufangen
- **Ziel:** Alice kann prüfen ob der öffentliche Schlüssel zum korrekten Ziel gehört



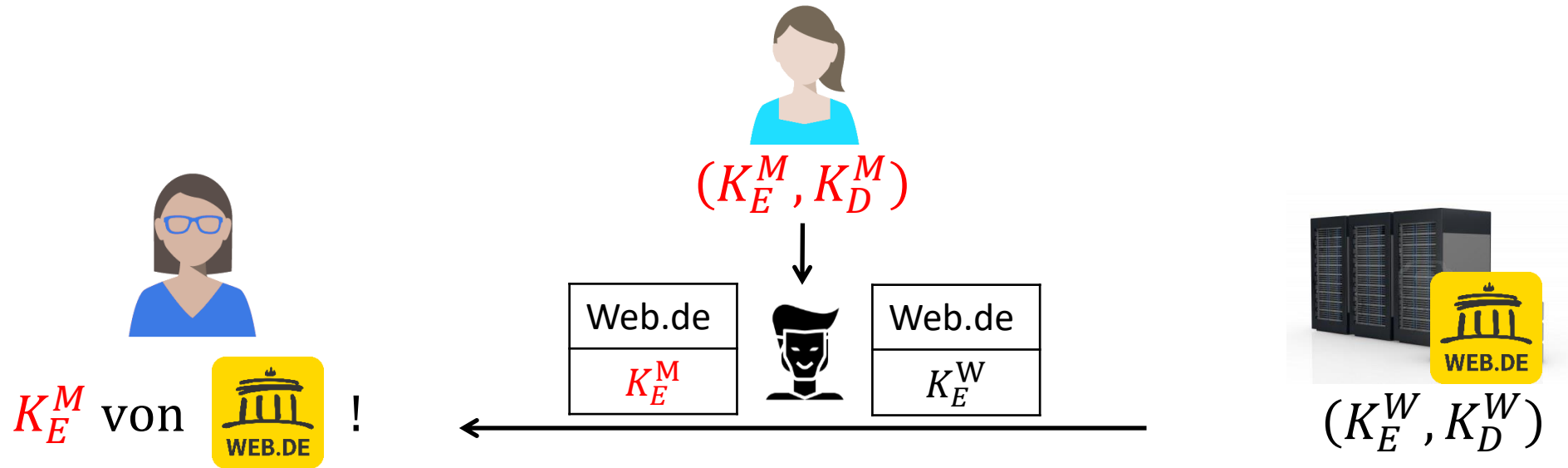




- **Idee:** Verknüpfen des öffentlichen Schlüssels des Webserver mit Attributen, die nachgeprüft werden können (z.B. DNS-Name oder IP).
- Zertifikate können weitere Informationen enthalten:
  - Gültigkeitsdauer (von / bis)
  - Verwendungszweck des Schlüssels (Signieren/Verschlüsseln)
  - Anwendungsspezifische 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



- **Problem:** Authentizität des Zertifikates ist nicht sichergestellt.
- Wenn digitale Signaturen nicht reichen → Mehr digitale Signaturen!



- 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 angehängen
- Nutzer\*in kann Zertifikat mit Schlüssel  $K_E^{CA}$  der CA prüfen

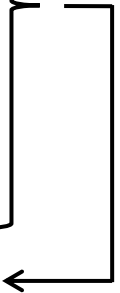


Schlüsselpaar  $(K_E^{CA}, K_D^{CA})$

**Web.de Zertifikat**

DNS Name:	Web.de
Öffentlicher Schlüssel:	$K_E^W$
Gültig von:	08/Nov/2021
Gültig bis:	09/Dez/2022
CA:	Name / Link
Hash:	0x12 34
Signatur Zertifikat:	$S$

$$S = Dec_{K_D^{CA}}(0x1234)$$





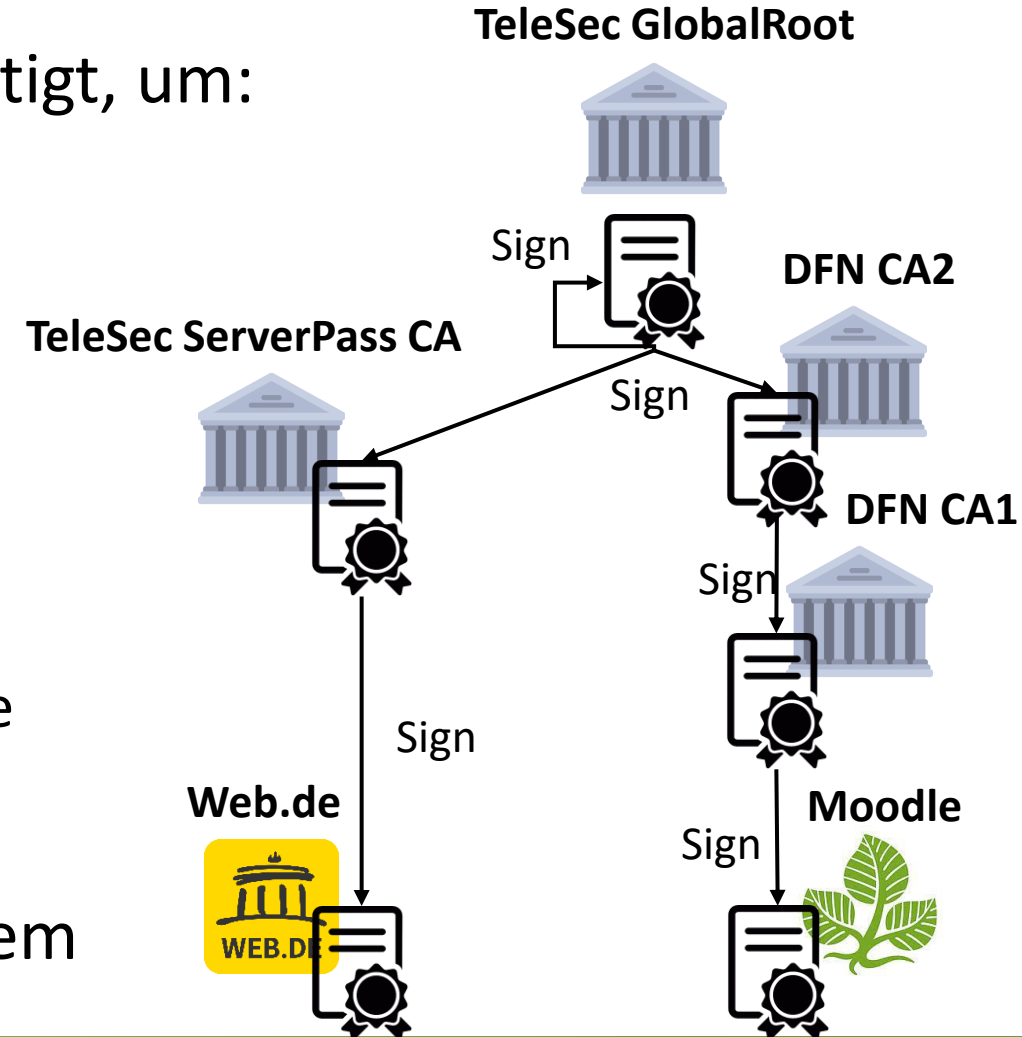
- Eine **Public Key Infrastruktur (PKI)** wird benötigt, um:

- 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



Alice kennt Wurzel-CA



1. Web.de Zertifikat unbekannt
2. Prüfe Gültigkeit und Identität
3. Prüfe Signatur



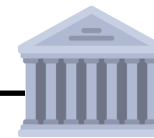
- a. CA Zertifikat unbekannt
- b. Hole CA Zertifikat
- c. Prüfe Gültigkeit und Identität
- d. Prüfe Signatur



- i. Wurzel-CA Zertifikat bekannt und verifiziert
- ii. Signatur CA Zertifikat OK

- e. CA Zertifikat Authentisch
- f. Web.de Zertifikat Signatur OK

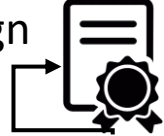
4. Web.de Zertifikat Authentisch



TeleSec GlobalRoot



Sign



TeleSec  
ServerPass CA



Sign



Sign

Web.de





- 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



[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](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](https://wiki.mozilla.org/CA:WoSign_Issues)

[PFS14]: [https://www.ifca.ai/pub/fc14/paper\\_100.pdf](https://www.ifca.ai/pub/fc14/paper_100.pdf)