LEHRERINFO ZUM VIDEO

Kryptografie und moderne IT-Sicherheit

FACH	STUFE	DAUER	VIDEO
Informatik	Gymnasium · Sek II	16 min	youtube.com/@Erklaerbaer

01 Lernziele

Nach dem Video können die Schülerinnen und Schüler …

- die Schutzziele (Vertraulichkeit, Integrität, Verfügbarkeit, Authentizität) zuordnen und das Kerckhoffs-Prinzip erklären,
- klassische Verfahren (Caesar, Vigenère) und ihre Schwächen (Häufigkeitsanalyse) erklären,
- symmetrische Verschlüsselung (XOR, One-Time-Pad, AES) und das Schlüsselaustauschproblem erklären,
- Diffie-Hellman und RSA mit kleinen Zahlen durchrechnen,
- Hashfunktionen, Passwort-Hashing mit Salt und digitale Signaturen erklären,
- TLS/HTTPS mit Zertifikaten, Ende-zu-Ende-Verschlüsselung und SQL-Injection einordnen.

02 Voraussetzungen

Caesar, HTTPS, Passwörter, Zwei-Faktor (Folge 13 · Grundlagen); Binärzahlen und XOR (Folge 2 und 4 · Advanced); Rest-Rechnung % (Folge 7). Für Kapitel 7: SQL (Folge 12 · Advanced).

03 Ablauf des Videos

ZEIT	KAPITEL	INHALT UND BEISPIEL
0:00	Einstieg	geheimer Schlüssel über ein offenes Netz · Ident · Fahrplan
0:35	Was du brauchst	Rückblick 13G, 02A, 04A, 07G
0:58	Schutzziele	vier Schutzziele am E-Banking · Alice, Bob, Eve · Kerckhoffs
1:50	Klassisch und geknackt	Vigenère mit KEY · Häufigkeitsanalyse · Enigma
2:42	Symmetrisch	XOR 0101 ⊕ 1100 · One-Time-Pad · AES · Schlüsselaustauschproblem

4:08	Diffie-Hellman	Farbmischung · p = 23, g = 5, a = 6, b = 15 → 2 · diskreter Logarithmus PAUSE-FRAGE a = 4, b = 3
5:55	RSA	Vorhängeschloss · n = 55, e = 3, d = 27 · 7 → 13 → 7 · Hybridverfahren PAUSE-FRAGE m = 2 · Merksatz
7:49	Hash und Signatur	SHA-256, Lawineneffekt · Downloads · Salt · bcrypt, Argon2 · Signatur · Merksatz
9:36	HTTPS, Messenger, Angriffe	Zertifikat, TLS-Handshake · Signal-Protokoll · MitM · SQL-Injection · Post-Quanten · Pentesting
11:21	Projekt-Baustein	Salt und Prepared Statements (16A) · kein `eval` im Rechner (15A)
11:47	Prüfungsfallen	fünf Fallen (siehe 04)
12:47	Zusammenfassung	Kernaussagen · zurück zum Einstieg
13:31	Quiz	4 Fragen · Endkarte Folge 14

04 Prüfungsfallen

- Hashen = Verschlüsseln – einen Hash kann man nicht zurückrechnen, eine Verschlüsselung mit dem Schlüssel schon.
- Signiert wird mit dem öffentlichen Schlüssel – signiert wird mit dem privaten, geprüft mit dem öffentlichen.
- Geheime Verfahren sind sicherer – nach Kerckhoffs sind offen geprüfte Verfahren sicherer.
- RSA verschlüsselt die ganze Webseite – asymmetrisch nur Schlüsselaustausch und Signatur; die Daten symmetrisch mit AES.
- HTTPS schützt vor SQL-Injection – HTTPS schützt den Transport; gegen Injection helfen Prepared Statements.

05 Einsatz im Unterricht

Farbmischung live: Diffie-Hellman mit Wasserfarben oder Lebensmittelfarbe nachspielen; die «Eve»-Gruppe sieht nur die Zwischenmischungen.

Rechnen: Aufgaben 3 und 4 mit dem Taschenrechner (mod-Taste) oder in Python mit `pow(g, a, p)`.

Lawineneffekt: In Python `hashlib.sha256("Erklärbär".encode()).hexdigest()` ausprobieren und ein Zeichen ändern.

Zertifikat ansehen: Im Browser auf das Schloss klicken, Aussteller und Gültigkeit einer bekannten Seite ablesen.

Hinweis: Alle Zahlenbeispiele sind nachgerechnet (DH: $5^6 \bmod 23 = 8$, $5^{15} \bmod 23 = 19$, gemeinsames Geheimnis 2; RSA: $7^3 \bmod 55 = 13$, $13^{27} \bmod 55 = 7$). Die Hash-Anfänge gelten für UTF-8-Kodierung. Die Salt-Beispiele zeigen das Prinzip mit SHA-256; in der Praxis nimmt man bcrypt, scrypt oder Argon2. NIST hat im August 2024 die ersten Post-Quanten-Standards veröffentlicht (FIPS 203, 204, 205). SQL-Injection wird nur so weit gezeigt, dass man sie erkennt und abwehrt; Übungen nur am eigenen Projekt (Folge 16 · Advanced). Rechtliches: siehe Folge 13 · Grundlagen.

06 Quellen

- W. Diffie, M. Hellman (1976): New Directions in Cryptography, IEEE Transactions on Information Theory 22(6)
- R. Rivest, A. Shamir, L. Adleman (1978): A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, CACM 21(2)
- NIST FIPS 197 (AES), FIPS 180-4 (SHA-2), FIPS 203/204/205 (Post-Quanten, 2024)
- IETF RFC 8446 (TLS 1.3) · RFC 9106 (Argon2) · OWASP Cheat Sheets: SQL Injection Prevention, Password Storage
- A. Kerckhoffs (1883): La cryptographie militaire · S. Singh: Geheime Botschaften, dtv

NAME _____ KLASSE _____ DATUM _____

Kryptografie und moderne IT-Sicherheit

1 Schutzziele ●○○

Ordne zu: a) Ein Webshop ist wegen einer Überlastung stundenlang nicht erreichbar. b) Jemand liest fremde Chats mit. c) Eine Rechnung wird unterwegs auf ein anderes Konto umgeschrieben. d) Eine Mail gibt sich als die Schulleitung aus.

2 XOR ●○○

Verschlüssle 1010 0110 mit dem Schlüssel 0110 1100 und **entschlüssle** das Ergebnis wieder.

3 Diffie-Hellman ●●○

p = 23, g = 5. Alice wählt a = 9, Bob wählt b = 7. **Berechne** A, B und das gemeinsame Geheimnis von beiden Seiten.

4 RSA ●●●

p = 3, q = 11. a) **Berechne** n und φ(n). b) e = 3: **Bestimme** d. c) **Verschlüssle** m = 4 und **entschlüssle** das Ergebnis wieder.

5 Passwörter speichern ●●○

Eine App speichert Passwörter als SHA-256-Hash ohne Salt. **Erkläre** zwei Probleme und wie man es besser macht.

6 Signatur

●●○

Beschreibe in Schritten, wie Bob eine von Alice signierte Datei prüft. Welche Schutzziele werden gesichert?

Lösungen

1 Schutzziele

a) Verfügbarkeit · b) Vertraulichkeit · c) Integrität · d) Authentizität.

2 XOR

1010 0110 ⊕ 0110 1100 = 1100 1010 · 1100 1010 ⊕ 0110 1100 = 1010 0110 (wieder der Klartext).

3 Diffie-Hellman

A = 5^9 mod 23 = 11 · B = 5^7 mod 23 = 17 · Alice: 17^9 mod 23 = 7 · Bob: 11^7 mod 23 = 7 → gemeinsames Geheimnis 7.

4 RSA

a) n = 33, φ(n) = 2 · 10 = 20 · b) d = 7, denn 3 · 7 = 21 = 20 + 1 · c) 4^3 = 64, 64 mod 33 = 31 · 31^7 mod 33 = 4 ✓.

5 Passwörter speichern

Gleiche Passwörter ergeben gleiche Hashes (man erkennt Mehrfachnutzung), und vorberechnete Tabellen knacken häufige Passwörter sofort; SHA-256 ist ausserdem sehr schnell, Brute Force wird leicht. Besser: zufälliger Salt pro Konto und ein langsames Verfahren wie bcrypt oder Argon2.

6 Signatur

1. Bob berechnet selbst den Hash der Datei. 2. Er entschlüsselt die Signatur mit Alices öffentlichem Schlüssel. 3. Stimmen beide Hashes überein, ist die Datei unverändert (Integrität) und stammt von Alice (Authentizität). Dass der öffentliche Schlüssel wirklich Alice gehört, bestätigt ein Zertifikat.

Q Quiz im Video

1 B (Integrität) · 2 A (privater Schlüssel) · 3 C (verschiedene Hashes bei gleichen Passwörtern) · 4 B (Faktorisieren grosser Zahlen ist schwer)