

IT-Sicherheit – Gesamt-Lernheft

Schutzziele, Kryptographie, Verschlüsselung, Hashwerte, Signaturen & Zertifikate

🧭 So benutzt du das Lernheft

- **Links** ist das Inhaltsverzeichnis. Der Punkt davor zeigt die IHK-Relevanz.
- Jeder Abschnitt hat ein **IHK-Etikett** und einen farbigen Kasten mit der Begründung. Rot = sehr wichtig, Orange/Gelb = mittel, Grau = gering.
- **Gelb unterstrichen** = Prüfungs-Schlagwort. Das solltest du erklären können.
- Knopf **„🎯 Nur IHK-wichtig“** blendet alles mit „gering“ aus. Gut zum Wiederholen kurz vor der Prüfung.
- Bei jedem Abschnitt kannst du **„gelernt“** abhaken. Der Fortschritt wird im Browser gespeichert.
- Suche oben (Taste `/`), mit Enter springst du zum nächsten Treffer. Lösungen sind eingeklappt – erst selbst überlegen!

sehr hoch hoch mittel bis hoch mittel gering bis mittel gering

Die IHK-Einstufungen sind eigene Einschätzungen aus den Einzel-Zusammenfassungen, keine amtlichen Angaben der IHK.

🔥 Top-Prüfungsthemen

sehr hoch 7 · hoch 5 · mittel bis hoch 1 · mittel 4 · gering bis mittel 1 · gering 2

- sehr hoch 🛡️ Block 4: Die Schutzziele – VIVA und CIA · 🔐 IT-Sicherheit 1 – Grundlagen
- sehr hoch 🔑 Block 3: Symmetrische Verschlüsselung · 🔐 IT-Sicherheit 3 – Verschlüsselung
- sehr hoch 🗝️ Block 4: Asymmetrische Verschlüsselung (Public Key) · 🔐 IT-Sicherheit 3 – Verschlüsselung
- sehr hoch 🤝 Block 5: Hybride Verfahren (SSL / TLS) · 🔐 IT-Sicherheit 3 – Verschlüsselung
- sehr hoch #️⃣ Block 1: Hashfunktionen · ✍️ IT-Sicherheit 4 – Signaturen & Zertifikate
- sehr hoch ✍️ Block 2: Digitale Signatur · ✍️ IT-Sicherheit 4 – Signaturen & Zertifikate
- sehr hoch 📜 Block 3: Digitale Zertifikate · ✍️ IT-Sicherheit 4 – Signaturen & Zertifikate
- hoch 📖 Block 2: Drei Begriffe, die oft verwechselt werden · 🔐 IT-Sicherheit 1 – Grundlagen
- hoch 📖 Block 1: Die Grundbegriffe · 🔐 IT-Sicherheit 2 – Einführung Kryptographie
- hoch 📖 Block 1: Kryptographie, Schlüssel und Algorithmus · 🔐 IT-Sicherheit 3 – Verschlüsselung
- hoch 📬 Block 6: Transport- vs. Ende-zu-Ende-Verschlüsselung · 🔐 IT-Sicherheit 3 – Verschlüsselung

- hoch 🏛️ Block 4: PKI – Public Key Infrastructure · ✍️ IT-Sicherheit 4 – Signaturen & Zertifikate

IHK	Abschnitt	Warum?
gering	🗺️ Block 1: Worum geht es im Fach? 🔐 IT-Sicherheit 1 – Grundlagen	Überblick
hoch	📖 Block 2: Drei Begriffe, die oft verwechselt werden 🔐 IT-Sicherheit 1 – Grundlagen	Begriffe werden gern abgefragt („Grenzen Sie ab …“)
mittel bis hoch	⚡ Block 3: Woher kommen Risiken? 🔐 IT-Sicherheit 1 – Grundlagen	oft als Einstieg in Aufgaben zu Schutzbedarf und Maßnahmen
sehr hoch	🛡️ Block 4: Die Schutzziele – VIVA und CIA 🔐 IT-Sicherheit 1 – Grundlagen	kommt in fast jeder Prüfung zum Thema Sicherheit
hoch	📖 Block 1: Die Grundbegriffe 🔐 IT-Sicherheit 2 – Einführung Kryptographie	die Begriffe musst du sicher benutzen können
mittel	🔄 Block 2: ROT-13 – Buchstaben ersetzen (Substitution) 🔐 IT-Sicherheit 2 – Einführung Kryptographie	als Beispiel für ein unsicheres Verfahren
gering bis mittel	🔀 Block 3: Transposition – Buchstaben umsortieren 🔐 IT-Sicherheit 2 – Einführung Kryptographie	
gering	🖼️ Block 4: Steganographie – Nachrichten verstecken 🔐 IT-Sicherheit 2 – Einführung Kryptographie	gut als Abgrenzung zur Kryptographie
hoch	📖 Block 1: Kryptographie, Schlüssel und Algorithmus 🔐 IT-Sicherheit 3 – Verschlüsselung	

IHK	Abschnitt	Warum?
mittel	🧱 Block 2: Block- und Stromverschlüsselung 🔐 IT-Sicherheit 3 – Verschlüsselung	
sehr hoch	🔑 Block 3: Symmetrische Verschlüsselung 🔐 IT-Sicherheit 3 – Verschlüsselung	
sehr hoch	🗝️ Block 4: Asymmetrische Verschlüsselung (Public Key) 🔐 IT-Sicherheit 3 – Verschlüsselung	
sehr hoch	🤝 Block 5: Hybride Verfahren (SSL / TLS) 🔐 IT-Sicherheit 3 – Verschlüsselung	„Erklären Sie, wie HTTPS funktioniert“ ist ein Klassiker
hoch	📬 Block 6: Transport- vs. Ende-zu-Ende-Verschlüsselung 🔐 IT-Sicherheit 3 – Verschlüsselung	
mittel	⏩ Block 7: Perfect Forward Secrecy und Diffie-Hellman 🔐 IT-Sicherheit 3 – Verschlüsselung	PFS als Begriff, Diffie-Hellman als Prinzip
sehr hoch	#️⃣ Block 1: Hashfunktionen ✍️ IT-Sicherheit 4 – Signaturen & Zertifikate	
sehr hoch	✍️ Block 2: Digitale Signatur ✍️ IT-Sicherheit 4 – Signaturen & Zertifikate	
sehr hoch	📜 Block 3: Digitale Zertifikate ✍️ IT-Sicherheit 4 – Signaturen & Zertifikate	HTTPS und Zertifikate sind AP2-Klassiker
hoch	🏛️ Block 4: PKI – Public Key Infrastructure ✍️ IT-Sicherheit 4 – Signaturen & Zertifikate	die Abkürzungen RA, CA und VA werden gern abgefragt

IHK	Abschnitt	Warum?
mittel	⚖️ Block 5: Elektronische Signaturen im Recht (eIDAS) ✍️ IT-Sicherheit 4 – Signaturen & Zertifikate	die drei Stufen kommen vor, vor allem die QES

🔐 IT-Sicherheit 1 – Grundlagen

📘 Grundlage: Folien „IT-Sicherheit: Einführung“ (GBS Fachschule, 9 Folien). Die Zusammenfassung folgt der Reihenfolge der Folien.

- **Fett** gedruckte Begriffe sind Prüfungs-Schlagwörter. Die solltest du erklären können.
- 🎯 Jeder Abschnitt hat eine Zeile „IHK-Relevanz“. Das ist eine eigene Einschätzung, keine amtliche Angabe.
- IT-Sicherheit passt zu diesen Prüfungsteilen (AO 2020):
  - **AP1** („Einrichten eines IT-gestützten Arbeitsplatzes“): Schutzbedarf, Schutzziele, Datenschutz, einfache Schutzmaßnahmen.
  - **AP2 Konzeption und Administration** und **Netzwerke**: Sicherheitskonzepte, Verschlüsselung, Backup, Zugriffsschutz.
  - **WiSo**: Datenschutz (DSGVO) am Rand.
- ✏️ Am Ende jedes Blocks stehen Übungsfragen. Die Lösungen stehen gesammelt am Ende.
- 🔎 „Ergänzung“ = steht nicht auf den Folien, aus Allgemeinwissen ergänzt.

Symbol	Bedeutung
🎯	IHK-Relevanz
💡	Merkhilfe
⚠️	Typischer Fehler / Falle
🔎	Ergänzung (nicht auf den Folien)

🗺️ Block 1: Worum geht es im Fach? IHK gering

> 🎯 **IHK: gering** Überblick

Der Themenplan des Fachs:

Thema	Frage dahinter	Schutzziel
Verschlüsselung / Kryptographie	Bleibt Vertrauliches vertraulich?	Vertraulichkeit
Signaturen und Zertifikate	Wurden Daten verändert? Ist die Quelle echt?	Integrität, Authentizität
Backup, USV usw.	Sind Daten und Systeme da, wenn man sie braucht?	Verfügbarkeit

Thema	Frage dahinter	Schutzziel
Zugangskontrolle	Kommt kein Unbefugter in Räume oder an Daten?	Vertraulichkeit
Angriffe & Schutzmaßnahmen	Welche Angriffe gibt es, wie schützt man sich?	alle
Datenschutz	Was verlangt die DSGVO?	– (Gesetz)
Sonstiges	Cloud, Tracking, E-Mail, Social Media, Blockchain	–

💡 Fast jedes Thema gehört zu einem Schutzziel (Block 4). Das hilft beim Einordnen.

📖 Block 2: Drei Begriffe, die oft verwechselt werden IHK hoch

> 🎯 **IHK: hoch** Begriffe werden gern abgefragt („Grenzen Sie ab …“)

Begriff	Was wird geschützt?	Pflicht?
Datensicherheit	**alle** Daten – vor Verlust, Manipulation, Ausspähen	Ermessen des Einzelnen / Unternehmens
Datenschutz	nur **personenbezogene** Daten	**Gesetz**: DSGVO, „Recht auf informationelle Selbstbestimmung“
Datensicherung	Kopie wichtiger Daten (**Backup**)	eine **Maßnahme** für Datensicherheit

💡 Merksatz: **Datenschutz schützt Menschen, Datensicherheit schützt Daten, Datensicherung ist das Backup.**

⚠️ Falle: Datensicherung ist **kein** Oberbegriff. Sie ist nur **eine** Maßnahme, um Datensicherheit zu erreichen.

🔎 Ergänzung: **Personenbezogene Daten** = alle Angaben, mit denen man eine Person erkennen kann. Beispiele: Name, Adresse, E-Mail, IP-Adresse, Personalnummer.

✏️ Übungsfragen Block 2

Ü2.1 Ein Unternehmen sichert jede Nacht seinen Fileserver auf Band. Ist das Datenschutz, Datensicherheit oder Datensicherung?

Ü2.2 Warum ist eine Kundenliste ein Fall für den Datenschutz, eine Konstruktionszeichnung aber nicht?

⚡ Block 3: Woher kommen Risiken? IHK mittel bis hoch

🎯 **IHK: mittel bis hoch** oft als Einstieg in Aufgaben zu Schutzbedarf und Maßnahmen



Risiken mit und ohne menschliches Eingreifen

Risiken mit und ohne menschliches Eingreifen

Ohne menschliches Eingreifen (niemand hat Schuld): defekte Hardware, Software-Fehler (**Bugs**), Stromausfall, Feuer, Wasser, Naturkatastrophen.

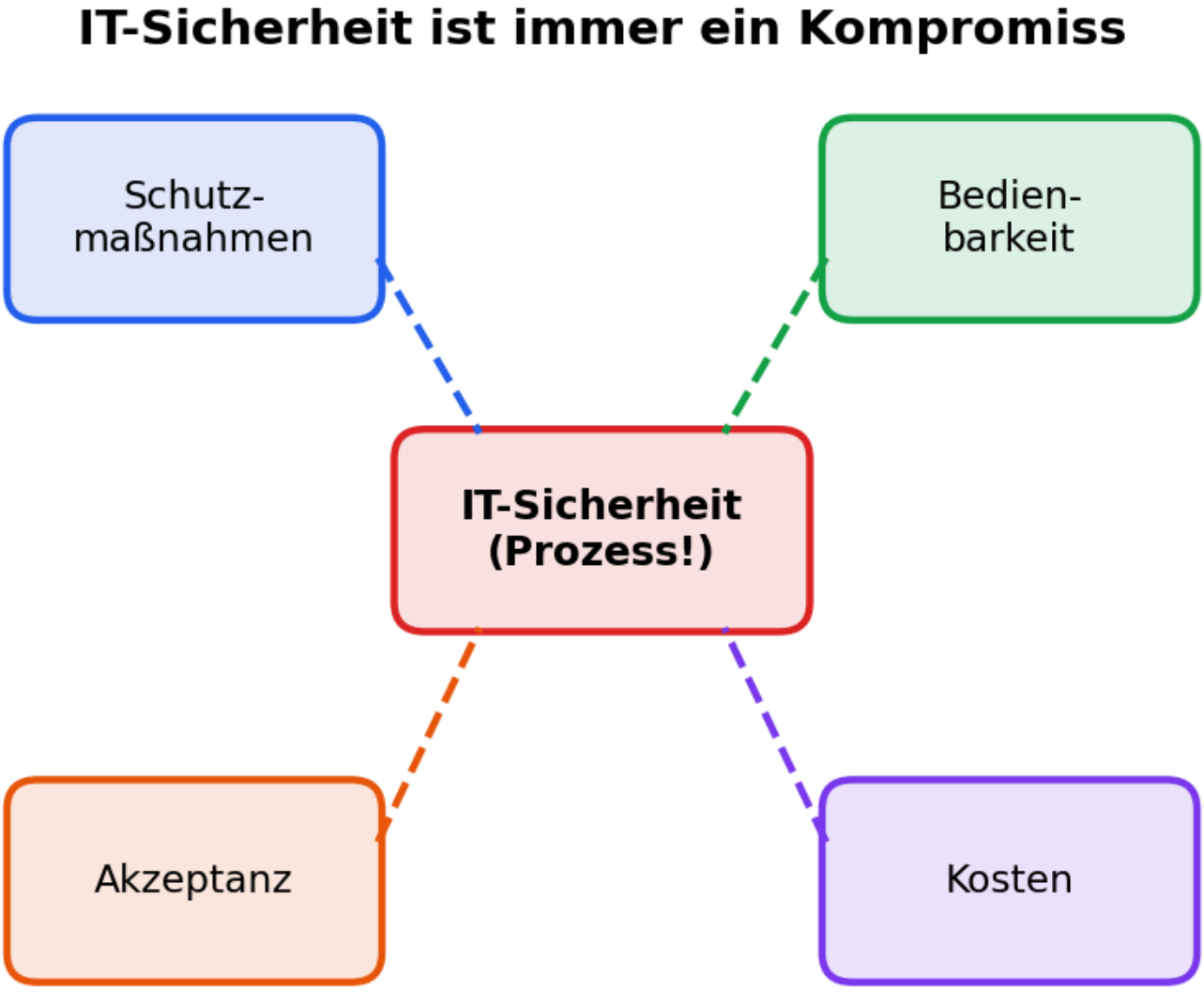
Mit menschlichem Eingreifen – drei Arten:

1. **Unachtsamkeit / Fahrlässigkeit**: falsche Eingaben, Passwort am Monitor, Stecker ziehen, versehentlich löschen, Mail-Anhang öffnen.
2. **Gezielte Angriffe**: Angriffe auf Server, **Schadsoftware**, Diebstahl, Sabotage.
3. **Fehler in der Organisation**: kein Notfallplan, keine Kontrolle, niemand ist zuständig.

💡 Der Mensch ist die größte Schwachstelle. Darum gehören **Schulungen** immer zu den Schutzmaßnahmen.

🔄 Sicherheit ist kein Zustand

- IT-Sicherheit ist ein **Prozess**. Man ist nie „fertig“. Neue Lücken und Angriffe kommen ständig dazu.
- IT-Sicherheit ist immer ein **Kompromiss** zwischen vier Dingen:



Kompromiss

Kompromiss

Beispiel: Ein 30-stelliges Passwort, das jede Woche wechselt, ist sehr sicher. Es ist aber unbequem (Bedienbarkeit). Mitarbeiter schreiben es auf einen Zettel (Akzeptanz). Am Ende ist es **unsicherer**.

🔎 Ergänzung: Der Prozess-Gedanke steckt auch im **PDCA-Zyklus** (Plan – Do – Check – Act) und im **BSI IT-Grundschutz**.

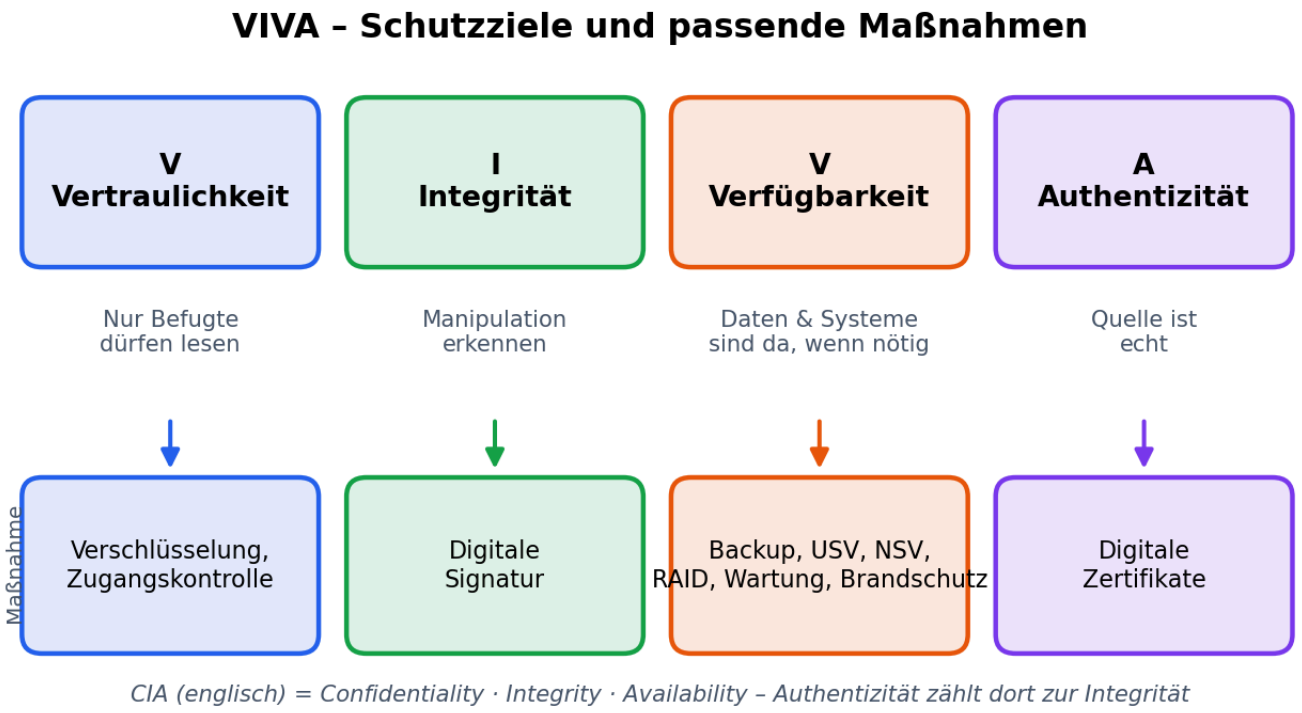
✏️ Übungsfragen Block 3

Ü3.1 Nennen Sie je zwei Risiken mit und ohne menschliches Eingreifen.

Ü3.2 Warum ist „Wir haben 2024 eine Firewall gekauft, also sind wir sicher“ falsch?

🛡️ Block 4: Die Schutzziele – VIVA und CIA IHK sehr hoch

> 🎯 **IHK: sehr hoch** kommt in fast jeder Prüfung zum Thema Sicherheit



VIVA – Schutzziele und Maßnahmen

VIVA – Schutzziele und Maßnahmen

Schutzziel	Bedeutung	Typische Maßnahme
Vertraulichkeit	Nur Befugte dürfen Daten sehen.	**Verschlüsselung**, Zugangs- und Zugriffskontrolle
Integrität	Daten sind nicht verändert. Manipulation kann man nicht immer verhindern, aber **erkennen**.	**Digitale Signatur** (🔎 auch Hashwerte)
Verfügbarkeit	Daten und Systeme sind da, wenn man sie braucht.	**Backup**, **USV**, NSV, **RAID**, Wartung, Brandschutz
Authentizität	Die Quelle ist echt und überprüfbar.	**Digitale Zertifikate**

Weitere Schutzziele: **Nichtabstreitbarkeit** (Verbindlichkeit), Verlässlichkeit, Anonymität.

CIA – die englische Variante

Englisch	Deutsch
Confidentiality	Vertraulichkeit
Integrity	Integrität
Availability	Verfügbarkeit

Bei **CIA** fehlt die Authentizität. Grund: Signaturen und Zertifikate funktionieren technisch fast gleich. Authentizität gilt hier als **Sonderfall der Integrität**. Manche deutschen Texte nennen sie dann „erweitertes Schutzziel“.

💡 Eselsbrücke: **VIVA** – „Vertraulich, Intakt, Verfügbar, Authentisch“.

🔎 Ergänzung: **USV** = unterbrechungsfreie Stromversorgung (Akku, überbrückt Minuten). **NSV** = Notstromversorgung, meist eine **Netzersatzanlage** (NEA, z. B. Diesel-Aggregat), für längere Ausfälle (Stunden bis Tage).

⚠️ Falle: **RAID ist kein Backup!** RAID schützt vor einem Plattendefekt (Verfügbarkeit). Löscht jemand eine Datei oder kommt Ransomware, ist sie auf allen Platten weg.

✏️ Übungsfragen Block 4

Ü4.1 Ordnen Sie zu: (a) Serverraum mit Chipkarte, (b) tägliches Backup, (c) E-Mail mit Signatur, (d) HTTPS-Zertifikat einer Bank.

Ü4.2 Ein Hacker ändert in einer Datenbank Kontonummern. Welches Schutzziel ist verletzt?

Ü4.3 Wofür steht CIA? Warum fehlt dort die Authentizität?

Ü4.4 Der Chef sagt: „Wir haben RAID 1, ein Backup brauchen wir nicht.“ Was antworten Sie?

✅ Lösungen

Ü2.1 Datensicherung (Backup). Sie ist eine Maßnahme für die Datensicherheit (Verfügbarkeit).

Ü2.2 Die Kundenliste enthält **personenbezogene** Daten (Namen, Adressen) → DSGVO gilt. Die Zeichnung enthält keine Personendaten → Datensicherheit, aber kein Datenschutz.

Ü3.1 Ohne Mensch: Stromausfall, Festplattendefekt, Wasserschaden, Bug. Mit Mensch: versehentliches Löschen, Phishing-Anhang öffnen, Hackerangriff, Diebstahl.

Ü3.2 Sicherheit ist ein Prozess. Neue Lücken und Angriffe entstehen ständig. Die Firewall muss gepflegt, aktualisiert und geprüft werden. Andere Risiken (Mensch, Stromausfall) deckt sie gar nicht ab.

Ü4.1 (a) Vertraulichkeit (Zugangskontrolle) · (b) Verfügbarkeit · (c) Integrität (+ Authentizität) · (d) Authentizität.

Ü4.2 Integrität.

Ü4.3 Confidentiality, Integrity, Availability. Authentizität wird als Sonderfall der Integrität gesehen, weil Signaturen und Zertifikate technisch fast gleich arbeiten.

Ü4.4 RAID schützt nur vor Plattenausfall. Gegen Löschen, Ransomware, Brand oder Diebstahl hilft nur ein Backup, am besten getrennt vom Server gelagert (🔎 3-2-1-Regel: 3 Kopien, 2 Medien, 1 außer Haus).

🔐 IT-Sicherheit 2 – Einführung Kryptographie

📘 Grundlage: Folien „Entschlüsseln Sie das mal …“ (GBS Fachschule, 13 Folien).

- **Fett** = Prüfungs-Schlagwort. 🎯 = eigene Einschätzung der IHK-Relevanz.
- Die alten Verfahren (ROT-13, Transposition) kommen selten direkt in der Prüfung vor. Sie erklären aber die **Begriffe**, die später wichtig sind: Klartext, Chiffrat, Algorithmus, Schlüssel.
- ✏️ Übungsfragen am Ende jedes Blocks, Lösungen ganz am Ende. 🔎 = Ergänzung.

📖 Block 1: Die Grundbegriffe IHK hoch

> 🎯 **IHK: hoch** die Begriffe musst du sicher benutzen können

Begriff	Bedeutung	Beispiel
Klartext	die lesbare Nachricht	„Wir verwenden Cookies …“
Chiffrat (= Geheimtext, Kryptogramm, Ciphertext)	die verschlüsselte Nachricht	„Jve irejraqra Pbbxvrf …“
Verschlüsseln	Klartext → Chiffrat	
Entschlüsseln	Chiffrat → Klartext	

🕵️ Wie knackt man ein Chiffrat? – „Educated Guesses“

Nicht wild raten, sondern **begründet vermuten**. Das spart Zeit, Geld und Rechenleistung.

1. **Was ist es?** → vermutlich ein Text.
2. **Welche Sprache?** → vermutlich Deutsch.
3. **Welches Verfahren?** → vermutlich einfache Buchstabenvertauschung.

Dann nach **Merkmalen** suchen:

- **Häufigkeit**: Der häufigste Buchstabe im Deutschen ist **e**.
- **Muster**: typische Paare wie „ch“, „sch“, „st“, „nd“.
- **Kurze Wörter**: 2–3 Buchstaben → um, zu, in, der, die, das, ein, und.
- Je **länger** das Chiffrat, desto besser funktioniert das.

💡 Genau das nennt man **Häufigkeitsanalyse**. Sie knackt jede einfache Buchstabenvertauschung.

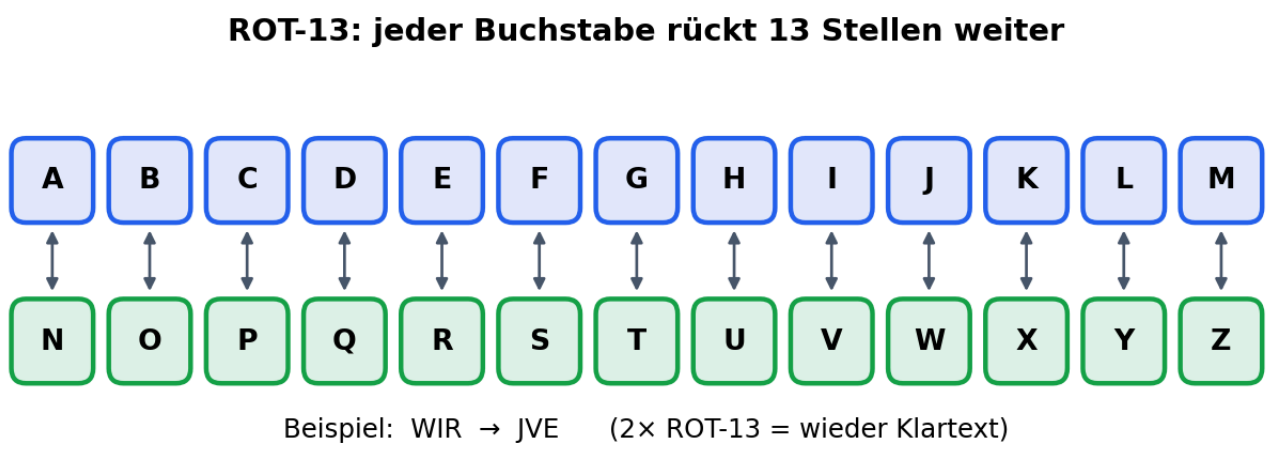
✏️ Übungsfragen Block 1

Ü1.1 Nennen Sie drei andere Wörter für „Chiffrat“.

Ü1.2 Warum ist ein langes Chiffrat leichter zu knacken als ein kurzes?

🔄 Block 2: ROT-13 – Buchstaben ersetzen (Substitution) IHK mittel

> 🎯 **IHK: mittel** als Beispiel für ein unsicheres Verfahren



ROT-13

ROT-13

- Jeder Buchstabe wird durch den Buchstaben **13 Stellen weiter** im Alphabet ersetzt.
- Fachbegriff: **monoalphabetische Substitution** (ein Buchstabe wird immer durch denselben ersetzt).
- Verwandt: **Cäsar-Chiffre** (beliebige Verschiebung, z. B. um 3).
- Besonderheit: Das Alphabet hat 26 Buchstaben. 13 + 13 = 26. Darum gilt: **zweimal ROT-13 = wieder Klartext**.

Beispiel: W → J, I → V, R → E. Also **WIR → JVE**.

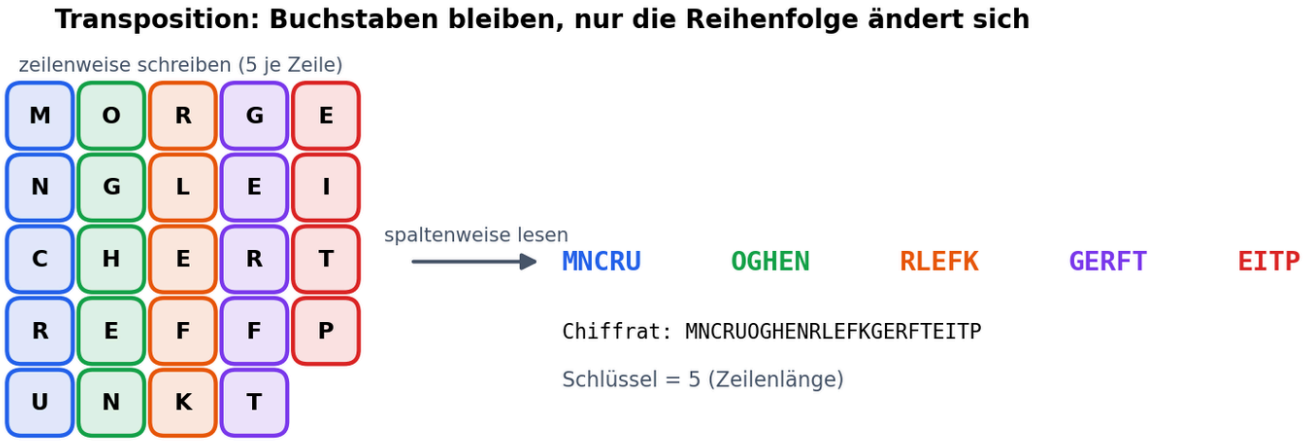
✏️ Übungsfragen Block 2

Ü2.1 Verschlüsseln Sie „IHK“ mit ROT-13.

Ü2.2 Warum knackt eine Häufigkeitsanalyse ROT-13 sofort?

🔀 Block 3: Transposition – Buchstaben umsortieren IHK gering bis mittel

> 🎯 **IHK: gering bis mittel**

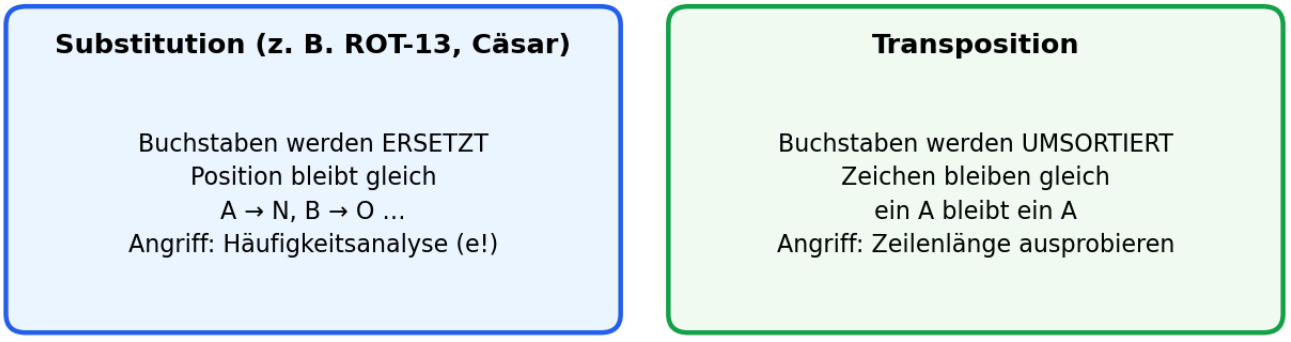


Transposition

Transposition

- Die Zeichen **bleiben gleich** (ein A bleibt ein A). Nur die **Reihenfolge** ändert sich.
- Vorbereitung (**Normieren**): alles groß schreiben, ä → ae, Leer- und Satzzeichen weg.
- Text in Zeilen fester Länge schreiben (hier 5), dann **spaltenweise** ablesen.
- Kennt man die Zeilenlänge (= den **Schlüssel**), kann man zurückrechnen. Leer- und Satzzeichen ergänzt man dann leicht.

Beispiel aus dem Unterricht: „Morgen, gleicher Treffpunkt“ → **MNCRUOGHENRLEFKGERFTEITP**



Substitution vs. Transposition

Substitution vs. Transposition

✏️ Übungsfragen Block 3

Ü3.1 Was ist der Unterschied zwischen Substitution und Transposition?

Ü3.2 Entschlüsseln Sie **HLAOL** (Zeilenlänge 3, der Klartext hat 5 Buchstaben).

🖼️ Block 4: Steganographie – Nachrichten verstecken IHK gering

> 🎯 **IHK: gering** gut als Abgrenzung zur Kryptographie

- Von griechisch „steganós“ + „gráphein“ = **verstecktes Schreiben**.
- Die Nachricht wird **in anderen Daten versteckt** (dem **Container**), meist Bilder oder Musik.
- Die Änderung liegt **unter der Wahrnehmungsschwelle**. Niemand sieht oder hört sie.
- Der Container wirkt völlig harmlos. Man merkt nicht einmal, **dass** etwas versteckt ist.

	Kryptographie	Steganographie
Ziel	Inhalt **unlesbar** machen	Existenz der Nachricht **verbergen**
Angreifer sieht	„Da ist etwas Geheimes“	„Nur ein Urlaubsgruß“

Beispiel aus dem Unterricht („Liebe Kolleginnen! Wir genießen nun endlich …“): Man zählt die Buchstaben jedes Wortes. **Gerade Länge = 1, ungerade = 0**. Je 8 Wörter ergeben ein Byte:

Binär	Dezimal	ASCII
01010011	83	**S**
01001111	79	**O**
01010011	83	**S**

→ Versteckte Botschaft: **SOS**

⚠️ Auf der Folie steht die Spalte „Hex“, die Werte 83 / 79 sind aber **Dezimal**zahlen. Hexadezimal wären es 53 / 4F.

💡 Beides lässt sich kombinieren: erst verschlüsseln, dann verstecken.

✏️ Übungsfragen Block 4

Ü4.1 Worin unterscheiden sich Kryptographie und Steganographie?

Ü4.2 Welches Zeichen ergibt die Bitfolge 01000001 (ASCII)?

✅ Lösungen

Ü1.1 Geheimtext, Kryptogramm, Ciphertext.

Ü1.2 Mehr Text = mehr Buchstaben. Dann treten Häufigkeiten und Muster deutlicher hervor.

Ü2.1 I → V, H → U, K → X. Also **VUX**.

Ü2.2 Jeder Buchstabe wird immer gleich ersetzt. Die Häufigkeiten bleiben erhalten. Der häufigste Chiffrat-Buchstabe ist vermutlich „e“ – damit ist die Verschiebung bekannt.

Ü3.1 Substitution ersetzt Zeichen (Position bleibt). Transposition vertauscht Positionen (Zeichen bleiben).

Ü3.2 5 Buchstaben in Zeilen zu 3 → Zeile 1 hat 3, Zeile 2 hat 2 Buchstaben. Die Spalten sind also 2 + 2 + 1 Zeichen lang: **HL | AO | L**. Zeilenweise lesen: H A L / L O → **HALLO**.

Ü4.1 Kryptographie macht den Inhalt unlesbar, man sieht aber, dass etwas verschlüsselt ist. Steganographie versteckt die Nachricht so, dass niemand merkt, dass es sie gibt.

Ü4.2 01000001 = 65 = **A**.

🔐 IT-Sicherheit 3 – Verschlüsselung

📘 Grundlage: Folien „IT-Sicherheit: Verschlüsselung“ (GBS Fachschule, 34 Folien).

- **Fett** = Prüfungs-Schlagwort. 🎯 = eigene Einschätzung der IHK-Relevanz.
- Dieses Kapitel ist **Kernstoff** für die Prüfung. Symmetrisch, asymmetrisch, hybrid und Ende-zu-Ende kommen in **AP2 (Konzeption und Administration, Netzwerke)** regelmäßig vor, Grundbegriffe auch in **AP1**.
- ✏️ Übungsfragen am Ende jedes Blocks, Lösungen ganz am Ende. 🔎 = Ergänzung (nicht auf den Folien). ⚠️ = Falle oder Hinweis zu einer Folie.

📖 Block 1: Kryptographie, Schlüssel und Algorithmus IHK hoch

> **🎯 IHK: hoch**

Kryptographie = „geheimes Schreiben“ (kryptos = geheim, graphein = schreiben). Daten werden mit einem **Schlüssel** nach den Regeln eines **Algorithmus** so umgewandelt, dass nur Berechtigte sie zurückverwandeln können.

	Was ist das?	Bei ROT-13	Bei der Transposition
Algorithmus	die **Vorschrift**: wie werden Schlüssel und Klartext verknüpft?	„Ersetze jeden Buchstaben durch den n-ten Nachfolger“	„Normiere den Text, nimm jeden n-ten Buchstaben“
Schlüssel	die **geheime Information** (digital: eine Folge aus 0 und 1)	n = 13	n = 5

🏛️ Kerckhoffs’ Prinzip (1883)

Bei ROT-13 hängt die Sicherheit am **Verfahren**. Wer es kennt, probiert die 25 möglichen Schlüssel einfach durch. Wird es geknackt, muss das ganze Verfahren getauscht werden.

Auguste **Kerckhoffs** forderte dagegen:

> Das System darf **keine Geheimhaltung** verlangen und muss ohne Schaden **in die Hände des Feindes fallen** können.

Was ist …	… öffentlich?	… geheim?
nach Kerckhoffs	der **Algorithmus**	nur der **Schlüssel**

Warum ist das besser?

- Einen Schlüssel geheim zu halten ist leichter als ein ganzes Verfahren.
- Ein Schlüssel ist schnell getauscht – ein Verfahren nicht.
- Wird ein Schlüssel bekannt, bleibt das System bestehen. Nur der Schlüssel wird ersetzt.
- Öffentliche Verfahren werden von vielen Fachleuten geprüft (**Peer Review**).
- Es gibt keine Möglichkeit für geheime Hintertüren (**Backdoors**).

💡 Merksatz: **„Security by Obscurity“ (Sicherheit durch Geheimhaltung des Verfahrens) ist schlecht.** Gute Verfahren wie AES sind öffentlich.

📜 Historische Beispiele

Vigenère-Chiffre (Trithemius 1508, Vigenère 1585): eine Cäsar-Chiffre mit **wechselnder** Verschiebung. Der Schlüssel ist ein Wort.

Klartext	**B**	**U**	**N**	**D**	**E**	**S**	**W**	**E**	**H**	**R**
Schlüssel	H	A	U	S	H	A	U	S	H	A
Verschiebung	7	0	20	18	7	0	20	18	7	0
Chiffrat	**I**	**U**	**H**	**V**	**L**	**S**	**Q**	**W**	**O**	**R**

Gleiche Klartext-Buchstaben werden verschieden verschlüsselt (beide E → L bzw. W). Darum hilft die einfache Häufigkeitsanalyse nicht mehr. Fachbegriff: **polyalphabetische Substitution**. Auch die **Enigma** arbeitete so.

✏️ Übungsfragen Block 1

Ü1.1 Was besagt Kerckhoffs’ Prinzip? Nennen Sie zwei Gründe dafür.

Ü1.2 Verschlüsseln Sie „IT“ mit Vigenère und dem Schlüssel „BB“.

🧱 Block 2: Block- und Stromverschlüsselung

IHK mittel

> 🎯 **IHK: mittel**

	Blockverschlüsselung	Stromverschlüsselung
Arbeitsweise	feste **Blöcke** (z. B. 128 Bit) werden mit dem Schlüssel verknüpft	ein **laufender Datenstrom** wird Bit für Bit verknüpft
Schlüssel	typisch 64 bis 256 Bit	ein **Schlüsselstrom** (pseudozufällig, meist aus einem Schlüssel abgeleitet)
Verknüpfung	je nach Verfahren	meist **XOR**
Beispiel 🔎	AES, DES	ChaCha20, RC4 (veraltet)

- Je **kürzer** der Schlüssel, desto leichter das Knacken durch Ausprobieren (**Brute Force**).
- **256-Bit-Schlüssel** gelten als **praktisch** nicht knackbar = nicht in angemessener Zeit mit angemessenen Mitteln.

⚠️ Die Folie nennt die Vigenère-Chiffre als Beispiel für Blockverschlüsselung. Das ist eine Vereinfachung: Sie verschlüsselt zwar in „Blöcken“ der Schlüssellänge, ist aber eigentlich eine polyalphabetische Substitution. Für die Prüfung: **AES** ist das Standardbeispiel für eine Blockchiffre.

🔎 Ergänzung XOR: gleich → 0, verschieden → 1. Zweimal XOR mit demselben Schlüssel ergibt wieder den Klartext: 1010 XOR 0110 = 1100, dann 1100 XOR 0110 = 1010.

✏️ Übungsfragen Block 2

Ü2.1 Berechnen Sie 1100 XOR 1010.

Ü2.2 Was bedeutet „praktisch nicht knackbar“?

🔑 Block 3: Symmetrische Verschlüsselung IHK sehr hoch

🎯 IHK: sehr hoch

Symmetrisch: EIN gemeinsamer Schlüssel

Problem: K muss vorher SICHER ausgetauscht werden (Shared Secret)



Symmetrische Verschlüsselung

Symmetrische Verschlüsselung

Zum Ver- und Entschlüsseln wird **derselbe Schlüssel** benutzt („gemeinsames Geheimnis“, **Shared Secret**, **Pre-Shared Key**).

➕ Vorteile	➖ Nachteile
einfach (Anwendung, Verständnis, Programmierung)	Der Schlüssel muss **sicher ausgetauscht** werden (**Schlüsselaustauschproblem**)
schnell (einfache Bit-Operationen, die jede CPU sehr schnell kann)	🔎 viele Schlüssel bei vielen Teilnehmern (siehe unten)

Verfahren

Verfahren	Block	Schlüssel	Status
DES (IBM, 1970er)	64 Bit	**56 Bit**	**unsicher** – 1998 in 56 Std., 1999 in rund 22 Std. per Brute Force geknackt
3DES (Triple DES)	64 Bit	2 × 56 Bit (🔎 oder 3 × 56)	🔎 veraltet, soll nicht mehr eingesetzt werden
AES	**128 Bit**	**128, 192 oder 256 Bit**	**Standard heute**, AES-256 gilt als praktisch nicht knackbar

⚠️ Hinweis zu 3DES: Die Folie zeigt „Key 1 → Key 2 → Key 1“. Genauer läuft es so: mit Key 1 **verschlüsseln**, mit Key 2 **entschlüsseln**, mit Key 1 wieder **verschlüsseln** (**EDE**-Verfahren).

🔎 **Wie viele Schlüssel braucht man?** (klassische Prüfungsfrage)

- Symmetrisch braucht **jedes Paar** einen eigenen Schlüssel: **n · (n − 1) / 2**. Bei 10 Personen: 10 · 9 / 2 = **45** Schlüssel.

- Asymmetrisch hat **jede Person** ein Schlüsselpaar: **2 · n**. Bei 10 Personen: **20** Schlüssel (10 öffentliche + 10 private).

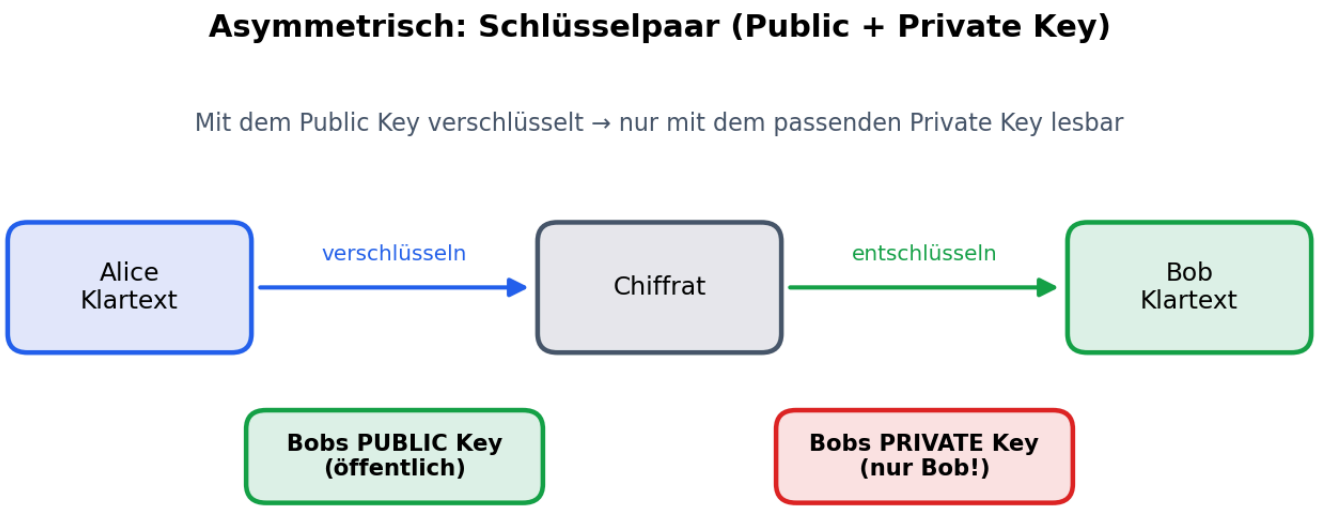
✏️ Übungsfragen Block 3

Ü3.1 Warum gilt DES heute als unsicher?

Ü3.2 Ein Team mit 6 Personen will paarweise symmetrisch verschlüsselt kommunizieren. Wie viele Schlüssel sind nötig?

🗝️ Block 4: Asymmetrische Verschlüsselung (Public Key) IHK sehr hoch

🎯 IHK: sehr hoch



Asymmetrische Verschlüsselung

Asymmetrische Verschlüsselung

- Jeder Teilnehmer hat ein **Schlüsselpaar**:
  - **Public Key** (öffentlicher Schlüssel) – darf jeder haben.
  - **Private Key** (privater Schlüssel) – bleibt **geheim**.
- Was mit dem einen Schlüssel verschlüsselt wurde, kann **nur der andere** entschlüsseln.

Ablauf (Alice schreibt an Bob):

1. Alice verschlüsselt mit **Bobs Public Key**.
2. Sie schickt das Chiffrat über einen unsicheren Kanal (z. B. das Internet).
3. Nur Bob kann es mit **seinem Private Key** entschlüsseln.

💡 Merksatz: **Verschlüsseln mit dem öffentlichen Schlüssel des Empfängers – entschlüsseln mit dem eigenen privaten.**

🔎 Ausblick (nächstes Thema): Bei der **digitalen Signatur** ist es umgekehrt. Man signiert mit dem **eigenen Private Key**, jeder prüft mit dem Public Key.

➕ Vorteile	➖ Nachteile
kein gemeinsames Geheimnis nötig → kein Schlüsselaustauschproblem	**sehr rechenintensiv** = langsam
	ungeeignet für große Datenmengen
	kompliziert (Verständnis, Anwendung, Programmierung)

Verfahren

Verfahren	Grundlage	Bemerkung
RSA (Rivest, Shamir, Adleman)	Multiplikation sehr großer Primzahlen – 🔎 sicher, weil man das Produkt kaum wieder in die Primzahlen **zerlegen** kann	bekanntestes, weit verbreitetes Verfahren
ECC (Elliptic Curve Cryptography)	elliptische Kurven	neuer, **schneller und kürzere Schlüssel** als RSA

✏️ Übungsfragen Block 4

Ü4.1 Bob will Alice eine vertrauliche Nachricht schicken. Mit welchem Schlüssel verschlüsselt er?

Ü4.2 Nennen Sie je einen Vor- und Nachteil von symmetrischer und asymmetrischer Verschlüsselung.

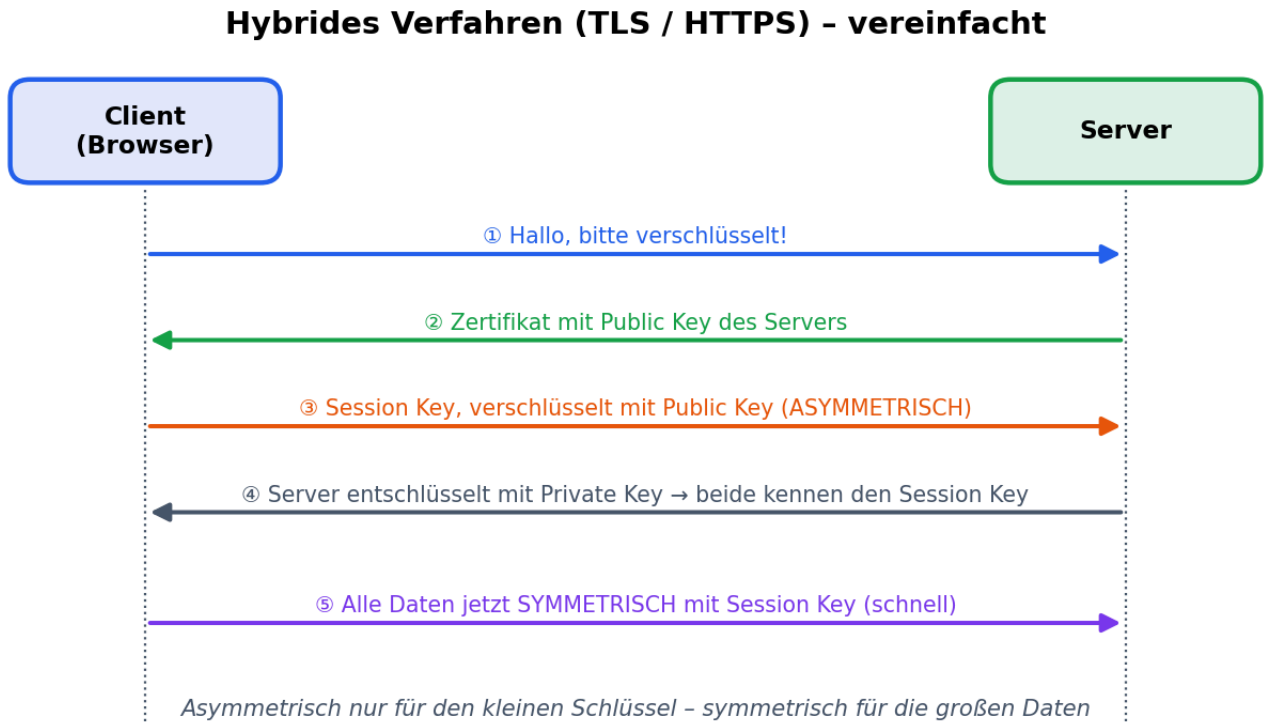
🤝 Block 5: Hybride Verfahren (SSL / TLS) IHK sehr hoch

🎯 **IHK: sehr hoch** „Erklären Sie, wie HTTPS funktioniert“ ist ein Klassiker

	Symmetrisch	Asymmetrisch
Geschwindigkeit	➕ schnell	➖ langsam
Einfachheit	➕ einfach	➖ kompliziert
Schlüsselaustausch	➖ gemeinsames Geheimnis nötig	➕ kein gemeinsames Geheimnis

Idee: das Beste aus beiden kombinieren.

- Den kleinen **Session Key** (z. B. 256 Bit) tauscht man **asymmetrisch** aus.
- Die eigentlichen Daten verschlüsselt man **symmetrisch** mit diesem Session Key.



Hybrides Verfahren

Hybrides Verfahren

Ablauf (wie auf der Folie):

1. Der Client fordert eine verschlüsselte Verbindung an.
2. Der Server schickt seinen **Public Key** in einem **Zertifikat**.
3. Der Client erzeugt einen zufälligen **Session Key**, verschlüsselt ihn mit dem Public Key des Servers und schickt ihn.
4. Der Server entschlüsselt ihn mit seinem **Private Key**.
5. Jetzt haben beide das gemeinsame Geheimnis. Ab hier geht alles **symmetrisch** weiter.

Protokolle: **SSL** (Secure Sockets Layer, veraltet) → heute **TLS** (Transport Layer Security). Einsatz z. B. bei **HTTPS**.

🔎 Ergänzung: Der Ablauf auf der Folie ist das klassische Verfahren mit RSA. Das aktuelle **TLS 1.3** verschickt den Session Key gar nicht mehr. Beide Seiten **berechnen** ihn mit **Diffie-Hellman** (Block 7). Das Zertifikat dient dann nur noch zum Nachweis, dass der Server echt ist.

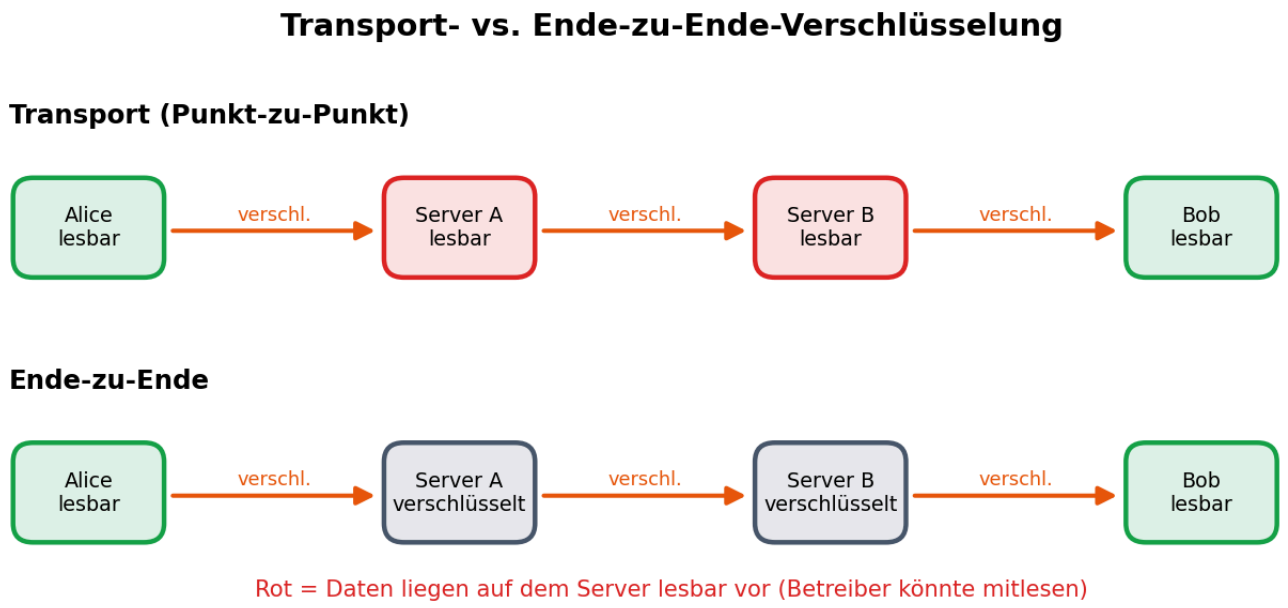
✏️ Übungsfragen Block 5

Ü5.1 Warum verschlüsselt man bei HTTPS nicht einfach alle Daten asymmetrisch?

Ü5.2 Wofür schickt der Server sein Zertifikat?

📬 Block 6: Transport- vs. Ende-zu-Ende-Verschlüsselung IHK hoch

🎯 IHK: hoch



Transport vs. Ende-zu-Ende

Transport vs. Ende-zu-Ende

	Transportverschlüsselung (Punkt-zu-Punkt)	Ende-zu-Ende-Verschlüsselung (E2E)
Verschlüsselt	nur **auf dem Weg** zwischen zwei Stationen	**auf dem ganzen Weg**
Lesbar	beim Sender, beim Empfänger **und auf jedem Server dazwischen**	**nur** beim Sender und beim Empfänger

	Transportverschlüsselung (Punkt-zu-Punkt)	Ende-zu-Ende-Verschlüsselung (E2E)
Beispiel 🔎	E-Mail über TLS (STARTTLS), HTTPS	PGP / S/MIME bei E-Mail, Signal, WhatsApp

💡 Merksatz: Bei Transportverschlüsselung kann der **Mail-Anbieter mitlesen**, bei Ende-zu-Ende nicht.

✏️ Übungsfragen Block 6

Ü6.1 Eine Firma schickt Gehaltsabrechnungen per E-Mail über einen externen Mail-Anbieter. Welche Verschlüsselung empfehlen Sie? Begründen Sie.

⏩ Block 7: Perfect Forward Secrecy und Diffie-Hellman IHK mittel

> 🎯 **IHK: mittel** PFS als Begriff, Diffie-Hellman als Prinzip

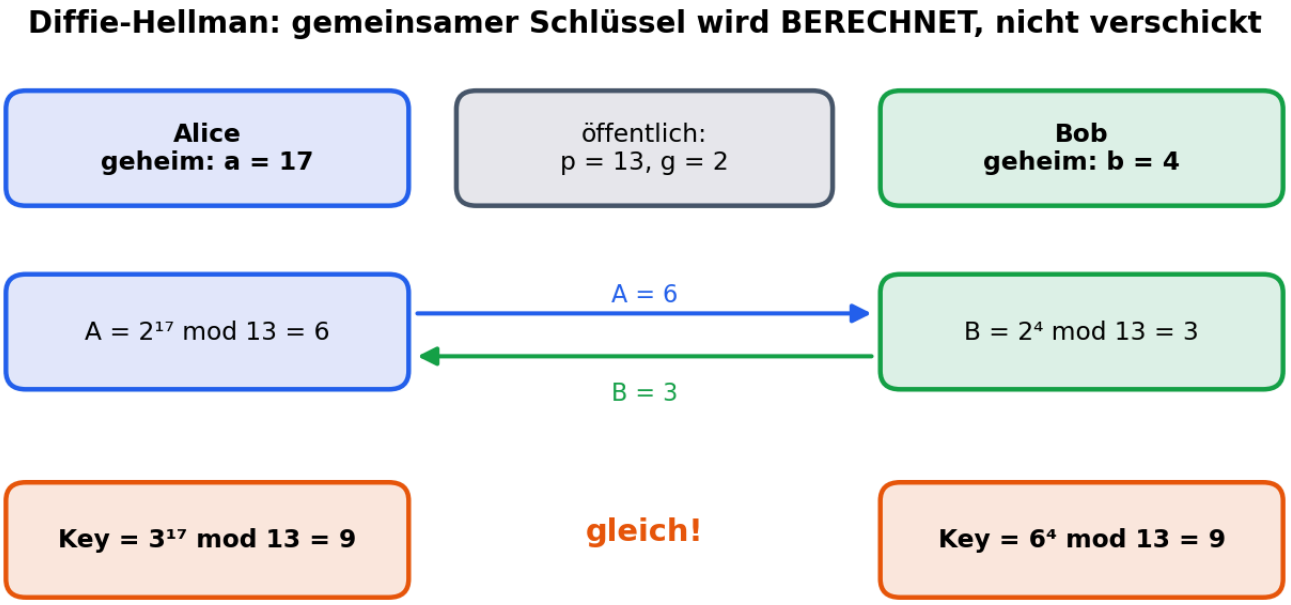
Das Problem beim klassischen Hybrid-Verfahren

1. Ein Angreifer **zeichnet alles auf**: den verschlüsselten Session Key und die ganze Kommunikation.
2. Zunächst kann er nichts damit anfangen.
3. Bekommt er **später** den Private Key des Servers (Diebstahl, Hack), entschlüsselt er den Session Key – und damit **nachträglich alles**.

Perfect Forward Secrecy (PFS) = Auch wenn der Private Key später bekannt wird, bleiben **alte Sitzungen** sicher. Der Session Key lässt sich nicht rekonstruieren („**Folgenlosigkeit**“).

Diffie-Hellman-Schlüsselvereinbarung (1976)

Whitfield **Diffie** und Martin **Hellman** (mit Ralph Merkle): Das gemeinsame Geheimnis wird **nicht verschickt, sondern von beiden berechnet**.



Ein Lauscher sieht nur p, g, A, B – ohne a oder b kann er 9 nicht ausrechnen

Diffie-Hellman

Diffie-Hellman

Schritt	Alice	öffentlich	Bob
1. Vereinbaren		Primzahl **p = 13**, Basis **g = 2**	
2. Geheimzahl wählen	a = 17		b = 4
3. Rechnen	A = g^a mod p = 2^17 mod 13 = **6**		B = g^b mod p = 2^4 mod 13 = **3**
4. Austauschen	schickt A = 6 →	A und B	← schickt B = 3
5. Schlüssel	B^a mod p = 3^17 mod 13 = **9**		A^b mod p = 6^4 mod 13 = **9**

Probe: 2^4 = 16, 16 mod 13 = **3** ✔ · 6^4 = 1296, 1296 − 99 · 13 = 1296 − 1287 = **9** ✔

- Ein Lauscher sieht nur p, g, A und B. Ohne a oder b kann er den Schlüssel praktisch nicht berechnen (bei echten, riesigen Zahlen).
- Werden a und b für **jede Sitzung neu** gewählt und danach gelöscht, entsteht **PFS**. 🔎 Fachbegriff: **DHE / ECDHE** (das E am Ende steht für „ephemeral“ = flüchtig).

🔎 **mod** = Rest bei der Division. 17 mod 5 = 2, weil 17 = 3 · 5 + **2**.

⚠️ Auf der Folie heißt g „Wurzel“. Gemeint ist eine **Primitivwurzel** (auch: Generator) modulo p.

✏️ Übungsfragen Block 7

Ü7.1 Was bedeutet Perfect Forward Secrecy?

Ü7.2 Rechnen Sie Diffie-Hellman mit p = 11, g = 2, a = 3, b = 4. Welcher gemeinsame Schlüssel entsteht?

📋 Spickzettel

Begriff	In einem Satz
Kerckhoffs	Algorithmus öffentlich, nur der Schlüssel geheim
Symmetrisch	1 Schlüssel für alles – schnell, aber Schlüsselaustausch-Problem
Asymmetrisch	Public + Private Key – kein Austausch nötig, aber langsam
Hybrid (TLS)	asymmetrisch den Session Key, symmetrisch die Daten
AES	symmetrischer Standard, 128/192/256 Bit
RSA / ECC	asymmetrische Standards
E2E	nur Sender und Empfänger können lesen
PFS	späterer Schlüsselklau entschlüsselt alte Sitzungen nicht
Diffie-Hellman	Schlüssel wird berechnet, nicht verschickt

✅ Lösungen

Ü1.1 Die Sicherheit darf nur von der Geheimhaltung des **Schlüssels** abhängen, nicht vom Verfahren. Gründe: Schlüssel sind leichter geheim zu halten und schneller zu tauschen. Das Verfahren kann öffentlich geprüft werden. Keine versteckten Hintertüren.

Ü1.2 B = Verschiebung 1. I + 1 = **J**, T + 1 = **U** → **JU**.

Ü2.1 1100 XOR 1010 = **0110**.

Ü2.2 Theoretisch geht es (alle Schlüssel ausprobieren), aber nicht in angemessener Zeit und mit angemessenen Mitteln – z. B. würde es Milliarden Jahre dauern.

Ü3.1 Der Schlüssel ist mit 56 Bit zu kurz. Er wurde schon 1998 in 56 Stunden und 1999 in rund 22 Stunden per Brute Force geknackt.

Ü3.2 $n \cdot (n - 1) / 2 = 6 \cdot 5 / 2 =$ **15** Schlüssel.

Ü4.1 Mit **Alices Public Key**. Nur Alice kann dann mit ihrem Private Key entschlüsseln.

Ü4.2 Symmetrisch: + schnell / − Schlüsselaustausch nötig. Asymmetrisch: + kein gemeinsames Geheimnis / − langsam.

Ü5.1 Asymmetrische Verfahren sind sehr langsam und für große Datenmengen ungeeignet. Darum wird nur der kleine Session Key asymmetrisch übertragen.

Ü5.2 Das Zertifikat enthält den **Public Key** des Servers. 🔎 Es bestätigt außerdem, dass der Server **echt** ist (Authentizität).

Ü6.1 **Ende-zu-Ende** (z. B. S/MIME oder PGP). Bei reiner Transportverschlüsselung liegen die Mails beim Anbieter lesbar vor. Gehaltsdaten sind personenbezogen und vertraulich (DSGVO).

Ü7.1 Wird der Private Key später gestohlen, kann man trotzdem keine früher aufgezeichneten Sitzungen entschlüsseln.

Ü7.2 $A = 2^3 \bmod 11 = 8 \cdot B = 2^4 \bmod 11 = 16 \bmod 11 = 5$. Alice: $5^3 \bmod 11 = 125 \bmod 11 = 125 - 121 =$ **4**. Bob: $8^4 \bmod 11 = 4096 \bmod 11 = 4096 - 372 \cdot 11 = 4096 - 4092 =$ **4**. Gemeinsamer Schlüssel: **4** ✔

✍️ IT-Sicherheit 4 – Signaturen & Zertifikate

📘 Grundlage: Folien „IT-Sicherheit: Signaturen & Zertifikate“ (GBS Fachschule, 30 Folien).

- **Fett** = Prüfungs-Schlagwort. 🎯 = eigene Einschätzung der IHK-Relevanz.
- Dieses Kapitel ist **Kernstoff**. Hashwerte, Signaturen und Zertifikate kommen in **AP1** (Schutzziele, Integrität) und in **AP2** (HTTPS, Zertifikate, PKI, E-Mail-Signatur) regelmäßig vor.
- ✏️ Übungsfragen am Ende jedes Blocks, Lösungen ganz am Ende. 🔎 = Ergänzung (nicht auf den Folien). ⚠️ = Falle oder Hinweis zu einer Folie.

#️⃣ Block 1: Hashfunktionen IHK sehr hoch

> 🎯 **IHK: sehr hoch**

Eine **Hashfunktion** bildet **beliebig große** Daten auf einen Wert **fester Länge** ab.

- Das Ergebnis heißt **Hashwert**, **Prüfsumme** oder **Message Digest**.
- Er ist der **„digitale Fingerabdruck“** einer Datei.

Prinzip-Beispiel der Folie (Quersumme bis zur Einstelligkeit):

Eingabe	Rechnung	Hash
132	1 + 3 + 2	**6**
56	5 + 6 = 11 → 1 + 1	**2**
8202	8 + 2 + 0 + 2 = 12 → 1 + 2	**3**
52572	5 + 2 + 5 + 7 + 2 = 21 → 2 + 1	**3**

⚠️ Das ist **keine brauchbare** Hashfunktion. Sie zeigt nur das Prinzip. 8202 und 52572 ergeben beide 3, das ist schon eine **Kollision**.

Anforderungen an eine kryptographische Hashfunktion

Eigenschaft	Bedeutung	Quersummen-Beispiel
Einwegfunktion	vom Hashwert kommt man **nicht** zurück zur Eingabe	✔ erfüllt

Eigenschaft	Bedeutung	Quersummen-Beispiel
Kollisionsresistenz	zwei verschiedene Eingaben ergeben **praktisch nie** denselben Hash	✘ nur 10 mögliche Werte (0–9) → spätestens der 11. Wert kollidiert
Lawineneffekt	winzige Änderung der Eingabe → **komplett anderer** Hash	✘
🔎 **feste Länge**	egal wie groß die Eingabe ist	✔
🔎 **deterministisch**	gleiche Eingabe → immer gleicher Hash	✔

Warum es Kollisionen immer geben muss: Es gibt unendlich viele mögliche Eingaben, aber nur endlich viele Hashwerte. Die Lösung ist, **sehr viele** Hashwerte zu ermöglichen. Mit 256 Bit gibt es **$2^{256} \approx 1{,}16 \cdot 10^{77}$** Möglichkeiten, also eine Zahl mit 78 Stellen. Eine zufällige Kollision ist damit praktisch ausgeschlossen.

💡 Merksatz: **Eine Hashfunktion gilt als sicher, solange niemand eine Kollision gezielt erzeugen kann.**

Verfahren

Verfahren	Länge	Status
MD5	128 Bit	❌ **unsicher**, Kollisionen nachgewiesen
SHA-1	160 Bit	❌ **unsicher**, Kollisionen nachgewiesen (🔎 2017 „SHAttered“)
SHA-256 / SHA-512 (SHA-2)	256 / 512 Bit	✅ **Standard**, z. B. bei Zertifikaten und Downloads
🔎 **SHA-3**	224–512 Bit	✅ neuer Standard, anderes Bauprinzip
RIPEMD-160, Whirlpool	160 / 512 Bit	selten

🔎 Für **Passwörter** nimmt man **langsame** Verfahren wie **bcrypt**, **Argon2** oder **yescrypt** (unter Debian `$y$` in `/etc/shadow`). SHA-256 ist dafür zu schnell, ein Angreifer könnte Milliarden Versuche pro Sekunde rechnen.

Notation und Lawineneffekt

Hashwerte schreibt man **hexadezimal**. 4 Bit ergeben 1 Hex-Ziffer:

`1111 1010 1011 0110 1101 0010 1011 1110` → `f a b 6 d 2 b e` → **fab6d2be**

Eingabe	SHA-1
Hallo Welt	`28cbbc72d6a52617a7abbfff6756d04bbad0106a`
Hallo Welt**!**	`726c3e8861ab0652a5043ea5faff6d3ef33fb209`
Hallo und guten Tag! Wie geht‘s?	`aedf6fce7d244cedb9e1a377aff3dd4da7e80585`

- Ein einziges „!“ ändert den **ganzen** Hash. Das ist der **Lawineneffekt**.
- Ein langer Text ergibt einen **gleich langen** Hash: SHA-1 hat immer 40 Hex-Ziffern = 160 Bit.
- 🔎 Werte nachgeprüft. Achtung: Mit einem Zeilenumbruch am Ende (z. B. `echo "Hallo Welt" | sha1sum`) kommt ein anderer Wert heraus. Ohne Umbruch geht es mit `printf 'Hallo Welt' | sha1sum`.

Einsatz

Einsatz	Wie?
Datenübertragung / Downloads prüfen	Hash an Quelle und Ziel bilden und vergleichen. Gleich → Übertragung in Ordnung. 🔎 `sha256sum debian.iso`, unter Windows `Get-FileHash`
Dubletten finden	Hashes vergleichen ist viel schneller als ganze Dateien
Passwörter speichern	nur der Hash wird gespeichert. Beim Login wird die Eingabe gehasht und verglichen. 🔎 mit **Salt**

🔎 **Salt** = zufälliger, **nicht geheimer** Zusatzwert pro Benutzer, der vor dem Hashen an das Passwort gehängt wird. Gleiche Passwörter ergeben so verschiedene Hashes. **Rainbow Tables** (vorberechnete Hash-Tabellen) werden nutzlos, und jedes Konto muss einzeln geknackt werden. Gegen schwache Passwörter hilft Salt aber **nicht**.

⚠️ Hashen ist keine Verschlüsselung

	Verschlüsselung	Hashfunktion
Schlüssel	ja	**nein**
umkehrbar	ja (mit Schlüssel)	**nein** (Einwegfunktion)
Ausgabelänge	wächst mit der Eingabe	**immer gleich**
Schutzziel	**Vertraulichkeit**	**Integrität**

✏️ Übungsfragen Block 1

Ü1.1 Nennen Sie zwei Anforderungen an eine kryptographische Hashfunktion und erklären Sie sie kurz.

Ü1.2 Ein Kollege sagt: „Unsere Passwörter sind sicher, die sind ja mit MD5 verschlüsselt.“ Nehmen Sie Stellung (zwei Fehler).

Ü1.3 Warum kann keine Hashfunktion Kollisionen vollständig ausschließen?

✍️ Block 2: Digitale Signatur IHK sehr hoch

> **🎯 IHK: sehr hoch**

Manipulationen (absichtlich, versehentlich oder durch Übertragungsfehler) lassen sich **nicht verhindern**. Eine digitale Signatur macht sie aber **erkennbar**.

Schutzziele: **Integrität** (Daten unverändert) und 🔎 **Authentizität** (stammt wirklich vom Absender). 🔎 Daraus folgt auch **Nichtabstreitbarkeit**: Der Absender kann später nicht leugnen, signiert zu haben.

Erster Versuch: Datei + Hash mitschicken

1. Der Absender bildet den Hash der Datei.
2. Er schickt Datei, Hash und den Namen der Hashfunktion.
3. Der Empfänger hasht die Datei selbst und vergleicht.

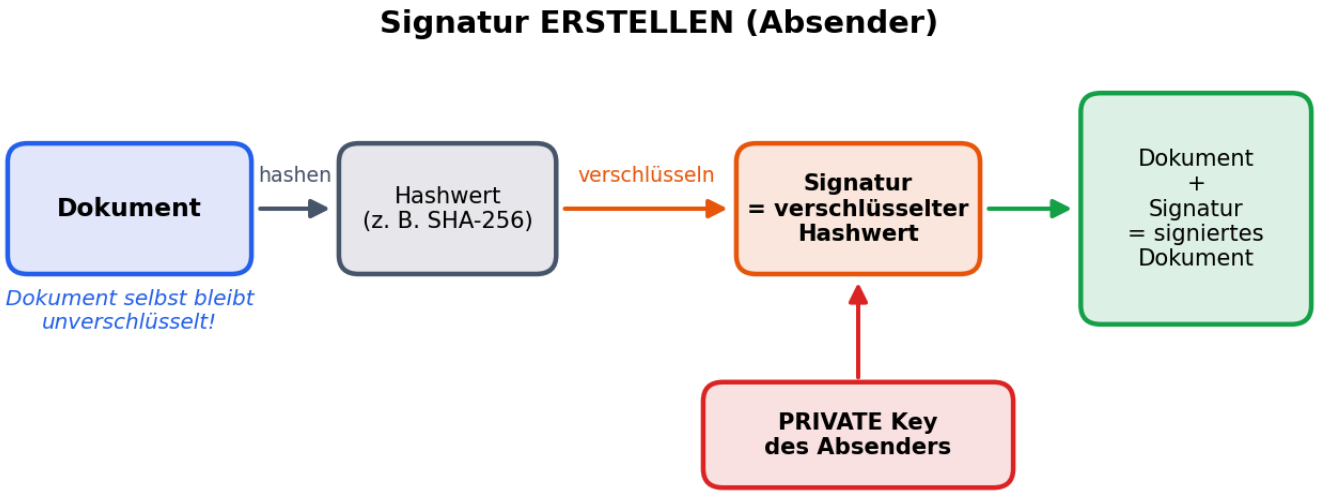
Der Haken: Ein Angreifer fängt alles ab, ändert die Datei, **berechnet einfach einen neuen Hash** und schickt beides weiter. Der Empfänger merkt nichts. → Der **Original-Hash muss geschützt werden**. Dafür braucht man die **asymmetrische Kryptographie**.

Asymmetrie „umgedreht“

Verschlüsseln mit …	Entschlüsseln mit …	Zweck
Public Key des Empfängers	Private Key des Empfängers	**Vertraulichkeit** (Verschlüsselung, ITS 3)
Private Key des Absenders	Public Key des Absenders	**Signatur**: jeder kann prüfen, nur der Absender konnte es erzeugen

💡 Merksatz: **Verschlüsseln → Public Key des Empfängers. Signieren → eigener Private Key.**

Signatur erstellen



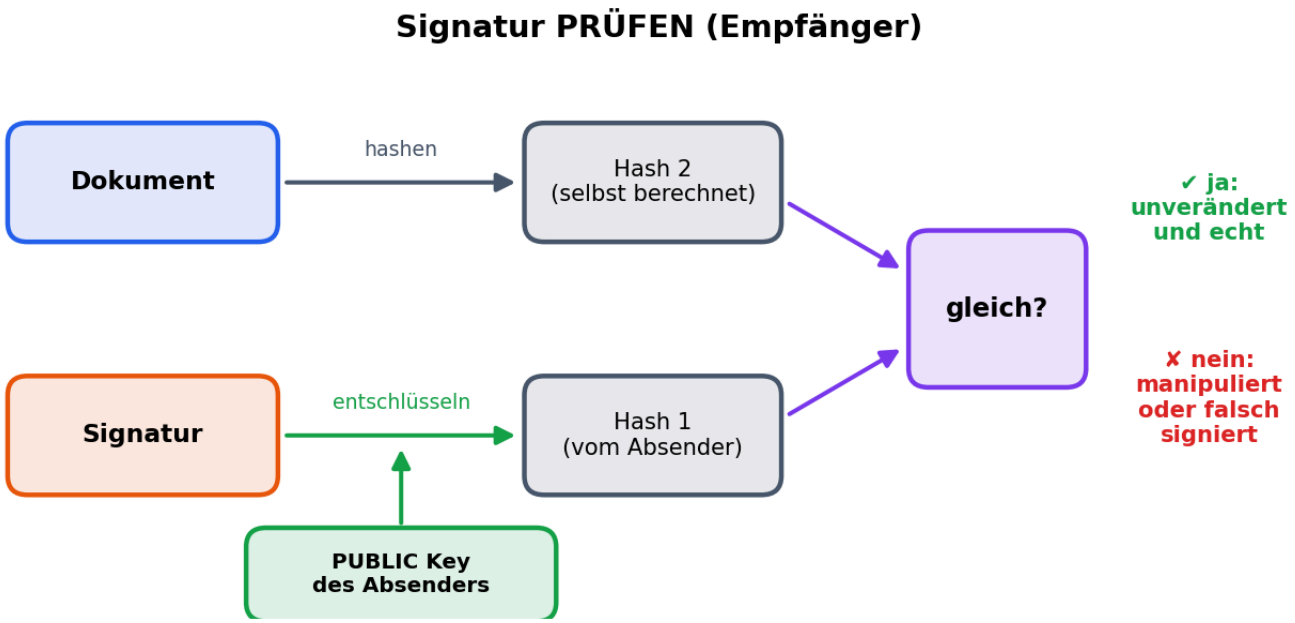
Signatur erstellen

Signatur erstellen

1. Den **Hash** des Dokuments bilden.
2. Den Hash mit dem **eigenen Private Key** verschlüsseln. Das Ergebnis ist die **Signatur**.
3. Dokument und Signatur zusammen verschicken. Das Dokument selbst bleibt **unverschlüsselt**.

Begriff der Folie	Bedeutung
Signaturfunktion	die benutzte Hashfunktion
Signaturschlüssel	der Private Key des Signierenden
Signatur	der verschlüsselte Hashwert

Signatur prüfen



Signatur prüfen

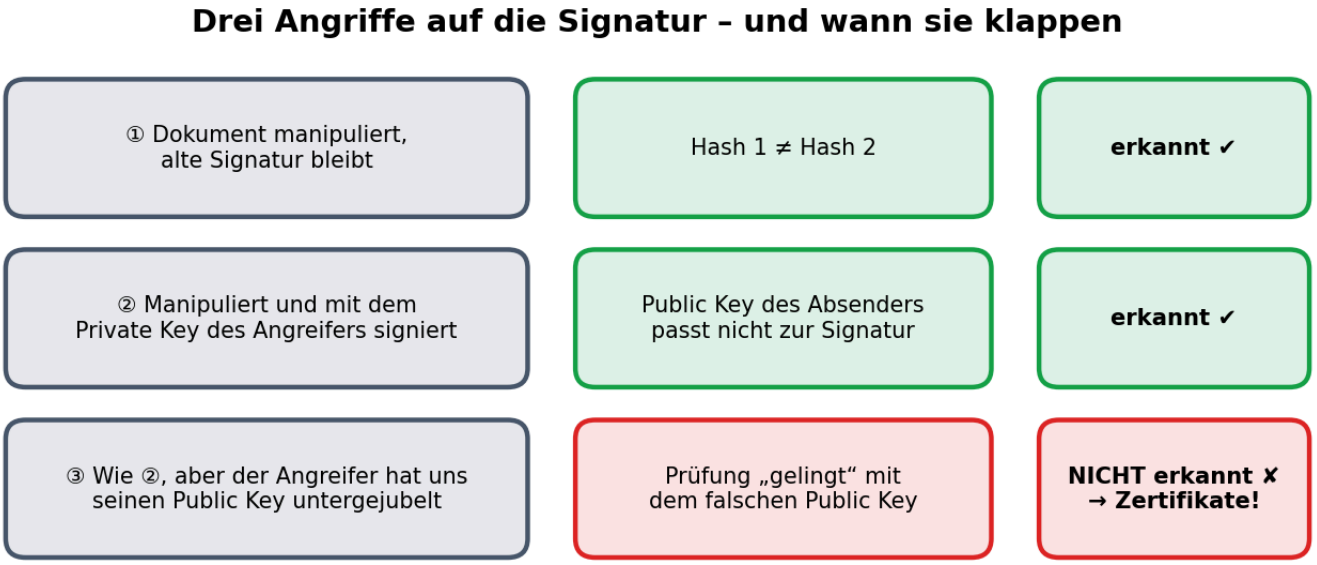
Signatur prüfen

1. Die Signatur mit dem **Public Key des Absenders** entschlüsseln → **Hash 1** (Original).
2. Das empfangene Dokument selbst hashen → **Hash 2**.
3. Vergleichen: **gleich** → Dokument unverändert und vom Absender. **Ungleich** → manipuliert.

⚠️ Eine Signatur macht das Dokument **nicht vertraulich**. Jeder kann es lesen. Wer beides will, signiert **und** verschlüsselt (🔎 z. B. S/MIME oder PGP bei E-Mail).

🔎 Das „Verschlüsseln mit dem Private Key“ ist das Schulmodell und passt genau für **RSA**. Andere Verfahren wie **ECDSA** signieren mathematisch anders. Das Prinzip „mit Private Key signieren, mit Public Key prüfen“ gilt aber überall.

Angriffe



Angriffe

Angriffe

Angriff	Ergebnis
① Dokument geändert, alte Signatur bleibt	Hash 1 ≠ Hash 2 → **erkannt**
② geändert und mit dem **Private Key des Angreifers** signiert	Mit dem echten Public Key des Absenders lässt sich die Signatur nicht sinnvoll entschlüsseln → **erkannt**
③ wie ②, und der Angreifer hat uns **seinen Public Key als den des Absenders untergejubelt**	Prüfung „gelingt“ → **nicht erkannt** ❌

💡 Das zentrale Problem: **Woher weiß ich, dass ein Public Key wirklich zu der Person oder dem Server gehört?** → **Zertifikate**.

✏️ Übungsfragen Block 2

Ü2.1 Warum reicht es nicht, eine Datei einfach zusammen mit ihrem Hashwert zu verschicken?

Ü2.2 Anna will Ben ein Dokument signiert schicken. Mit welchem Schlüssel signiert Anna, mit welchem prüft Ben?

Ü2.3 Welche Schutzziele erfüllt eine digitale Signatur, welches nicht?

📜 Block 3: Digitale Zertifikate IHK sehr hoch

> 🎯 **IHK: sehr hoch** HTTPS und Zertifikate sind AP2-Klassiker

Ein **Zertifikat** ist im Kern ein **Public Key, der von einer vertrauenswürdigen Stelle signiert** wurde. Es bestätigt: „Dieser Public Key gehört wirklich zu shop.com." Schutzziel: **Authentizität**.

⚠️ Das lässt sich **nicht rein kryptographisch** lösen. Am Ende muss man **jemandem vertrauen**.

Bestandteile nach X.509

Feld	Frage
Aussteller (Zertifizierungsstelle, CA)	Wer hat zertifiziert?
Inhaber (Subject, z. B. `shop.com`)	Was wurde zertifiziert?
Public Key des Inhabers	um den geht es
Signaturalgorithmus (z. B. sha256WithRSAEncryption)	Wie wurde signiert?
Gültigkeitsdauer (gültig von / bis)	Wie lange gilt es?
Seriennummer	eindeutige Nummer, wichtig zum **Sperren**
… und die **Signatur der CA** über alle diese Angaben	

🔎 **Laufzeit von Webserver-Zertifikaten:** Auf der Folie stehen 1–2 Jahre. Das ist veraltet. Seit März 2026 sind höchstens **200 Tage** erlaubt. Ab 2027 sinkt die Grenze auf 100 Tage, ab 2029 auf **47 Tage**. Deshalb erneuert man Zertifikate heute **automatisch** (z. B. **Let's Encrypt** mit **ACME**/certbot, Laufzeit 90 Tage).

Zertifikatsprüfung = Signaturprüfung

1. Die Signatur im Zertifikat mit dem **Public Key der CA** entschlüsseln → Original-Hash.
2. Die Zertifikatsdaten selbst hashen.
3. Vergleichen. Gleich → Zertifikat und damit der Public Key sind **echt**.
4. 🔎 Außerdem prüft der Browser: Ist es **noch gültig**, passt der **Name** zur aufgerufenen Adresse, wurde es **gesperrt**?

Wem vertraut man? Die „vertrauenswürdigen Stellen“

- Um die CA-Signatur zu prüfen, braucht man den **Public Key der CA**. Wer garantiert dessen Echtheit?
- Lösung: Betriebssystem und Browser bringen ab Werk eine Liste von **Root-Zertifikaten** mit (🔎 **Zertifikatsspeicher / Root Store**, unter Windows `certmgr.msc`, unter Debian `/etc/ssl/certs`).
- **Wer** darin steht, entscheiden die **Hersteller** (Microsoft, Apple, Google, Mozilla …). Wir können dem nur vertrauen.

Ergänzung: Vertrauenskette (Chain of Trust)



Der Browser prüft von rechts nach links, bis er bei einer Root-CA ankommt, der er vertraut

Vertrauenskette

Vertrauenskette

🔎 **Zertifikatskette (Chain of Trust):** Die Root-CA signiert meist nicht direkt, sondern eine **Zwischen-CA** (Intermediate), und diese signiert das Server-Zertifikat. Der Browser prüft die Kette, bis er bei einer Root-CA ankommt, der er vertraut.

🔎 **Selbstsigniertes Zertifikat:** Aussteller = Inhaber, keine CA hat bestätigt. Der Browser warnt („Verbindung ist nicht privat“). Das ist in Ordnung für interne Tests oder Laborserver, aber nicht für öffentliche Seiten.

✏️ Übungsfragen Block 3

Ü3.1 Was ist ein digitales Zertifikat? Antworten Sie in einem Satz.

Ü3.2 Nennen Sie vier Angaben, die ein X.509-Zertifikat enthält.

Ü3.3 Ein Kunde ruft das Firmen-Intranet auf und bekommt die Warnung „Zertifikat nicht vertrauenswürdig“. Nennen Sie zwei mögliche Ursachen.

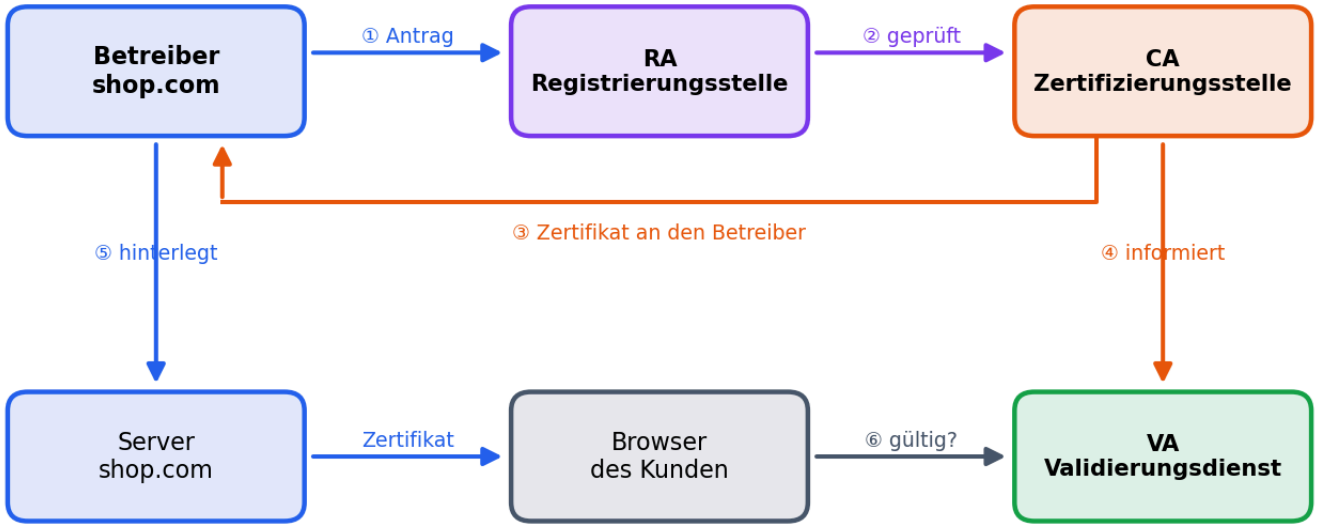
🏛️ Block 4: PKI – Public Key Infrastructure IHK hoch

> 🎯 **IHK: hoch** die Abkürzungen RA, CA und VA werden gern abgefragt

PKI = ein System, das Zertifikate **ausstellt, verwaltet und prüft**.

Stelle	Name	Aufgabe
RA	Registration Authority – **Registrierungsstelle**	nimmt Anträge an, **prüft die Identität** des Antragstellers, gibt sie an die CA weiter
CA	Certificate Authority – **Zertifizierungsstelle**	**stellt das Zertifikat aus** (signiert es) und informiert die VA
VA	Validation Authority – **Validierungsdienst**	beantwortet Anfragen: **Ist dieses Zertifikat gültig?**

PKI – Ablauf am Beispiel shop.com



PKI-Ablauf

PKI-Ablauf

Ablauf am Beispiel shop.com:

1. Der Betreiber beantragt ein Zertifikat bei der **RA**.
2. Die RA prüft den Antrag und gibt ihn an die **CA** weiter.
3. Die CA stellt das Zertifikat aus …
4. … und informiert die **VA**.
5. Der Betreiber hinterlegt das Zertifikat auf seinem **Webserver**.
6. Ruft ein Kunde shop.com auf, fragt der **Browser** bei der VA nach, ob das Zertifikat gültig ist.

🔎 Gesperrte Zertifikate (z. B. weil der Private Key gestohlen wurde) stehen in einer **Sperrliste (CRL, Certificate Revocation List)**. Online abfragen kann man sie per **OCSP** (Online Certificate Status Protocol).

✏️ Übungsfragen Block 4

Ü4.1 Wofür stehen RA, CA und VA, und was macht jede Stelle?

Ü4.2 Der Private Key eines Webservers wurde gestohlen. Was muss mit dem Zertifikat passieren, und wie erfahren Browser davon?

⚖️ Block 5: Elektronische Signaturen im Recht (eIDAS) IHK mittel

> 🎯 **IHK: mittel** die drei Stufen kommen vor, vor allem die QES

🔎 Rechtsgrundlage ist die EU-Verordnung **eIDAS** (gilt in allen EU-Staaten).

Stufe	Sicherheit	Technik	Beispiele	vor Gericht
Einfache elektronische Signatur	sehr niedrig	**keine** Kryptographie	Name unter einer Mail, eingescannte Unterschrift, Unterschrift auf dem Paket-Lesegerät	einsetzbar, abe leicht zu bestreiten. Das Gericht entscheidet fre
Fortgeschrittene elektronische Signatur	hoch	**digitales Zertifikat**. Der Aussteller ist egal, auch ein **selbst erstelltes** Zertifikat geht	Signatur-Datei im Anhang (🔎 z. B. S/MIME, PGP)	einsetzbar, Echtheit muss **bewiesen** we
Qualifizierte elektronische Signatur (**QES**)	sehr hoch	technisch wie die fortgeschrittene, aber mit Zusatzanforderungen (siehe unten)	Vertrag, Bescheid, beA	**gleichgestell mit der handschriftli Unterschrift** (weit)

Zusätzlich bei der QES:

- Das Zertifikat gibt es erst nach einer **Identitätsprüfung** (z. B. Personalausweis).
- Es muss ein **qualifiziertes Zertifikat** eines **anerkannten** Vertrauensdienstes sein. **Keine** selbst erstellten Zertifikate.
- Signiert wird mit einer **sicheren Signaturerstellungseinheit (SSEE)**, z. B. Chipkarte + Kartenleser.
- Die SSEE ist von einer anerkannten Stelle **geprüft**.
- Der **Private Key** liegt **nur in der SSEE** und kann nicht herauskopiert werden.

🔎 Heute gibt es auch die **Fernsignatur**: Der Private Key liegt in einem zertifizierten Hochsicherheitsmodul (HSM) beim Anbieter, und man löst die Signatur per App oder TAN aus. Das ist ebenfalls eine QES.

Anerkannte Anbieter in Deutschland (Folie): D-Trust (Bundesdruckerei), Bundesnotarkammer, Signtrust (Deutsche Post), S-Trust (Deutscher Sparkassenverlag), Telesec (Deutsche Telekom). Die EU-weite Liste steht unter *eidas.ec.europa.eu*.

💡 Merksatz: **Nur die QES ersetzt die Unterschrift auf Papier.**

🔎 Bezug zu **Recht**: Wo das Gesetz die **Schriftform** verlangt, kann sie meist durch die QES ersetzt werden (§ 126a BGB). Ausnahme: Die **Kündigung eines Arbeitsverhältnisses** muss weiterhin auf Papier eigenhändig unterschrieben sein (§ 623 BGB). Eine E-Mail oder QES reicht dort **nicht**.

✏️ Übungsfragen Block 5

Ü5.1 Ihr Betrieb will Verträge mit Kunden rein digital unterschreiben lassen, rechtlich so sicher wie auf Papier. Welche Signaturstufe brauchen Sie, und was ist dafür nötig?

Ü5.2 Warum ist eine Signatur mit einem selbst erstellten Zertifikat keine QES?

📋 Spickzettel

Begriff	In einem Satz
Hashfunktion	beliebige Daten → Fingerabdruck fester Länge, nicht umkehrbar
Kollisionsresistenz	zwei Eingaben ergeben praktisch nie denselben Hash
Lawineneffekt	kleine Änderung → völlig anderer Hash
MD5 / SHA-1	veraltet, Kollisionen bekannt
SHA-256	aktueller Standard, 256 Bit
Salt	zufälliger Zusatz pro Passwort, nicht geheim, gegen Rainbow Tables

Begriff	In einem Satz
Digitale Signatur	Hash, verschlüsselt mit dem eigenen Private Key → Integrität + Authentizität
Signatur prüfen	mit dem Public Key des Absenders entschlüsseln, selbst hashen, vergleichen
Zertifikat	Public Key + Angaben, signiert von einer CA → Authentizität
X.509	Standard für den Aufbau von Zertifikaten
Root Store	im System/Browser hinterlegte vertrauenswürdige Root-Zertifikate
PKI	RA prüft, CA stellt aus, VA bestätigt die Gültigkeit
CRL / OCSP	Sperrliste / Online-Abfrage gesperrter Zertifikate
QES	qualifizierte Signatur = rechtlich wie handschriftlich

✅ Lösungen

Ü1.1 Zwei von: **Einwegfunktion** (aus dem Hash lässt sich die Eingabe nicht berechnen), **Kollisionsresistenz** (zwei verschiedene Eingaben ergeben praktisch nie denselben Hash), **Lawineneffekt** (kleinste Änderung → völlig anderer Hash).

Ü1.2 1. Hashen ist **keine Verschlüsselung**: kein Schlüssel, nicht umkehrbar. 2. **MD5 ist veraltet und unsicher** (Kollisionen, extrem schnell zu berechnen). Besser: ein langsames Verfahren mit Salt, z. B. bcrypt, Argon2 oder yescrypt.

Ü1.3 Es gibt unendlich viele mögliche Eingaben, aber nur endlich viele Hashwerte (feste Länge). Irgendwann müssen sich also zwei Eingaben einen Hash teilen. Man macht die Zahl der Hashwerte nur so groß (z. B. 2^{256}), dass eine Kollision praktisch nicht zu finden ist.

Ü2.1 Ein Angreifer kann die Datei verändern, einen **neuen Hash** berechnen und beides weiterschicken. Der Empfänger merkt nichts. Der Hash muss deshalb mit dem Private Key des Absenders geschützt (signiert) werden.

Ü2.2 Anna signiert mit **ihrem eigenen Private Key**. Ben prüft mit **Annas Public Key**.

Ü2.3 Erfüllt: **Integrität** und **Authentizität** (🔎 plus Nichtabstreitbarkeit). **Nicht** erfüllt: **Vertraulichkeit**, denn das Dokument bleibt lesbar.

Ü3.1 Ein Zertifikat ist ein **Public Key mit Angaben zum Inhaber, signiert von einer vertrauenswürdigen Zertifizierungsstelle (CA)**. Es bestätigt, dass der Schlüssel wirklich zum Inhaber gehört.

Ü3.2 Vier von: Aussteller (CA), Inhaber, Public Key des Inhabers, Signaturalgorithmus / Hashfunktion, Gültigkeitsdauer, Seriennummer.

Ü3.3 Zwei von: Das Zertifikat ist **selbstsigniert** oder von einer **internen CA**, die auf dem Client nicht im Zertifikatsspeicher steht. Es ist **abgelaufen**. Der **Name passt nicht** zur Adresse (z. B. `intranet` statt `intranet.firma.de`). Es wurde **gesperrt**. Die **Zwischen-CA fehlt** auf dem Server.

Ü4.1 **RA** (Registrierungsstelle) nimmt Anträge an und prüft die Identität. **CA** (Zertifizierungsstelle) stellt das Zertifikat aus bzw. signiert es. **VA** (Validierungsdienst) bestätigt auf Anfrage, ob ein Zertifikat gültig ist.

Ü4.2 Das Zertifikat muss bei der CA **gesperrt (widerrufen)** werden. Dann muss ein **neues Schlüsselpaar** mit neuem Zertifikat her. Browser erfahren es über die **Sperrliste (CRL)** bzw. **OCSP**, also über den Validierungsdienst.

Ü5.1 Die **qualifizierte elektronische Signatur (QES)**. Nötig sind ein **qualifiziertes Zertifikat** eines **anerkannten Vertrauensdienstes** (z. B. D-Trust) nach **Identitätsprüfung** und eine **sichere Signaturerstellungseinheit** (Chipkarte + Leser oder Fernsignatur beim Anbieter).

Ü5.2 Bei einem selbst erstellten Zertifikat hat **niemand die Identität geprüft**. Jeder könnte eins auf einen beliebigen Namen ausstellen. Die QES verlangt ein **qualifiziertes Zertifikat eines anerkannten Anbieters** und eine geprüfte SSEE.

Zusammengeführt aus 4 Einzel-Zusammenfassung(en) · Stand 25.11.2026