IT-Sicherheit 3 (Verschlüsselung) – Zusammenfassung mit IHK-Bezug

Kerckhoffs, symmetrisch / asymmetrisch / hybrid, DES · AES · RSA · ECC, TLS, Ende-zu-Ende, PFS, Diffie-Hellman

🔐 IT-Sicherheit 3 – Verschlüsselung

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

🧭 So liest du dieses Dokument

- **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-Relevanz: **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	n = 13	n = 5

	Was ist das?	Bei ROT-13	Bei der Transposition
	Folge aus 0 und 1)		

🏛️ 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
Chi	**I**	**U**	**H**	**V**	**L**	**S**	**Q**	**W**	**O**	**R**

Klartext	B	U	N	D	E	S	W	E	H	R
ffrat										

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-Relevanz: **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.

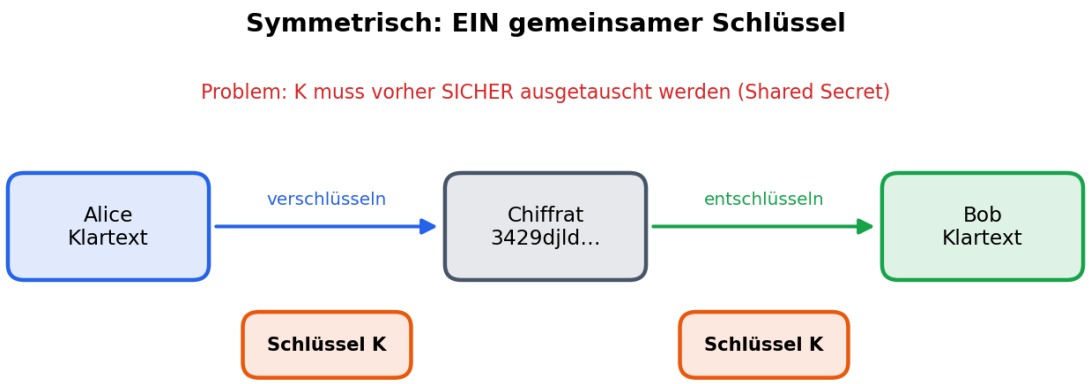
✏️ Übungsfragen Block 2

Ü2.1 Berechnen Sie 1100 XOR 1010.

Ü2.2 Was bedeutet „praktisch nicht knackbar“?

🔑 Block 3: Symmetrische Verschlüsselung

🎯 IHK-Relevanz: **sehr hoch**.



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

Verfahren	Block	Schlüssel	Status
			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).

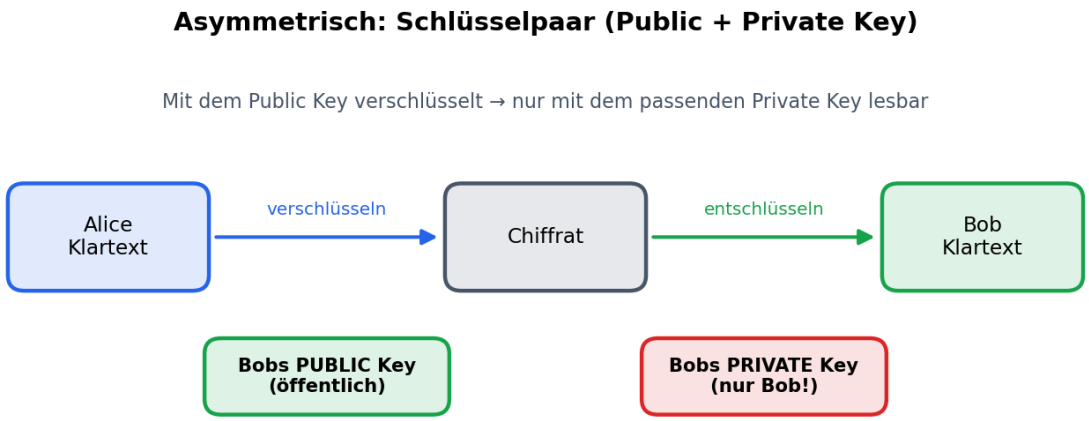
✏️ Übungsfragen Block 3

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-Relevanz: **sehr hoch**.



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.

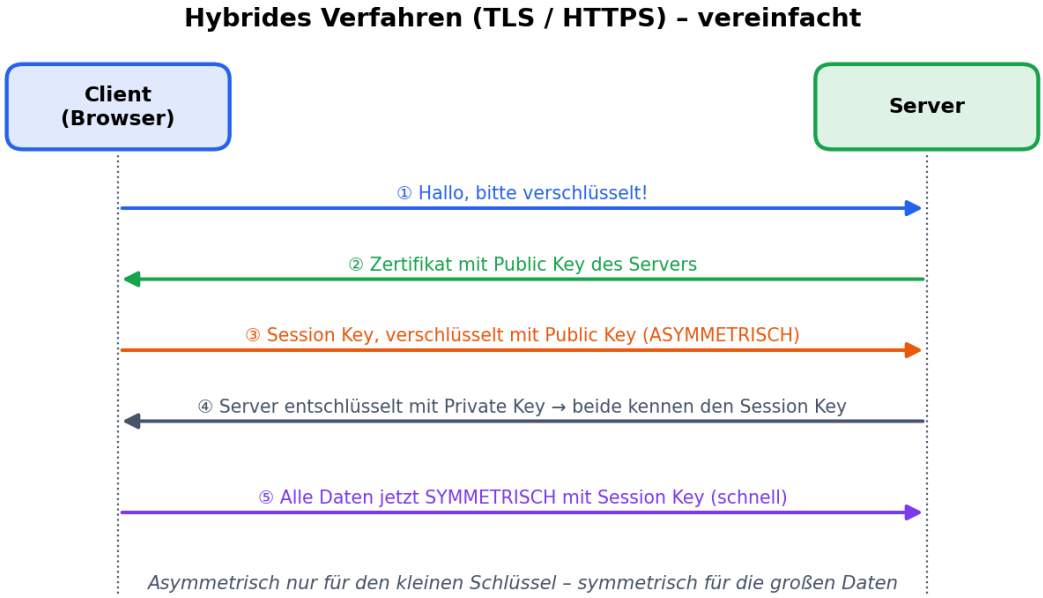
🤝 Block 5: Hybride Verfahren (SSL / TLS)

🎯 IHK-Relevanz: **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

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.

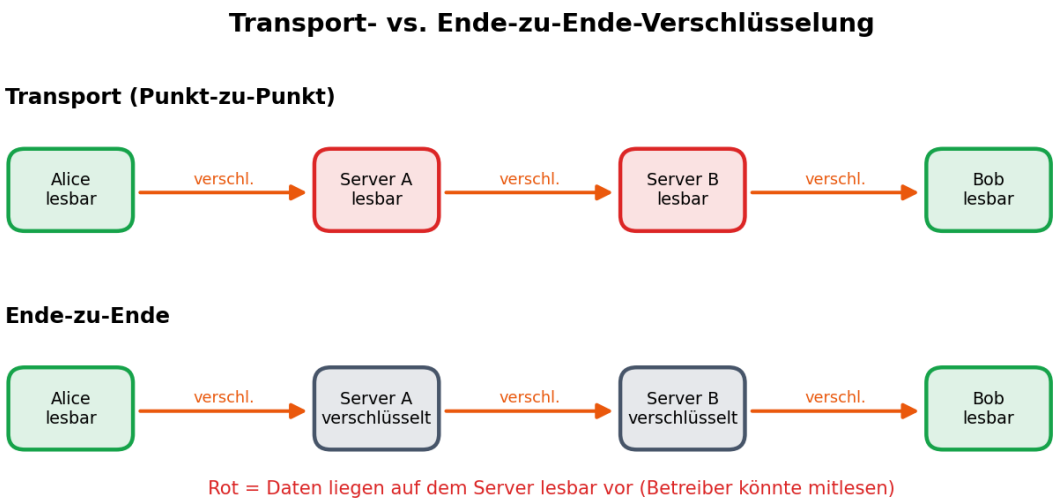
✏️ Übungsfragen Block 5

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-Relevanz: **hoch**.



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
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-Relevanz: **mittel** – PFS als Begriff, Diffie-Hellman als Prinzip.

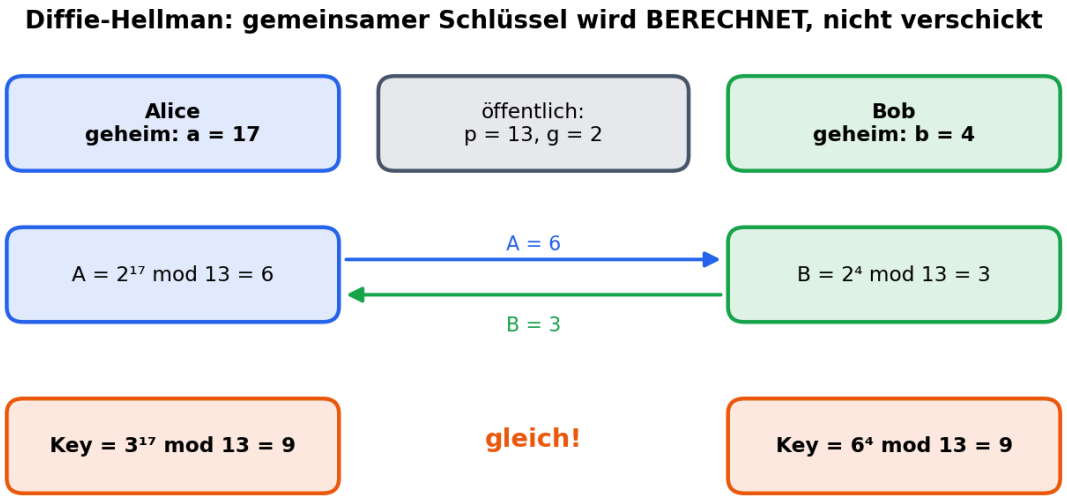
Das Problem beim klassischen Hybrid-Verfahren

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

Schritt	Alice	öffentlich	Bob
1. Vereinbaren		Primzahl **p = 13**, Basis **g = 2**	

Schritt	Alice	öffentlich	Bob
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

Begriff	In einem Satz
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 mod 11 = 8 · B = 2^4 mod 11 = 16 mod 11 = 5. Alice: 5^3 mod 11 = 125 mod 11 = 125 − 121 = **4**. Bob: 8^4 mod 11 = 4096 mod 11 = 4096 − 372 · 11 = 4096 − 4092 = **4**. Gemeinsamer Schlüssel: **4** ✔️