Authentifikation

Vom Passwort zu FIDO2 und Zero-Knowledge

/SSS Webinar, 17. Februar 2022

whoami

- Nationales Zentrum für Cybersicherheit NCSC
- eSECURITY Technologies Rolf Oppliger
- Universität Zürich
- Artech House

→ rolf-oppliger.ch or rolf-oppliger.com

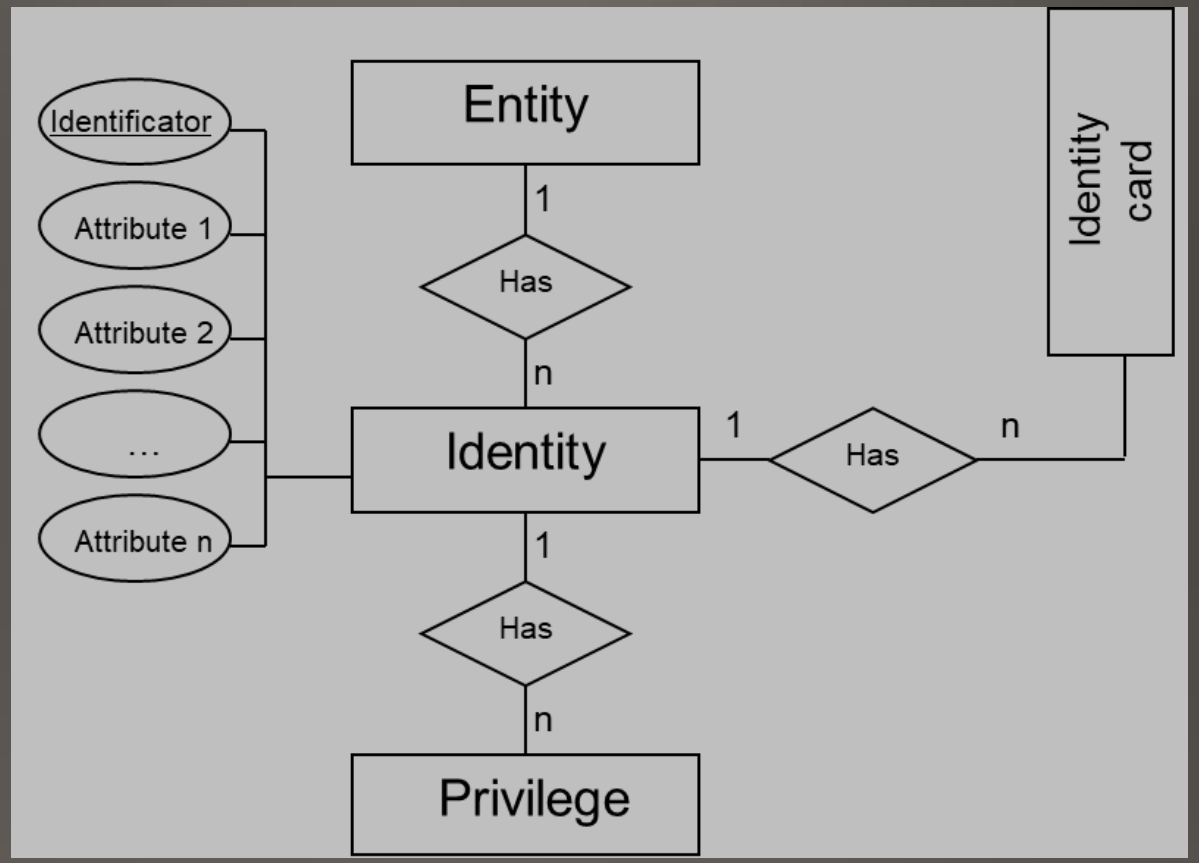
eSECURITY® Academy esecurity.academy

Reiseplan

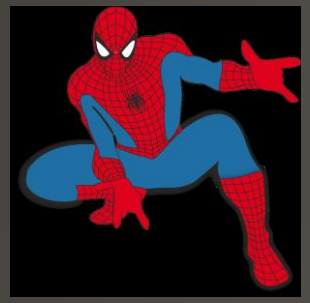
1. Vorbereitung

- Der englische Begriff «Authentication» wird als «Authentifikation» oder «Authentifizierung» übersetzt (und «Authentikation» wird synonym verwendet)
- Authentifiziert werden
  - Entitäten (z.B. Personen, Prozesse, Systeme, …)
    → **Entitätenauthentifikation** («Entity Authentication» bzw. «Peer Entity Authentication»)
  - Nachrichten → **Nachrichtenauthentifikation** («Message Authentication»)
- Im Fokus steht die Entitätenauthentifikation für Personen (d.h. Personen-authentifikation)

- Eine **Identität** ist eine Menge von Attributen, die einer Entität (z.B. Person) zugeordnet sind
- Bestimmte Attribute können eindeutige **Identifikatoren** sein
- Gilt auch für «digitale» oder «elektronische» Identitäten
- Umgangssprachlich werden solche Begriffe verwendet, um hervorzuheben, dass sich eine Entität im Cyberraum bewegt
- Kontraverses Thema



Identifikation

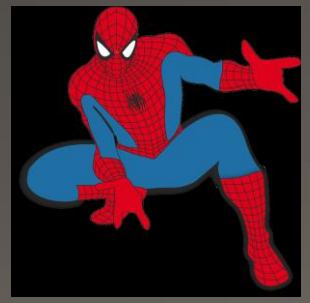


Ich bin (heisse) Spiderman

Authentifikation

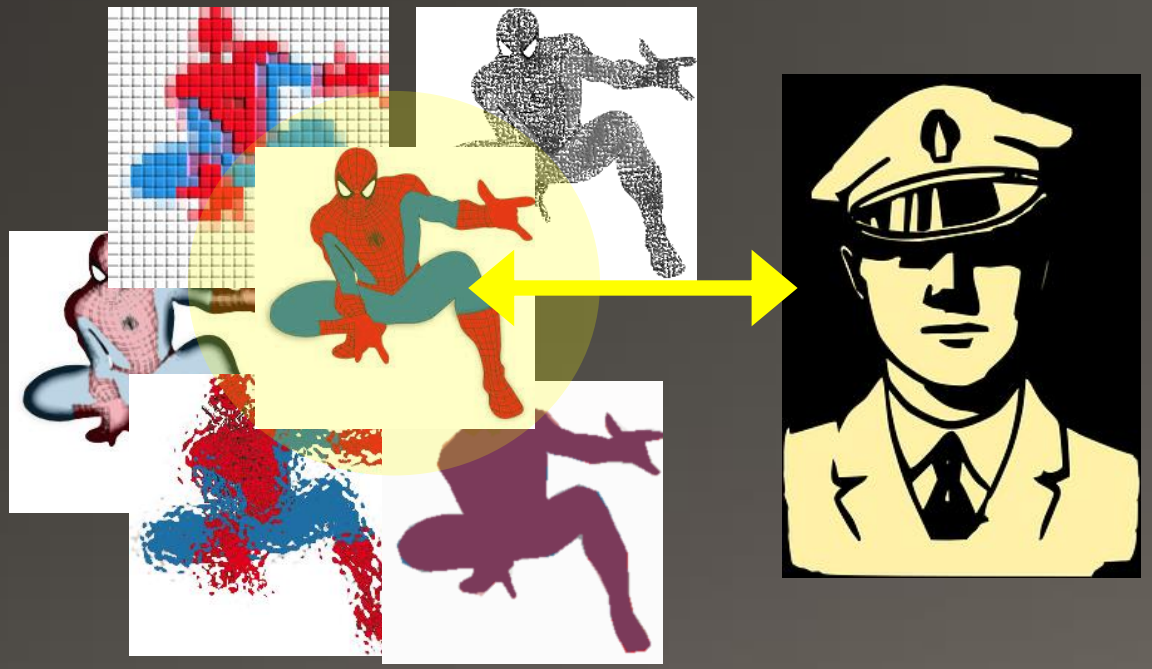
Irgend eine Form eines Beweises

Autorisation



Straffreie Bewegung im 3-dimensionalen Raum zwecks Bekämpfung des «Bösen»

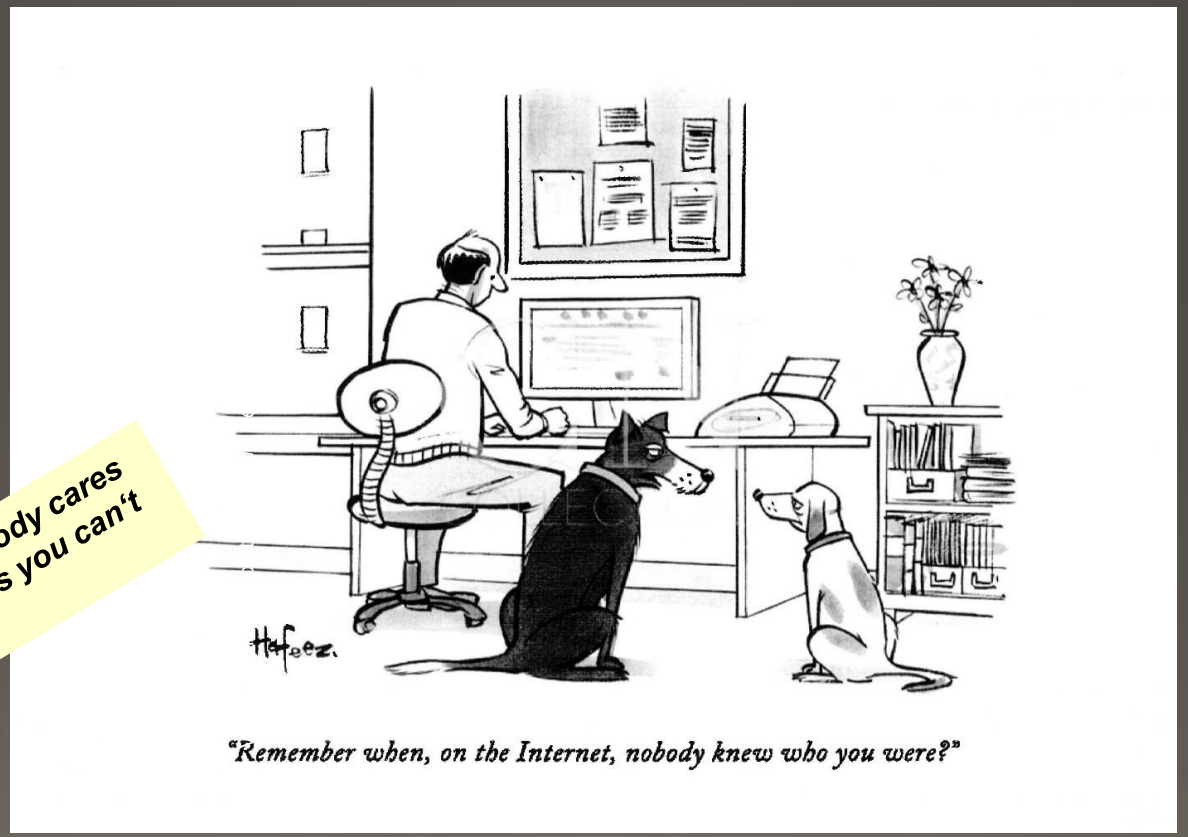
- Grundsätzlich geht es bei der Personenauthentifikation um einen Tradeoff zwischen **Sicherheit** und **Benutzerfreundlichkeit** / **Bequemlichkeit** («usability» / «convenience»)



- Eine Authentifikation ist aber nicht für jeden Geschäftsprozess eine zwingende Voraussetzung

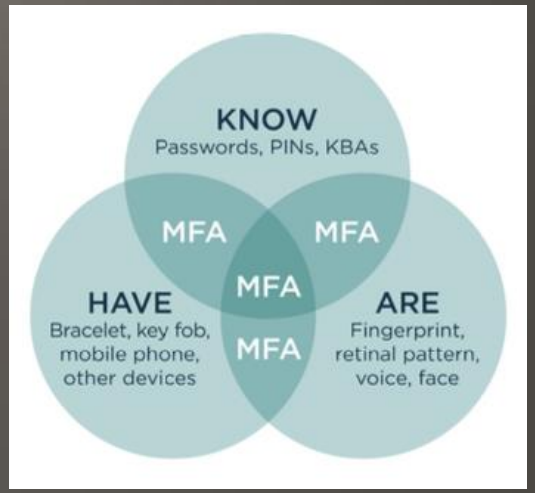
© Peter Steiner's famous cartoon in *The New Yorker* (July 5, 1993)

«On the Internet, nobody cares you're a dog - unless you can't pay your debts.»



© Peter Vidani's follow-up cartoon in *The New Yorker* (February 23, 2015)

- Ansätze
  - «Etwas haben» (z.B. Schlüssel, Billet, Handy, …)
  - «Etwas wissen» (z.B. PIN, Passwort, Schlüssel, …)
  - «Etwas sein» (z.B. biometrische Merkmale)
  - «Irgendwo sein» (z.B. Telefonnummer, IP-Adresse, GPS, WLAN, …)
- In der IT werden in erster Linie «Etwas wissen»-Ansätze verwendet
- … oft in Kombination mit anderen Ansätzen
  → Multi-Factor Authentication (MFA)
- 2FA ist besonders weit verbreitet

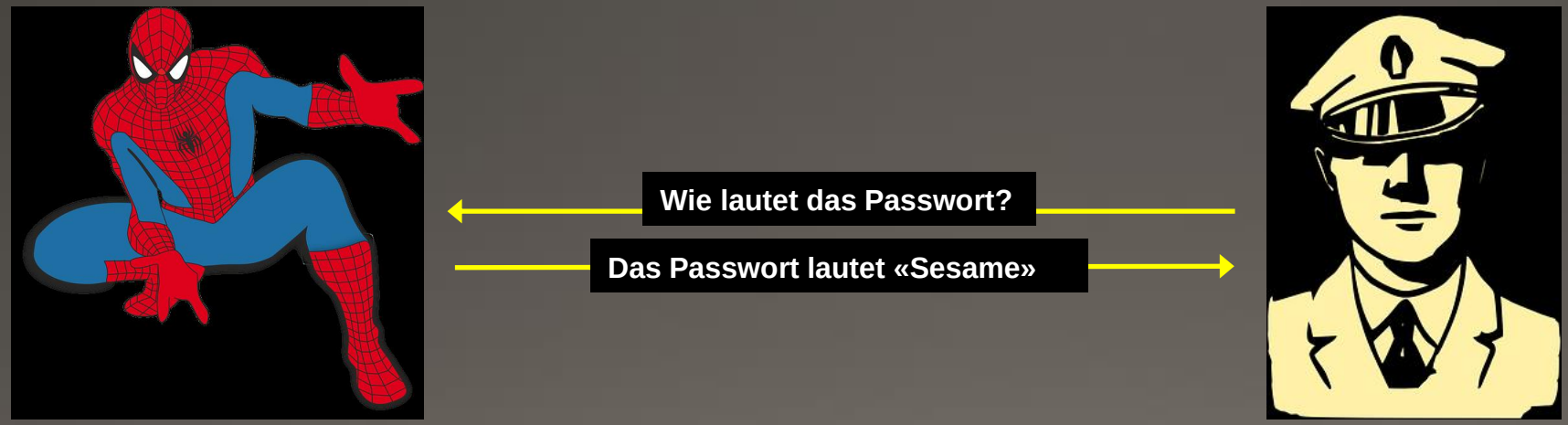


- Eine Frage, die sich in der Praxis stellt, betrifft die Stärke eines Authentifikations- und Identitätsnachweismittels (Mittel)
- Unterscheidungsparameter (Dimensionen)
  - Wer hat den Benutzer registriert und wie (z.B. persönliches Vorsprechen mit Identitätsausweis oder automatisierter «Ausroll»-Prozess)?
  - Auf welchen Technologien und Techniken basiert das Mittel?
  - Schützt das Mittel z.B. vor «Sniffing»- und «Replay»-Angriffen?
  - Wie wird das dem Authentifikationsverfahren zugrunde liegende Geheimnis geschützt (z.B. Hardware)?
  - Für welche Anwendungen / Use Cases soll das Mittel verwendet werden (z.B. E-Government)?
  - …

- Beispielsklassifikation
  - **Tief:** Die Authentifikationsinformation ist statisch und immer gleich, d.h. sie kann von einem Angreifer abgegriffen und z.B. für einen «Replay»-Angriff und einer Identitätstäuschung missbraucht werden
  - **Mittel:** Die Authentifikationsinformation ist dynamisch (d.h. ändert sich bei jeder Authentifikation) und kann entsprechend nicht für einen «Replay»-Angriff und eine Identitätstäuschung missbraucht werden
  - **Hoch:** Die Authentifikationsinformation ist dynamisch und hängt von einem kryptografischen Schlüssel ab, der in einem dedizierten Hardware-Modul («auslesesicher») gespeichert ist
  - **Hoch+:** Das Mittel erfüllt die Anforderungen der Stufe «hoch» und ist von einer zuständigen Stelle formal anerkannt

2. Passwort

- Passwörter basieren auf dem «Etwas wissen»-Ansatz und haben eine lange (Entstehungs- und Leidens-) Geschichte
- Viele Punkte gelten auch für «Bearer Tokens» (z.B. Cookies)



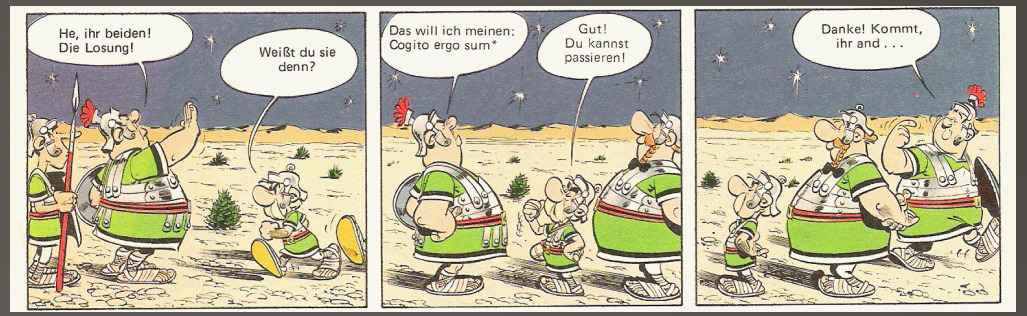
- Hohe Benutzerfreundlichkeit
- Einfache Implementierbarkeit
- Flexibilität im Einsatz
- …

25 Of The Most Popular Passwords

1	123456	6	123456789	11	admin	16	starwars	21	hello
2	password	7	letmein	12	welcome	17	123123	22	freedom
3	12345678	8	1234567	13	monkey	18	dragon	23	whatever
4	qwerty	9	football	14	login	19	passw0rd	24	qazwsx
5	12345	10	iloveyou	15	abc123	20	master	25	trustno1

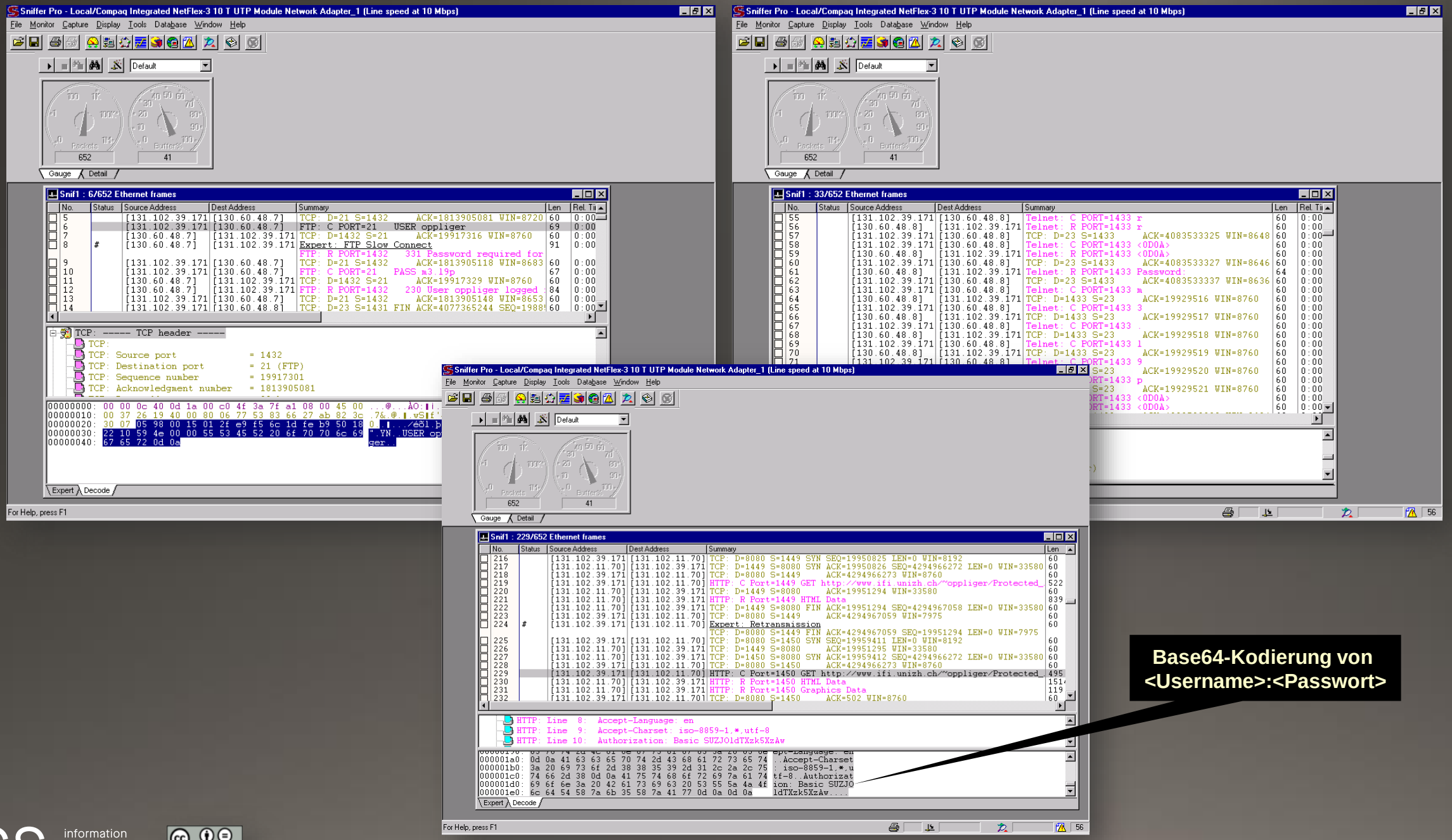
- Geringe Sicherheit
  - Schwache Passwörter
  - «Social Enineering»-Angriffe (Video)
  - «Password Guessing»-Angriffe
  - «Sniffing»-Angriffe
  - …

«Brute Force»
(+/- Precomputation z.B. mit Hilfe von «Rainbow»-Tabellen)



© Goscinny/Uderzo, Grosser Asterix-Band X, „Asterix als Legionär“, EHAPA-Verlag GmbH, Stuttgart, 1973, p. 41

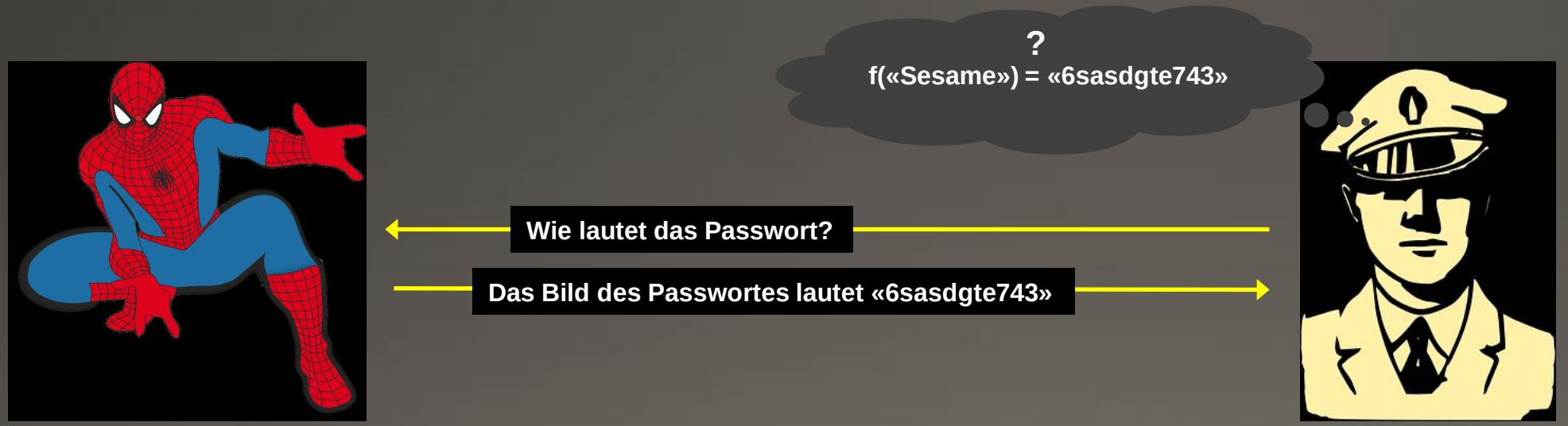
© Goscinny/Uderzo, Grosser Asterix-Band X, „Asterix als Legionär“, EHAPA-Verlag GmbH, Stuttgart, 1973, p. 40



3. Verbesserungen

- «Quick Wins»
- Einsatz symmetrischer Kryptografie
- Einsatz asymmetrischer Kryptografie

- **«Quick Wins»**
  - Die Wahl von schwachen Passwörtern können (technisch und/oder organisatorisch) verhindert werden (→ proaktives Passworttesten)
  - Server-seitig können Passwörter nicht als Klartext sondern als Bild einer Einweg-Funktion f abgespeichert sein



- «Password Guessing»-Angriffe können mit Hilfe von «Salt and Pepper» erschwert werden
  - Das «Salt» dient der Individualisierung von f (dadurch müssen auch die Angriffe individualisiert werden)
  - Das «Pepper» dient der zusätzlichen Verschlüsselung der Passwortdatei (dadurch wird der Zugriff erschwert)

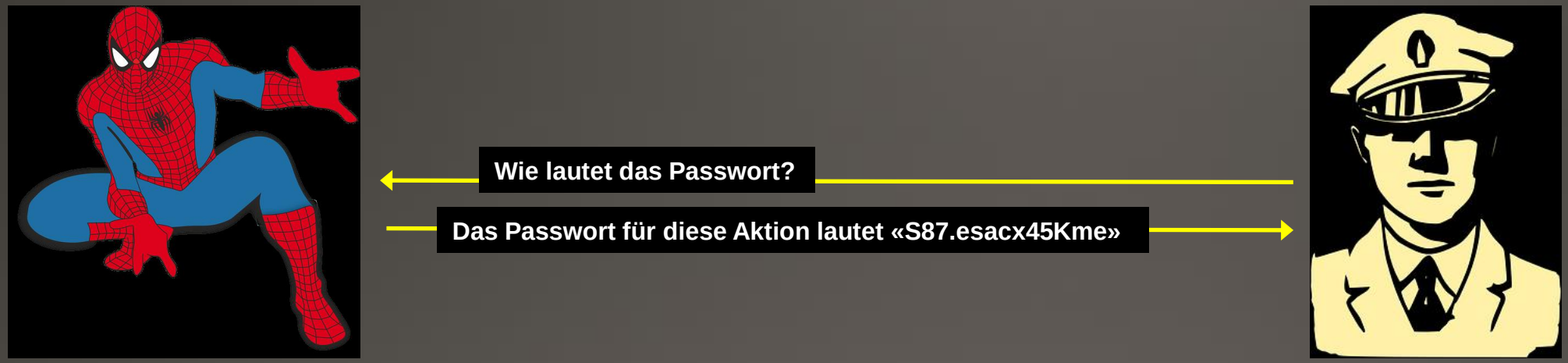
- Die wirksame Verhinderung solcher Angriffe erfordert den Einsatz eines Passwort-basierten «Authenticated Key Exchange» (PAKE) Protokolls
- Grundidee: Anstelle einer Passwortübertragung wird das Passwort verwendet, um einen Schlüsselaustausch (z.B. Diffie-Hellman Schlüsselaustausch) zu verschlüsseln bzw. zu authentifizieren (→ asymmetrische Kryptografie)
- Beispiele: EKE, AKE, PAKE, SRP, SPEKE, OPAQUE, cPace, …

- Bei einem **symmetrischen PAKE** kennt der Server das Benutzerpasswort
- Bei einem **asymmetrischen PAKE (aPAKE)** kennt der Server das Passwort nicht (d.h. er kennt nur den Hashwert)
- Unter Verwendung von «Salt» können Precomputation-Angriffe z.B. mit Hilfe von «Rainbow-Tabellen» erschwert werden
- Bei einem **«starken» aPAKE** Protokoll muss das verwendete «Salt» gegenüber dem zugreifenden Benutzer nicht offengelegt werden
- **OPAQUE** (OPRF + PAKE) ist ein starkes aPAKE Protokoll, das den Stand der Technik bei der Passwort-basierten Authentifikation markiert
- Das Protokoll wird z.B. auch für Ende-zu-Ende-verschlüsselte Backups in WhatsApp verwendet (Einstellungen > Chat-Backup)

- Das Erzwingen periodischer Passwort-wechsel («Password Aging») bringt aus der Sicht der Sicherheit nicht viel bzw. ist unter Umständen sogar kontraproduktiv
- Insbesondere auch zur Verhinderung von «Sniffing»-und «Replay»-Angriffen sind stärkere Authentifikationsverfahren er-forderlich

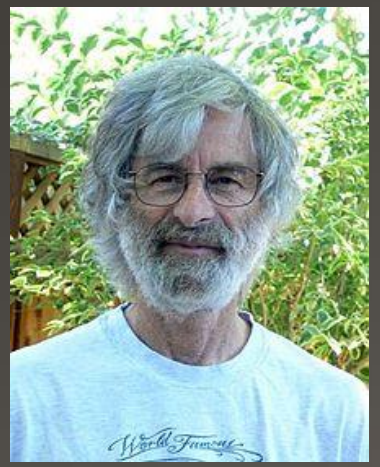
- Dabei spielt die (symmetrische und/oder asymmetrische) Kryptografie eine entscheidende Rolle

- **Einsatz symmetrischer Kryptografie:** Einweg-Passwörter und Challenge-Response-Verfahren (OTP)



- Solche Verfahren bedingen ein (symmetrisches und geteiltes) Geheimnis
- Es gibt sehr viele Formen und Varianten, wie z.B. Streichlisten, TAN-Listen, «normale» OTP- und Challenge-Response-Verfahren und -Systeme, sowie SMS-Verifikationscodes

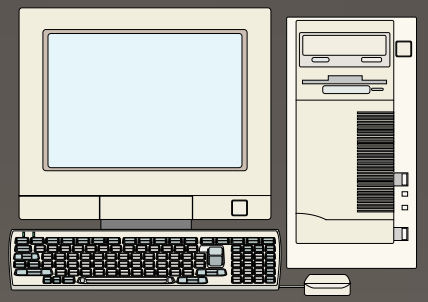
- Beispiel 1: S/KEY (Bellcore) und One-time Passwords In Everything (OPIE) → Leslie Lamport (1981)
- Verbreitung auch aufgrund von U.S. amerikanischen Exportbeschränkungen für kryptografische Systeme



$h^t(pw)$

Authentifikation mit OTP $h^{t-i}(pw)$ für i=1, ...,t-1

Ersatz von $h^{t-i+1}(pw)$ mit $h^{t-i}(pw)$ in der Passwortdatei



$h(a_i) \stackrel{?}{=} a_{i-1}$

$h^{1000}(pw)$	$= a_0$
$h^{999}(pw)$	$= a_1$
$h^{998}(pw)$	$= a_2$
...	
$h^2(pw)$	$= a_{998}$
$h(pw)$	$= a_{999}$

- Beispiel 2: SecurID von RSA Security

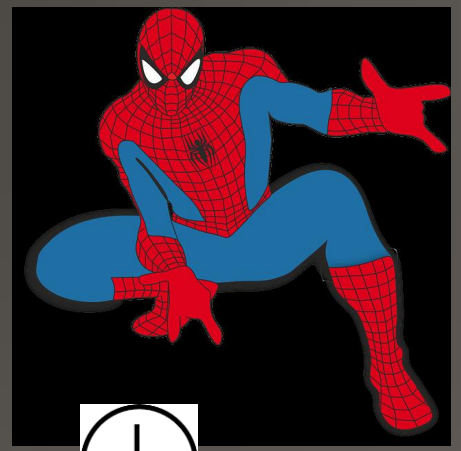
$x_0 = t_0$

$OTP_1 = AES(s,x_0)$

$OTP_2 = AES(s,x_1)$

…

$OTP_n = AES(s,x_{n-1})$



$i = 1,\ldots,n:$

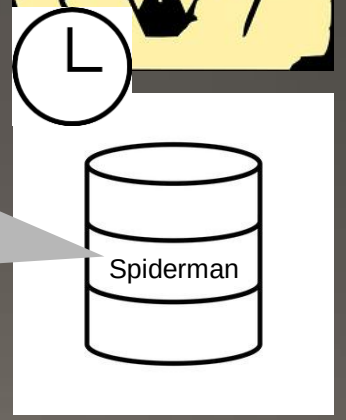
$OTP_i = AES(s,x_{i-1}) = AES^i(s,x_0)$

s = Seed (Schlüssel)
t_0 = Startzeit
t = Aktuelle Zeit
Δt = Intervall

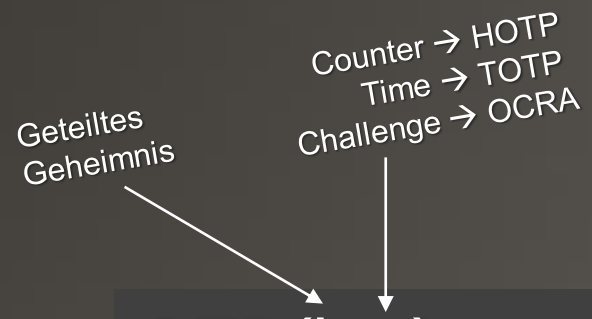
$n = (t-t_0)/\Delta t$

s = Seed (Schlüssel)
t_0 = Startzeit
t = Aktuelle Zeit
Δt = Intervall

$n = (t-t_0)/\Delta t$



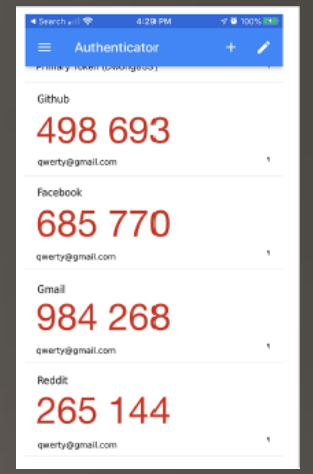
- Beispiel 3: Open AuTHentication (OATH) Standards



$$OTP(k,n) = truncate(HMAC(k,n))$$

Google Authenticator

Microsoft Authenticator



- HMAC-based One-time Password (OATH-HOTP) → RFC 4226
- Time-based One-time Password (OATH-TOTP) → RFC 6238
- OATH Challenge-Response Algorithm (OCRA) → RFC 6287

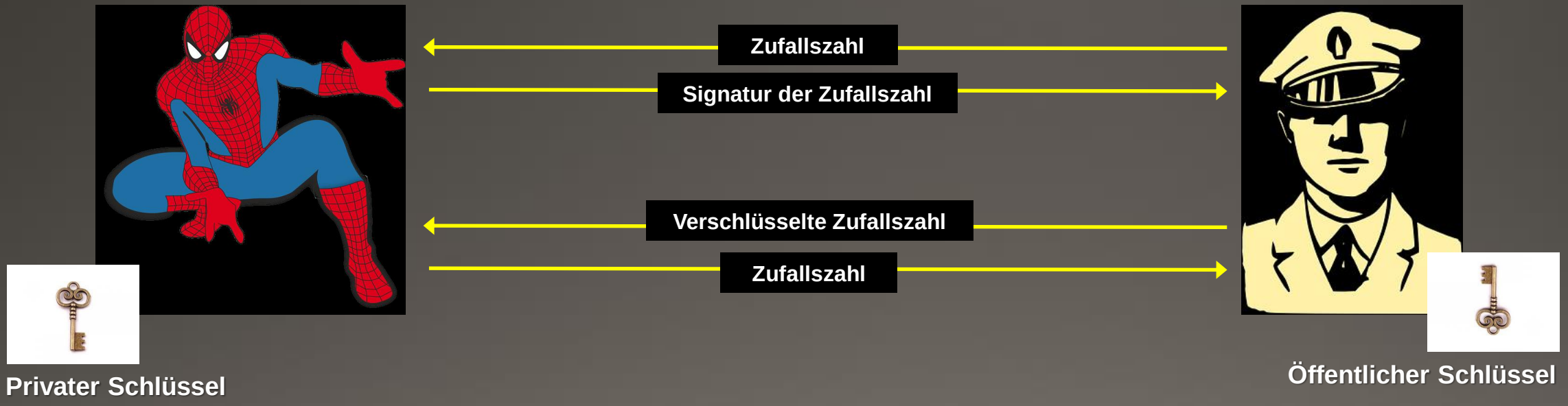
- Gegenüber Passwörtern stark verbesserte Sicherheit
- Flexible Implementierbarkeit (in Hard- und Software)
- …

- Z.T. schwierige Benutzerführung
- Zusatzlösung (nicht integriert)
- Zentrale Datenbank
- …

- Zentrale Datenbank → «Single Point of Failure»
- Das ist insbesondere dann problematisch, wenn die Datenbank nicht selbst betrieben wird (z.B. «RSA Security Breach» in 2011)
- Ein paar OTP-Verfahren benutzen einen unabhängigen zweiten Kanal, der auch in die Sicherheitsbetrachtung mit einbezogen werden muss (z.B. SMS-Verifikationscode, Mobile ID, …)
- Der Teufel steckt im Detail

- **Einsatz asymmetrischer Kryptografie:** Mit Hilfe eines privaten Schlüssels kann sich ein Benutzer authentifizieren (die Verifikation erfolgt mit dem dazugehörigen öffentlichen Schlüssel)

  → Eine zentrale Datenbank (mit Geheimnissen) ist nicht erforderlich

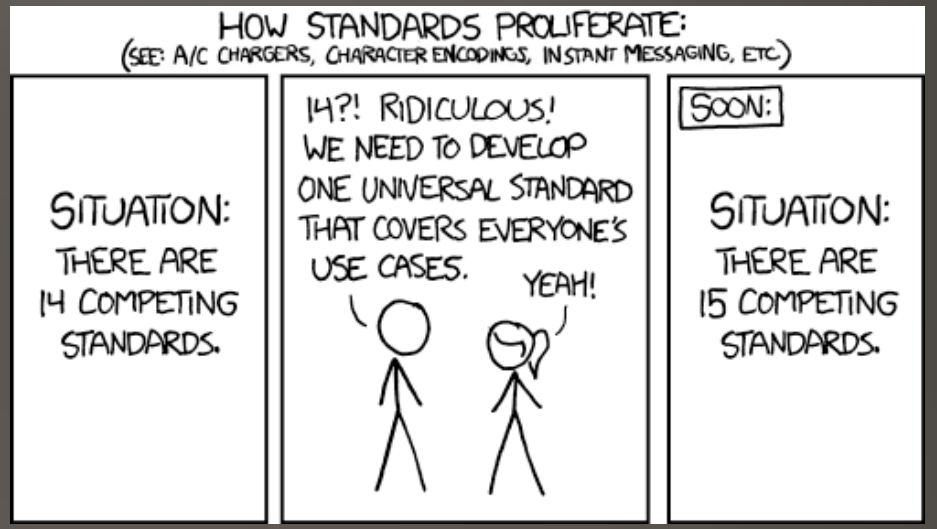


- Im einfachsten Fall werden die öffentlichen Schlüssel manuell verteilt und installiert (z.B. OPAQUE, Secure Shell (SSH), …)
- Wenn die Authentizität der Schlüssel nachträglich überprüft werden kann wird das entsprechende Vertrauensmodell als **Trust On First Use (TOFU)** bezeichnet
- TOFU stellt eine zunehmend weit verbreitete Alternative zu Zertifikaten und **Public Key Infrastrukturen (PKI)** bzw. dem **Web of Trust (WOT)** dar
- Allerdings ist die Skalierbarkeit von TOFU begrenzt und Benutzer überprüfen die Authentizität der Schlüssel kaum (z.B. Signal / WhatsApp)

- Beim Einsatz von Zertifikaten und PKI kommen erschwerend hinzu
  - Rollout
  - Separierung (Authentifikation / Signatur / Verschlüsselung)
  - Anerkennung in «Trust Stores»
  - Revozierung (z.B. CRL, OCSP ± Stapling, Certificate Transparency, …)
  - …
- Beim Einsatz von Smartcards kommen zusätzlich noch erschwerend hinzu
  - Schlechte Unterstützung auf den meisten Plattformen (Betriebssystemen)
  - «Schwieriger» Formfaktor
  - …

4. FIDO2

- Seit 2013 verfolgt die Fast IDentity Online (FIDO) Allianz das Ziel, offene und lizenzfreie Industriestandards für die (weltweite) Authentifizierung im Internet (ohne Passwörter) zu entwickeln
- Vorgängerprotokolle
  - Universal Authentication Framework (UAF)
  - Universal 2nd Factor (U2F)
- FIDO2 ist aus dem Bestreben entstanden, UAF und U2F zu vereinen und dadurch eine möglichst grosse Marktdurchdringung zu erreichen



© https://xkcd.com/927/

Web Client (Browser)

Web Server (Relying Party)

WebAuthn (W3C)

Benutzer

Login Request

App ID, Challenge

Counter, Response

Challenge

Response

Optional:
«Device Attestation»

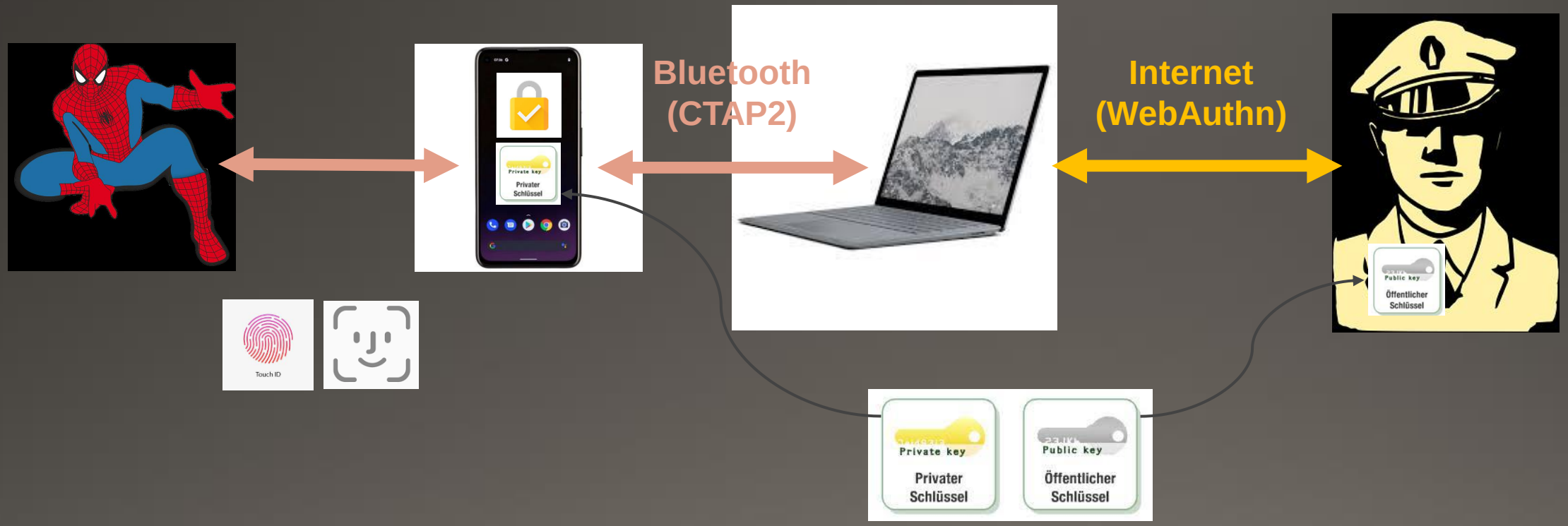
Optional:
«User (Presence) Verification»

Client-To-Authenticator Protocol 2 (CTAP2)

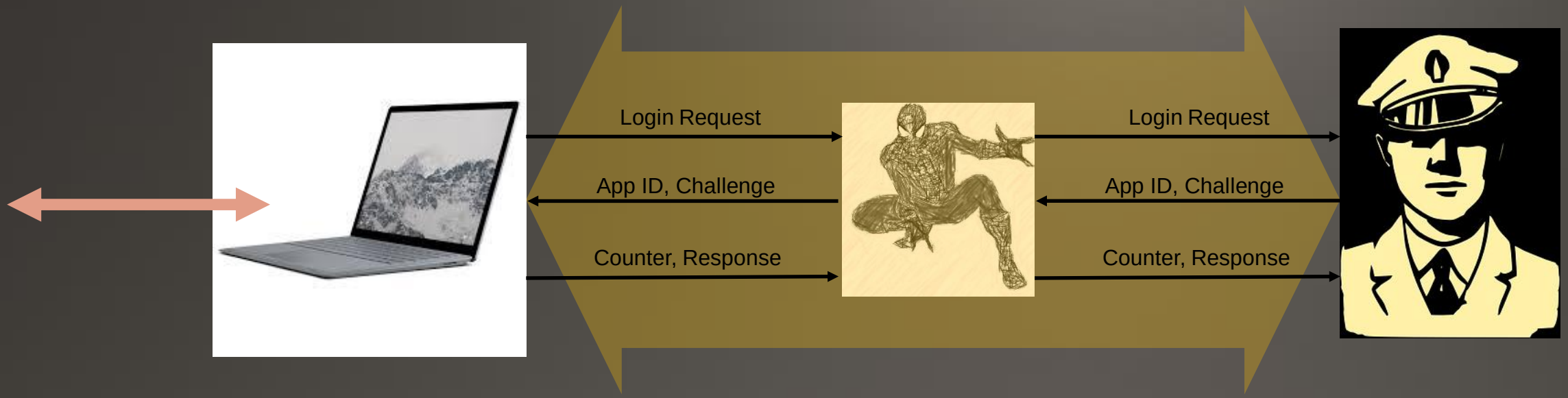
Authenticator

Für jede Applikation erzeugt der «Platform» oder «Roaming Authenticator» ein asymmetrisches Schlüsselpaar (zum Signieren von Challenges bzw. Erzeugen von Responses)

- Beispiel: Google Smart Lock



- FIDO2 ist sehr gut aufgestellt
  - WebAuthn und CTAP2 werden von allen handelsüblichen Browsern unterstützt
  - Authenticators gibt es in allen möglichen Formfaktoren (inkl. Smartphones)
- Damit kann FIDO2 die Passwort-basierte Authentifikation mit einer sichereren und benutzerfreundlichen Alternative ersetzen
- Problemfelder / Verbesserungsmöglichkeiten
  - Sicherheit der Plattform (insbesondere beim Einsatz von «Platform Authenticators» unter Berücksichtigung von Malware-basierten Angriffen)
  - Beweisbare Sicherheitseigenschaften
  - «Zero-Knowledge»-Eigenschaft
  - MITM-Angriffe
  - …



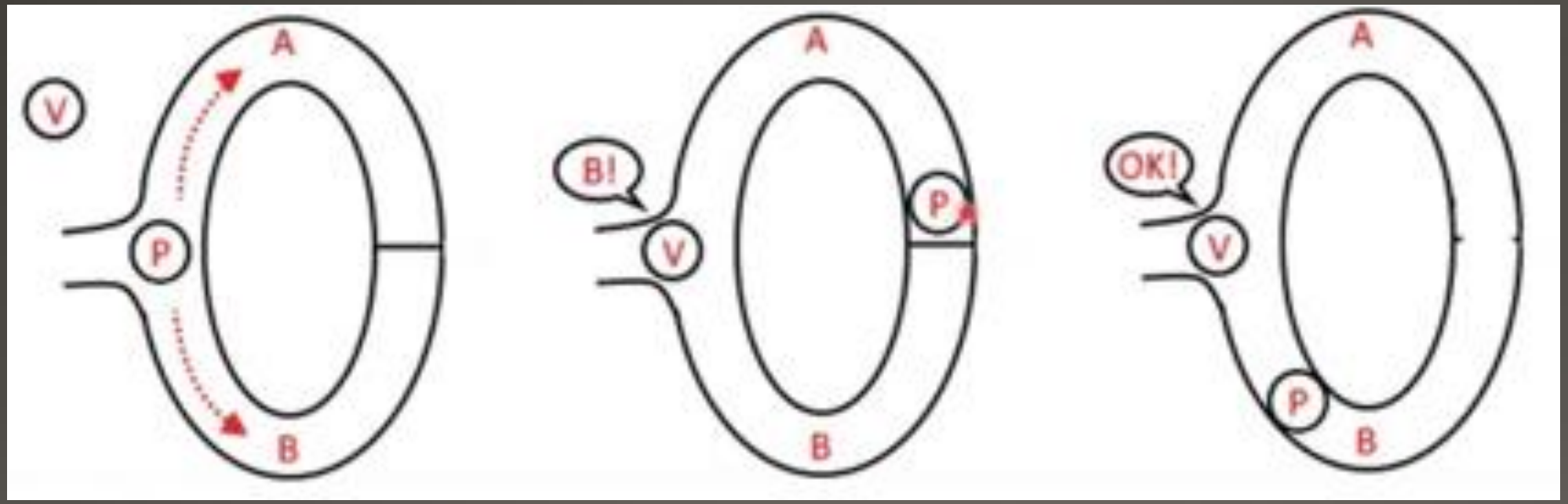
- MITM-Angriffe können mit Hilfe des «Token Binding»-Protokolls (RFC 8471 und RFC 8472) teilweise verhindert bzw. erschwert werden
- Bei der Erzeugung der Response signiert der Client die «Token Binding ID» mit
- Diese ID referenziert ein asymmetrisches Schlüsselpaar, das im Rahmen des TLS-Verbindungsaufbaus (mit Hilfe der TLS-Erweiterung `token_binding`) zwischen Client und Server etabliert worden ist

5. Zero-Knowledge

- Für die Authentifikation bietet sich u.U. auch der Einsatz eines interaktiven Beweissystems an
- Solche Systeme basieren auf «Tatbeweisen»
- Beispiele
  - «Pepsi-Test»
  - Gedankenlesen
  - Kenntnis einer mathematischen Formel (z.B. Faktorisieren)
  - …
- Aufgrund ihrer Interaktivität sind solche Beweise abstreitbar, nicht übertragbar und für aussenstehende Dritte nicht überzeugend

- Interaktive Beweissysteme wurden 1985 vorgeschlagen (Silvio Micali, Shafi Goldwasser und Charles Rackoff)
- Neben Vollständigkeit und Korrektheit kann ein interaktives Beweissystem auch die Zero-Knowledge-Eigenschaft aufweisen
- Formal wird diese Eigenschaft mit der Simulierbarkeit nachgewiesen, d.h. man kann korrekte Protokolltranskripte erzeugen, ohne die zu beweisende Tatsache (z.B. Geheimnis) kennen zu müssen
- Damit eignen sich Zero-Knowledge-Beweise auch für die Authentifikation

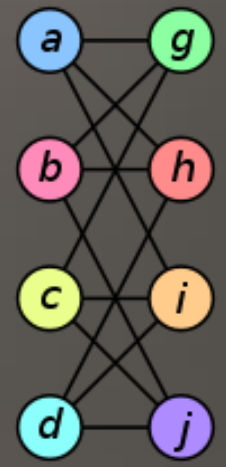
- Beispiel 1: Kenntnis eines Geheimwortes zur Öffnung einer geheimen Tür (→ How to Explain Zero-Knowledge Protocols to Your Children)

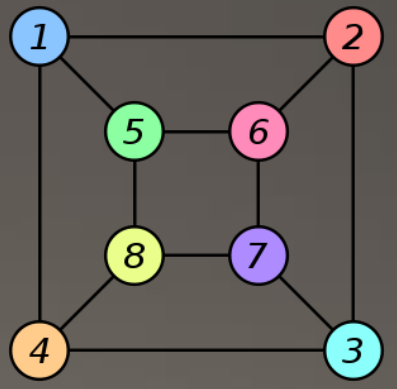


© Scott Twombly (YouTube channel)

- <u>Beispiel 2:</u> Kenntnis einer Lösung für das NP-vollständige Dreifarbenproblem (3COLOR) für einen Graphen G, d.h. 3COLOR(G)
  - P erzeugt einen zu G isomporphen Graphen G‘ = π(G) $\cong$ G und stellt diesen V zu
  - V fordert P heraus, entweder eine Lösung für 3COLOR(G‘) oder π zu liefern
  - P stellt V die entsprechende Antwort zu
  - Das wird solange wiederholt, bis V überzeugt ist, dass P die Lösung für 3COLOR(G) kennt

G





$\pi(a) = 1$
$\pi(b) = 6$
$\pi(c) = 8$
$\pi(d) = 3$
$\pi(g) = 5$
$\pi(h) = 2$
$\pi(i) = 4$
$\pi(j) = 7$

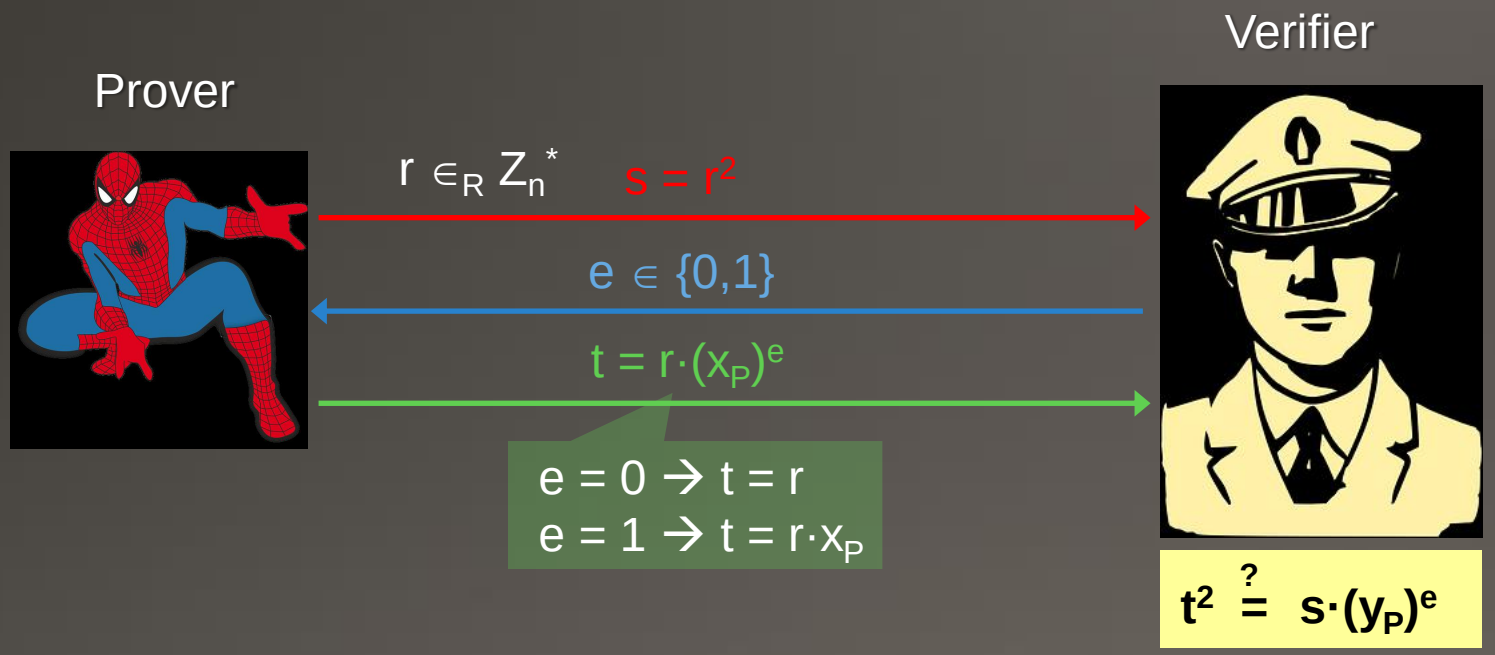
G‘ $\cong$ G

- 1987 haben Amos Fiat and Adi Shamir das erste Authentifikationsprotokoll mit der Zero-Knowledge-Eigenschaft vorgeschlagen

- Das Protokoll basiert auf der Einwegfunktion des modularen Quadrierens, d.h. es ist berechenbar unmöglich, modulare Quadratwurzeln zu berechnen (ohne die Faktorisierung des Moduls zu kennen)
- Seither sind viele andere Authentifikationsprotokoll mit der Zero-Knowledge-Eigenschaft vorgeschlagen worden (auch auf anderen Einwegfunktionen basierend und mit zum Teil anderen Eigenschaften)

- Voraussetzungen für das Fiat-Shamir-Protokoll
  - Modul n = pq ist öffentlich bekannt (p und q nicht)
  - P hat privaten Schlüssel x_P (zufällig ausgewähltes Element aus Z_n^*) und publiziert den öffentlichen Schlüssel (n,y_P) mit $y_P = x_P^2 \bmod n$
  - Man kann zeigen, dass die Faktorisierung von n und die Berechnung von Quadratwurzeln in Z_n^* berechenbar äquivalent sind
  - Das Protokoll wird in Runden ausgeführt
  - In jeder Runde beträgt die Betrugswahrscheinlichkeit ½
  - Nach k Runden beträgt die Betrugswahrscheinlichkeit $(½)^k$
  - Diese Wahrscheinlichkeit kann mit einem geeigneten k beliebig klein gemacht werden

- Runde im Fiat-Shamir-Protokoll



Alle Berechnungen und Gleichungen sind modulo n (mod n) zu verstehen

- Bemerkungen
  - Aus der Kenntnis von beiden Antworten für ein e (d.h. e = 0 und e = 1) und ein bestimmtes r kann x_P berechnet werden: $x_P = r \cdot x_P / r$
  - Daraus folgt, dass jemand, der x_P nicht kennt, für eine Runde nicht beide Antworten liefern kann
  - Ein Angreifer kann aber je Runde e erraten und sich auf diesen Fall vorbereiten
    - e = 0: Normaler Protokolldurchlauf (aber x_P fehlt für e = 1)
    - e = 1: Zufällige Wahl von t und Berechnung von $s = t^2 / y_P = t^2 \cdot y_P^{-1}$ (aber $r = s^{1/2}$ fehlt für e = 0)
  - Entsprechende Protokolltranskripte können simuliert werden (ohne x_P zu kennen)

- Beispiele für p = 3, q = 5, n = pq = 15, $x_p = 7$, $y_p = 7^2 \bmod 15 = 4$, $y_p^{-1} = 4$ (es gilt $4 \cdot 4 = 16 \equiv 1 \pmod{15}$)
  - Runde 1: e = 0 → r = 2, s = 4, t = 2 — $4 \equiv 4 \bmod 15$ — (4,0,2)
  - Runde 2: e = 1 → t = 3, $s = 3^2 \cdot 4 \bmod 15 = 6$ — $9 \equiv 6 \cdot 4 \bmod 15$ — (6,1,3)
  - Runde 3: e = 1 → t = 7, $s = 7^2 \cdot 4 \bmod 15 = 1$ — $49 \equiv 1 \cdot 4 \bmod 15$ — (1,1,7)
  - Runde 4: e = 0 → r = 8, s = 4, t = 8 — $64 \equiv 4 \bmod 15$ — (4,0,8)
  - …
- (4,0,2), (6,1,3), (1,1,7) und (4,0,8) sind korrekte Protokolltranskripte, die keine Information über $x_p = 7$ liefern können (weil sie ohne Kenntnis von x_p erzeugt worden sind)

- In der Theorie sind Zero-Knowledge-Protokolle vorteilhaft, weil sie beliebig oft durchlaufen werden können (mit frischen r bzw. s)
- Allerdings sind bestimmte MITM-Angriffe immer noch möglich (bei der Authentifikation vor Ort → Mafia Fraud oder Terrorist Fraud)

- Jede Form von «Token Binding» zerstört die Simulierbarkeit und damit auch die Zero-Knowledge-Eigenschaft, d.h. andere Lösungsansätze sind hier erforderlich (z.B. «Distance Bounding»-Protokolle)

- In der Praxis stellt die Interaktivität von Zero-Knowledge Verfahren das hauptsächliche Problem dar
- In bestimmten Situationen kann die Interaktion durch den Einsatz von kryptografischen Hashfunktionen ersetzt werden
  → Nicht-interaktive Zero-Knowledge-Beweisverfahren
- Aus dem nicht-interaktiven Fiat-Shamir-Protokoll resultiert z.B. ein Signatursystem (gilt auch für andere Protokolle)
- Nicht-interaktive Zero-Knowledge-Verfahren (z.B. BulletProof, zk-SNARK, zk-STARK, …) werden insbesondere im Umfeld von Krypto-währungen erforscht und eingesetzt (z.B. als Overlay-Netzwerk von Bitcoin)

6. Schlussfolgerungen und Ausblick

© David Wong, *Real-World Cryptography*, Manning Publications Co., 2021, Seite 230

- Das lang angekündige Ende des Passwortes zeichnet sich ab
- Nichtsdestotrotz gibt es immer noch sinnvolle Einsatzgebiete für Passwörter und enstprechende (a)PAKE-Protokolle (z.B. OPAQUE und ähnliche Protokolle)
- Im praktischen Einsatz wird sich FIDO2 durchsetzen und das Passwort verdrängen (zunächst bei Web-Anwendungen)
- FIDO2 bietet hohe Sicherheit und Benutzerfreundlichkeit / Bequemlichkeit
- Zero-Knowledge wird sich in anderen Bereichen auch durchsetzen (z.B. Kryptowährungen, E-Voting, …) und baut sich als Hype auf

thank you
danke
謝謝
ngiyabonga
teşekkür ederim
рахмат
Баярлалаа
spas
mersi
kia ora
barka
welalin
tack
mahalo
спасибо
faafetai lava
vinaka
blagodaram
dank je
misaotra
matondo
paldies
grazzi
tapadh leat
спасибі
kiitos
dankie
nanni
nandri
dhanyavad
gracias
хвала
bedankt
mauruuru
köszönöm
dankon
aĉiŭ
akun
asante
manana
tenki
obrigada
murakoze
enkosi
bayarlalaa
gràçie
hvala
djiere dieuf
tau
mochchakkeram
mamnun
дякую
chokrane
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
go raibh maith agat
taiku
sobodi
děkuji
obrigado
sagolun
sukriya
kop khun krap
arigatô
takk
dakujem
trugarez
mèsi
didi madloba
najis tuke
ありがとう
tanemirt
grazie
rahmet
shukriya
mercé
мерси
terima kasih
kam sah hamnida
rahmat
dhanyavadagalu
diolch
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
merci