

Zusammenfassung Modul Kryptologie

David Jäggli

13. September 2024

Inhaltsverzeichnis

1	Allg	3
1.1	Terminologie	3
1.2	Trapdoor Weiterführung	4
1.2.1	Einwegfunktion ohne Trapdoor	4
1.2.2	Einwegfunktion mit Trapdoor	4
2	Schutzmechanismen	5
2.1	Sicherheitsanforderungen	5
2.2	Geheimhaltung / Verschlüsselung	6
2.3	Authentizität	6
3	Symmetrische Kryptographie	7
3.1	Blockschiffren	7
3.1.1	Blockschiffren Eig.	7
3.1.2	Good to know	8
3.2	Stromchiffren	8
3.2.1	One-Time-Pad (OTP)	9
3.2.2	XOR	9
3.2.3	Beurteilung von Stromchiffren	9
3.3	Blockchiffren	9
3.3.1	ECB-Modus für mehrere Blöcke	9
3.3.2	CBC-Modus für mehrere Blöcke	9
3.3.3	OFB-Modus = Output Feedback Mode	10
3.3.4	CTR-Modus (Counter Mode)	10
3.4	Auswirkungen bei Bit Manipulation im Chiffretext	11
3.5	Integritätsschutzmechanismen	11
4	Asymmetrische Kryptographie	12
4.1	Allgemein	12

4.2	Einwegpermutationen	12
4.3	Elliptic Curve Cryptography (ECC)	14
4.3.1	Requirements	15
4.3.2	Punktaddition	16
4.3.3	Neutrales und Inverses Element	16
4.3.4	Allgemeine Form & Zusammenfassung:	18
4.3.5	Formel zur Addition von 2 Punkten	18
4.3.6	Double and add Algorithmus	19
4.3.7	Bestimmung aller Punkte	20
4.3.8	Bestimmung ob Punkt auf Graf ist	20
4.4	RSA	21
4.4.1	Anzahl der Primzahlen 1-x	21
4.4.2	Anzahl n-stelligen Primzahlen	21
4.4.3	Dezimalstellen ausrechnen:	22
4.5	Diskreter Logarithmus bei ECC	22
4.5.1	Key exchange	22
4.6	Verschlüsselung mit EC nach Volker Müller	23
5	Blinde Signaturen	24
5.1	Beweis der Korrektheit	25
5.2	Beispiel mit Zahlen	26
5.3	Basis-Test	27
6	Einführung in die Public-Key Infrastruktur (PKI)	28
6.1	Verschlüsseln und Signieren (repetition)	28
6.1.1	Verschlüsseln	28
6.1.2	Signieren	28
6.2	Zertifikate	29
6.2.1	Herstellung eines Zertifikats	29
6.2.2	Installation eines neues (Root-)Zertifikates	29
6.2.3	Überprüfung der Echtheit eines Zertifikates vom Betriebssystem	29
6.2.4	Zertifikatsklassen	30
7	Protokolle	31
7.1	User Authentication	31
7.2	False-rates	31
7.3	Verifikationen	31
7.3.1	One to many	31
7.3.2	Many to one	31
7.4	Paralellsession Attacke	32
8	Quantenkryptographie	33
8.1	Polarization	33
8.2	Quantum Key Exchange	33

1 Allg

Allgemeine Begriffe und Definitionen.

1.1 Terminologie

Kryptographie	Entwerfen von Krypto-Algorithmen
Kryptoanalyse	Brechen von Krypto-Algorithmen
Perfekte Sicherheit	Unendlich viele Ressourcen sind equivalent zu raten
Unkeyed Kryptographie	Hashfunktionen
Symmetrische Krypt.	Beide den gleichen Schlüssel - $\mathcal{O}(n^2)$
Asymmetrische Krypt.	Öffentlicher und privater Schlüssel - $\mathcal{O}(n)$
Stromchiffren	Verschlüsselung von einzelnen Zeichen
Blockchiffren	Verschlüsselung von Blöcken
MAC	Message Authentication Code
DES	Data Encryption Standard
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
CA	Certificate Authority
CRL	Certificate Revocation List - Zertifikatssperrliste
Mutual	Two-way Authentication. Beide Parteien authentifizieren sich
EWf (mit) Trapdoor	Es existiert Geheimnis (Trapdoor) mit der sich $f^{-1}(y)$ einfach berechnen lässt. Die Sicherheit ist, dass dieser "Trick" nur einer Partei bekannt ist.

1.2 Trapdoor Weiterführung

Untenstehend sind die Unterschiede von Einwegfunktionen mit und ohne Trapdoor näher erläutert.

1.2.1 Einwegfunktion ohne Trapdoor

Beispiele und Erläuterungen für Einwegfunktionen ohne Trapdoor:

- Diffie-Hellman $y = f(x) = a^x \mod p$.
 - p prim ist rechenintensiv aber einfach.
 - Das Inverse (disk. Log.) $x \equiv f^{-1}(y) \equiv \log_a(y) \mod p$ ist für alle schwierig.
- Elliptische Kurven
 - Multiplikation $Q = k \cdot P$, k eine unbekannte Zahl, P, Q bekannte Punkte.
 - Das Bestimmen von k aus $Q = k \cdot P$ ist für alle schwierig.
 - Die Punkt-Division von $k = \frac{Q}{P}$ gibt es in diesem Sinne nicht.

1.2.2 Einwegfunktion mit Trapdoor

Beispiele und Erläuterungen für Einwegfunktionen mit Trapdoor:

- Bei RSA: Die e-te Wurzel mod N berechnen: Trapdoor
 - Das Potenzieren $y = f(x) \equiv x^e \mod N$ ist rechenintensiv aber einfach.
 - Das Inverse $x \equiv f^{-1}(y) \equiv y^{1/e} \equiv \sqrt[e]{y} \mod N$ ist extrem schwierig.
 - Das Berechnen der “e-ten Wurzel mod N” ist das eine Problem, das Faktorisieren von $N = p \cdot q$ das andere. Der RSA kann gebrochen werden, wenn nur eines der zwei Probleme gelöst ist.

2 Schutzmechanismen

Kleine Übersicht:

Sicherheitsdienst Krypt. Mechanismus	Vertraulichkeit	Datenin- tegrität	Partnerauthentizi- tät	Keiner dieser Dienste
Sym. Verschlüsselung	X			
Asym. Verschlüsselung	X			
Diffie-Hellman Schlüs- selaustausch Protokoll				X
Hybride Verschlüsse- lung	X			
MAC- Berechnung		X		
Digitale Signatur		X		
C-R Protokoll mit MAC			X	
C-R Protokoll mit digita- ler Signatur			X	

Abbildung 1: Mechanismus vs Sicherheitsdienst

2.1 Sicherheitsanforderungen

- Vertraulichkeit (Abhören)
- Integrity (Daten Verändern)
- Insertion (Daten Einfügen)
- Non repudiation of origin (Abstreiten die Meldung geschickt zu haben)
- Replay (Abfangen und wieder senden)
- Delete
- Non repudiation of receipt (Abstreiten die Meldung erhalten zu haben)
- Authentisierung

2.2 Geheimhaltung / Verschlüsselung

K_e = Key Encrypt

K_d = Key Decrypt

Ist $K_e = K_d \rightarrow$ symmetrische Verschlüsselung.

Ist $K_e \neq K_d \rightarrow$ asymmetrische Verschlüsselung.

2.3 Authentizität

K_g = Key Generate

K_v = Key Verify

Ist $K_g = K_v \rightarrow$ MAC.

Ist $K_g \neq K_v \rightarrow$ digitale Signatur.

3 Symmetrische Kryptographie

Note: Stromschiffren bieten keine Integrität.

Note: Blockchiffren bieten Integrität.

Ausserdem können Blockchiffren verwendet werden für:

- Hashfunktionen
- Pseudo random number generators
- Message Authentication Code (MAC)

3.1 Beschreibung von Blockchiffren

Grundsätzlicher Ablauf/Aufbau:

1. n-Bit Inputblockgrösse
2. n-Bit Outputblockgrösse
3. k-Schlüsselbit
4. x-Verschlüsselungsdurchläufe

übliche Blockgrössen: 64, 128, 256 Bit

Anzahl Runden: 10, 12, 14, 16

Übliche Schlüssellängen: k = 56 (DES), 112 (3DWS_2key),

k = 128 (AES), 168 (3DES_3key)

k = 192, 256 (AES) Bit

3.1.1 Eigenschaften von Blockchiffren

1. (Gleicher Schlüssel) Änderung eines Input bits ändert die Hälfte der Outputbits
2. (Gleicher Input) Änderung eines Schlüsselbits ändert die Hälfte der Outputbits
3. Mit Binomialkoeffizient kann gezeigt werden, dass diese Eigenschaften ideal sind.

Allgemein ist folgender Binom Ideal wenn $m = n/2$:

$$\binom{n}{m} = \frac{n!}{m! \cdot (n - m)!}$$

3.1.2 Good to know

- 3DDES_2key (112 Bit Schlüssellänge) hat Sicherheit von 57-60 Bit
- 3DES_3key (168 Bit Schlüssellänge) hat Sicherheit von 112 Bit
- AES braucht für 128 Bit 10 Runden, für 192 Bit 12 Runden und für 256 Bit 14 Runden
- AES braucht für Entschlüsselung doppelt so lange wie für Verschlüsselung
- AES hat nicht den gleichen Algorithmus für Ver-/Entschlüsselung
- PRESENT ist optimiert für 8-Bit Prozessor (IoT)
- PRESENT braucht weniger Energie, SW & HW

3.2 Stromchiffren

XOR - Ziemlich straight forward, aufgrund von Key wird ein Pseudo-Random Sequence generiert, mit der dann der Plaintext verschlüsselt (XORd) wird. Entschlüsselung genau gleich

Abkürzungen: C = Cipher

M = Message

S = Sequence

Verschlüsselung: $C = M \oplus S$

Entschlüsselung: $M = C \oplus S = (M \oplus S) \oplus S = M \oplus (S \oplus S) = M \oplus 0 = M$

Charakteristiken:

- Keygenerator
 - (Bankenwelt) of ein Blockverschlüssler
 - (Netzwerkwelt) of spezielle Konstruktion mit Hashfunktionen wie SHA-1
- Symmetrisch
- Vor allem bei Link-Verschlüssler als HW erhältlich.
- Keine Authentizität/Integrität
- Ein Verändern im Chiffretext verändert nur die betroffenen Zeichen
- Wird jedoch ein Bit hinzugefügt oder entfernt, ist die Hölle los

3.2.1 One-Time-Pad (OTP)

True-random Key, welcher gleich lang ist wie der Plaintext \rightarrow perfekte Sicherheit.

OTP ist eine Stromchiffre, nicht jede Stromchiffre ist ein OTP.

3.2.2 XOR

Zwei gleiche Ausdrücke XORen ergibt 0.

Beispiel:

$$6B \oplus A5 = CE = (6XORA)(BXOR5)$$

3.2.3 Beurteilung von Stromchiffren

1. Schutzziele: Schutz gegen das Abhören der Meldung
2. Mit was für Typen / Angriffen muss gerechnet werden: gegen intelligente Gegner, welche passiven Angriff (abhören) durchführen
3. Stromchiffre Mechanismus: Symmetrische Verschlüsselung

3.3 Blockchiffren

Guess what, verschlüsseln immer einen Block und hängen diese hintereinander.

3.3.1 ECB-Modus für mehrere Blöcke

Electronic Code Book Mode, einfach alle Blöcke einzeln verschlüsseln.

$$C_i = E(M_i, K)$$

Gleiche Blöcke ergeben gleiche Chiffretexte.

3.3.2 CBC-Modus für mehrere Blöcke

Cipher Block Chaining, XOR des vorherigen Chiffretextes (oder IV) mit dem Plaintext. Respektive braucht einen IV (Initialisierungsvektor) für den ersten Block. Danach dient der Chiffretext des vorherigen Blocks als IV für den nächsten Block.

$$C_i = E(M_i \oplus C_{i-1}, K) \text{ mit } C_0 = IV$$

Gleiche Blöcke ergeben nicht mehr gleiche Chiffretexte.

Wenn die Länge des Plaintextes nicht durch die Blockgrösse teilbar ist, muss der letzte Block anders behandelt werden. Zum Beispiel einfach abschneiden. Die Entschlüsselung ändert nicht.

Die Formel für den letzten Block ist: $C_4 = E(C_3, K) \oplus M_4$

3.3.3 OFB-Modus = Output Feedback Mode

Ist grundsätzlich eine Stromchiffre. Hier wird Blockchiffre wird nur als Zufallszahlengenerator verwendet.

OFB-Modus, OFB = Output Feedback

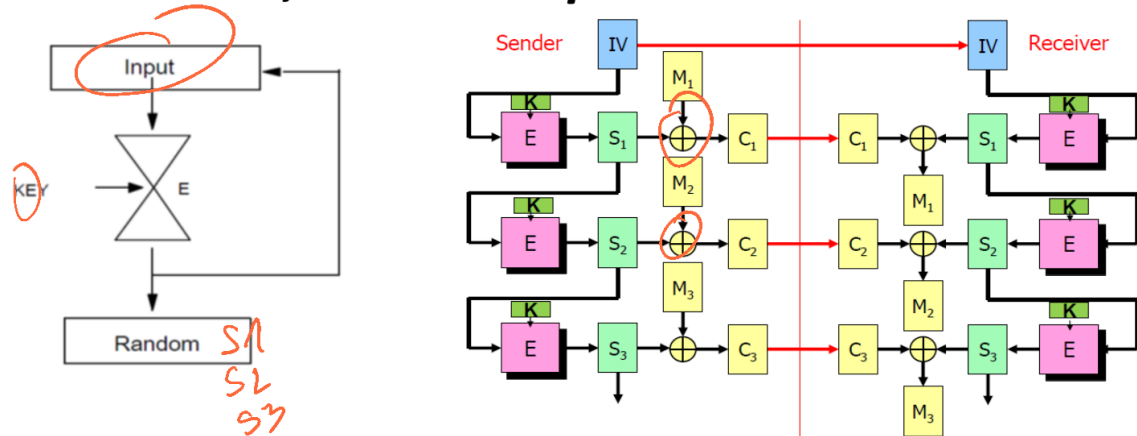


Abbildung 2: OFB Mode

Verschlüsseln:

$$C_i = S_i \oplus M_i \text{ mit } S_i = E(S_{i-1}, K) \text{ und } S_0 = IV$$

Entschlüsseln:

$$M_i = S_i \oplus C_i \text{ mit } S_i = E(S_{i-1}, K) \text{ und } S_0 = IV$$

3.3.4 CTR-Modus (Counter Mode)

Der XOR Schlüssel ist nicht mehr der vorherige (verschlüsselte) Block sondern einfach $IV + i$.

Hat Vorteile von beiden:

1. Ist parallelisierbar
2. Verschlüsselung eines grossen Files / Harddisk möglich
3. Gleicher Klartext ergibt nicht gleicher Output
4. Vorausberechnung von Schlüsselmaterial ist möglich
5. Vertauschen von Chiffreblöcken ist nicht mehr einfach.

Mathematische Form: $C_i = S_i \oplus M_i$ mit $S_i = E(IV + (i - 1), K)$

Strom diffbe

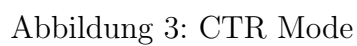


Abbildung 4: Auswirkung auf Klartext bei Bit Manipulation im Chiffretext

3.5 Integritätsschutzmechanismen

MAC halt irgendwie mit CBC oder CTR Mode und HMAC mit Hashing Algorithmus.

4 Asymmetrische Kryptographie

4.1 Allgemein

Sicherheitsvergleich klassisch:

Symmetric	56	80	112	128	192	256
RSA n	512	1024	2048	3072	7680	15360
ECC p	112	160	224	256	384	512
Key-Ratio	5:1	6:1	9:1	12:1	20:1	30:1

Sicherheit mit Quantencomputer:

1. Für Faktorisierung bei RSA K braucht ca. $2k$ Qubits
2. Für DL bei ECC von k -Bits braucht ca. $K \approx 5k + 8\sqrt{k} + 4 \cdot \log_2(k)$ Qubits

4.2 Einwegpermutationen

RSA, ECC und Diffie-Hellman sind genau genommen Einwegpermutationen.

Die Farben blau, grün und rot kann man auf $3! = 6$ Arten anordnen:

(B, G, R), (B, R, G), (G, B, R), (G, R, B), (R, B, G), (R, G, B)

Im RSA-System:

$$N = 3 \cdot 11 \text{ und } e = 13 \Rightarrow d = 13^{-1} \bmod \varphi(N) = 17 = 13^{-1} \bmod 20$$

Nun kann man alle Werte $x \in \mathbb{Z}_{33} = \{0, \dots, 32\}$ mit $y \equiv x^{13} \bmod 33$ berechnen verschlüsseln.

Beispielaufgabe:

Die 3-stellige Zahl x wird mit der Formel $y \equiv (a \cdot x + b) \bmod N$ verschlüsselt. Die Entschlüsselungsfunktion lautet: $x \equiv a^{-1} \cdot (y - b) \bmod N$. Dabei ist $N = 11 \cdot 23 \cdot 41$ ein Produkt von drei Primzahlen; der Wert N ist öffentlich bekannt. Die Werte a und b bilden in der Form $(a; b)$ den geheimen Schlüssel. Die Werte a und b sind für die Verschlüsselung und Entschlüsselung geeignete Werte aus der Menge $\{2; 3; \dots; N - 1\}$

Aus wie vielen möglichen Schlüsseln der Form $(a; b)$ können bei dieser Verschlüsselung ausgewählt werden? Es ist die exakte Zahl anzugeben.

Allgemein:

- Für b sind alle möglichen Werte aus der Menge $\{2; 3; \dots; N-1\}$ erlaubt. Das sind $N-2$ Möglichkeiten.
- Für a sind aus der Menge $\{2; 3; \dots; N-1\}$ alle Werte erlaubt, die teilerfremd zu N sind. Da die Zahl 1 aber nicht drin sein darf lautet die Anzahl der möglichen Werte für a somit: $\varphi(N)-1 = \varphi(r \cdot s \cdot t)-1 = \varphi(r) \cdot \varphi(s) \cdot \varphi(t)-1 = (r-1) \cdot (s-1) \cdot (t-1)-1$
- somit gibt es total $(N-2) \cdot [(r-1) \cdot (s-1) \cdot (t-1)-1]$ Möglichkeiten

Mit Zahlen:

$$r = 11; s = 23; t = 41 \Rightarrow N = 11 \cdot 23 \cdot 41 = 10'373$$

$$\Rightarrow N - 2 = 10'371$$

$$\Rightarrow \varphi(N) - 1 = \varphi(r \cdot s \cdot t) - 1 = (11-1) \cdot (23-1) \cdot (41-1) - 1 = 10 \cdot 22 \cdot 40 - 1 = 8'799$$

4.3 Elliptic Curve Cryptography (ECC)

- Was: mathematische Objekte, die man als Public-Key Kr. verwenden kann
- Warum:
 - wenige und schnelle Operationen (statt viele langsame wie RSA) wegen Multiplikationen statt Exponentiationen
 - wenig Speicherplatz (für Chipkarten ein Kriterium)
 - Es gibt Standards
 - Patent-freie Algorithmen
- Man kann:
 - Signieren
 - Schlüsselaustauschen
 - Hybrid Verschlüsseln
 - (Wie RSA)
- Könnte RSA aufgrund der langen Schlüssel ablösen (vielleicht auch Quantenalgorithmen)
- Es wird addiert und verdoppelt statt multipliziert und potenziert (RSA)
- Anzahl der Punkte kann Atome im Weltall übertreffen
- Ist eine Einwegfunktion ohne Trapdoor
- Die Sicherheit liegt im diskreten Logarithmus Problem
- Diskreter Logarithmus: für P, Q bei $q = i \cdot P$ kann i nicht einfach berechnet werden
- Der Schnittpunkt des Koordinatensystems ist $P(0, 0)$, $P(0, 0)$ kann Inhalt von EC sein.
- Der unendlich ferne Punkt \mathcal{O} ist das neutrale (null-)Element
- Die Gruppenordnung ist Anzahl der Punkte (somehow NICHT p)
- Für gleiche Sicherheit: Ein 3072 Bit RSA entspricht etwa einer 256 Bit EC.

4.3.1 Requirements

Hat immer die Form:

$$y^2 = x^3 + ax + b$$

und ist NICHT

$$y = f(x)$$

Nichtsingularitätsbedingung:

$$4a^3 + 27b^2 \not\equiv 0 \pmod{p}$$

Diese muss erfüllt sein, ansonsten gibt es mehrere Wurzeln (Ergebnisse).

Und ist sie erfüllt, hat die Kurve sowas wie eine Spitze oder Überschneidungen.

Diese beiden Bedingungen stellen sicher, dass die Verbindung zwischen zwei Punkten genau einen weiteren Punkt schneidet. Bei Tangenten wird der Berührungspunkt doppelt gezählt.

4.3.2 Punktaddition

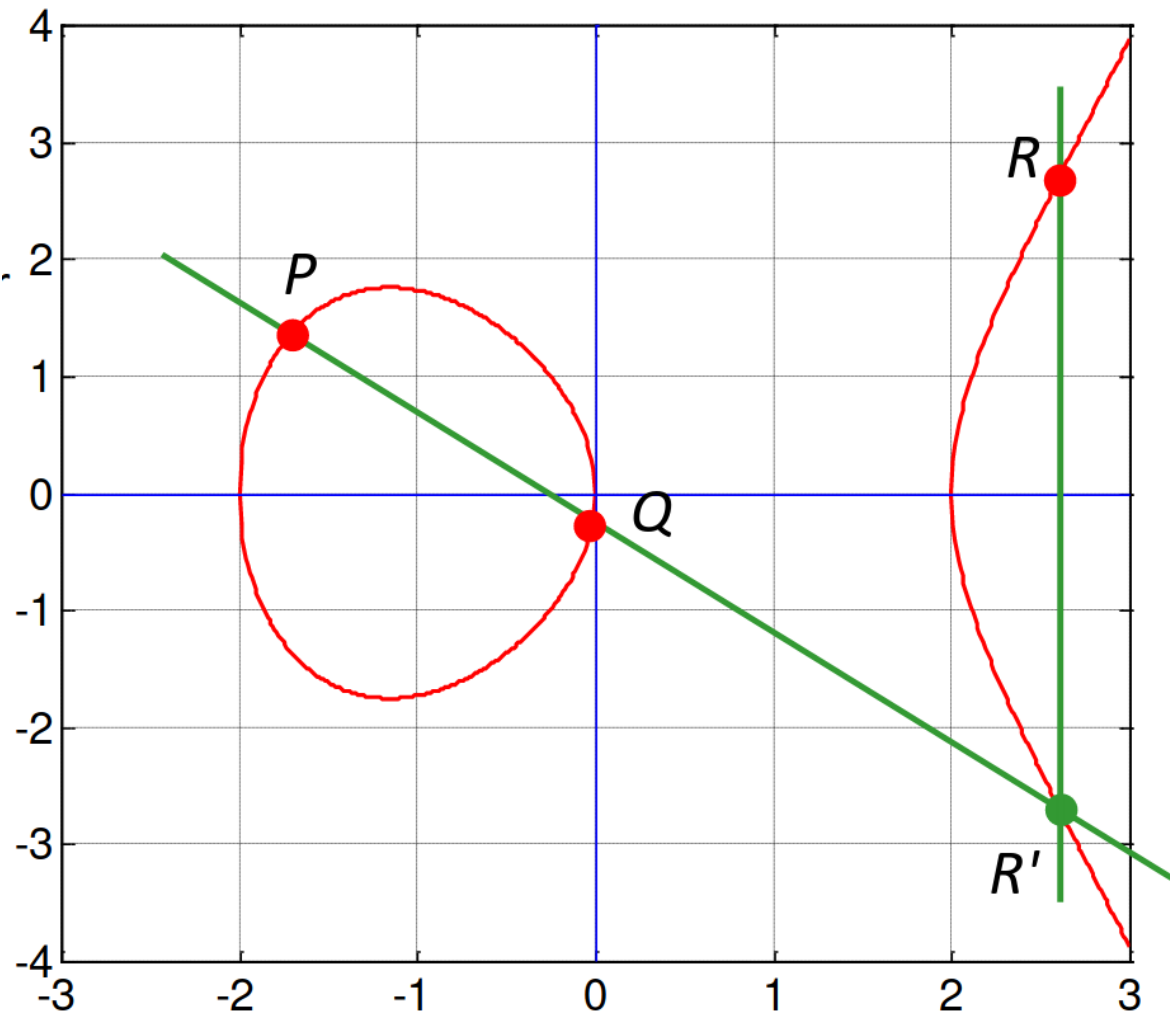


Abbildung 5: Elliptic Curve Punkt Addition

$$P + Q = R$$

4.3.3 Neutrales und Inverses Element

Inverses Element resp. Spiegelung an der x-Achse:

$$P'(x, -y) = -P(x, y)$$

Punktaddition:

$$P' + P = P + P' = \mathcal{O}$$

Das neutrale Element ist der Punkt im unendlichen, also

$$\mathcal{O}(x, \infty)$$

Dann einfach noch die einzelnen Koordinaten mit p modulo rechnen.

Nun das Ganze mit $\text{mod } p$, p eine Primzahl

Allgemeine Form:

$$y^2 \equiv x^3 + ax + b \pmod{p}$$

Nichtsingularitätsbedingung:

$$4a^3 + 27b^2 \not\equiv 0 \pmod{p}$$

Beispiel:

$$y^2 \equiv x^3 + 8x + 5 \pmod{11}$$

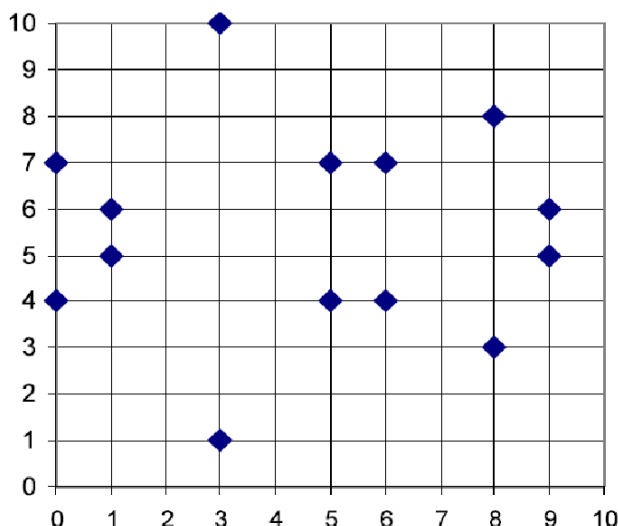


Abbildung 6: Elliptic Curve with Modulo

Ein Punkt P mit welchem man alle Punkte ausrechnen kann ist ein Basispunkt. Hier als Beispiel $(k \cdot)P(0, 4)$. Wenn wir P mit sich selbst addieren oder den Faktor k vorndran 1,2,3 ... ausrechnen, erhalten wir alle Punkte der Kurve.

Die Koordinaten muss man dabei jeweils mit p modulo rechnen.

Das k vor dem P muss dann modulo die Gruppenordnung (wie viele verschiedene Punkte dass es gibt) gerechnet werden. Da sich die Punkte wiederholen.

4.3.4 Allgemeine Form & Zusammenfassung:

Remember:

- allgemeine Form: $E = \{(x; y) | y^2 = x^3 + ax + b \mod p\}$
- Nichtsingularitätsbedingung: $4a^3 + 27b^2 \neq 0$
- E = Zusammenfassung aller Punkte, welche die Bedingung erfüllen
- p muss zwischen 256 und 512 Bit sein
- a & b können beliebig sein, müssen aber die beiden Gesetze erfüllen
- Eine Gruppe heist zyklisch wenn es einen erzeugenden Punkt P gibt.
- Mit einen erzeugenden Punkt P kann man alle anderen Punkte erzeugen $k \cdot P$
- ist die Gruppenordnung prim \rightarrow alle Elemente ausser \mathcal{O} sind erzeugend

Bereich der Punkte kann berechnet werden mit:

$$p + 1 - 2\sqrt{p} \leq |E| \leq p + 1 + 2\sqrt{p}$$

Die effektive Range wird durch a und b bestimmt, mit p kann nur die Randzahlen berechnet werden.

4.3.5 Formel zur Addition von 2 Punkten

Zuerst Steigung berechnen, danach können x und y berechnet werden.

$$\begin{aligned}s &\equiv \frac{y_2 - y_1}{x_2 - x_1} \mod p \\x_3 &\equiv s^2 - x_1 - x_2 \mod p \\y_3 &\equiv s(x_1 - x_3) - y_1 \mod p\end{aligned}$$

Falls man $P + P$ (Verdoppelung) rechnen muss, resp wenn $x_1 = x_2; y_1 = y_2$ dann ändert sich die Steigungsformel zu:

$$s \equiv \frac{3 \cdot x_1^2 + a}{2 \cdot y_1} \pmod{p}$$

Remember: bei $x \cdot y \pmod{p} \equiv (x \pmod{p} \cdot y \pmod{p}) \pmod{p}$

Beispiel mit $P(0; 4)$ und $Q(3; 1)$ in Kurve $y^2 \equiv x^3 + 8x + 5 \pmod{11}$:

$$\begin{aligned} s &\equiv \frac{1 - 4}{3 - 0} \pmod{11} \\ &\equiv \frac{-3}{3} \pmod{11} \\ &\equiv (-3) \cdot 3^{-1} \pmod{11} \\ &\equiv [(-3) \pmod{11} \cdot 3^{-1} \pmod{11}] \pmod{11} \\ &\equiv 8 \cdot 4 \pmod{11} \\ &\equiv 32 \pmod{11} = 10 \end{aligned}$$

Danach Punkte:

$$\begin{aligned} x_3 &\equiv 10^2 - 0 - 3 \pmod{11} = 9 \\ y_3 &\equiv 10 \cdot (0 - 9) - 4 \pmod{11} = 5 \end{aligned}$$

4.3.6 Double and add Algorithmus

Analog kann zum Square and Multiply Algorithmus effizient gerechnet werden.

Beispiel: $28 \cdot P$

- $28 = 16 + 8 + 4 = 2^4 + 2^3 + 2^2 = 11100$
- Mit der ersten 1 macht man nichts
- wegen 1: $p \rightarrow 2p \rightarrow 2p + p = 3p$ | double and add
- wegen 1: $3p \rightarrow 6p \rightarrow 6p + p = 7p$ | double and add
- wegen 0: $7p \rightarrow 14p$ | double
- wegen 0: $14p \rightarrow 28p$ | double

4.3.7 Bestimmung aller Punkte

A) Bestimmung aller Punkte

x	0	1	2	3	4	5	6
$x^3 + 3x + 2 \bmod 7$	2	5	2	3	1	2	5

y	0	1	2	3	4	5	6
$y^2 \bmod 7$	0	1	4	2	2	4	1

Abbildung 7: Bestimmung aller Punkte

Für $x = 0$: (0; 3) und (0; 4)
 Für $x = 1$: keine da $y^2 \bmod 7 \neq 5$
 Für $x = 2$: (2; 3) und (2; 4)
 Für $x = 3$: keine
 Für $x = 4$: (4; 1) und (4; 6)
 Für $x = 5$: (5; 3) und (5; 4)
 Für $x = 6$: keine

4.3.8 Bestimmung ob Punkt auf Graf ist

Beispiel:

Kurve: $E : y^2 \equiv x^3 + x + 1$ über \mathbb{Z}_{19} Punkt: $Q(15, 13)$

1. Punkt in Gleichung einsetzen
2. Links von Gleichung ausrechnen $\bmod p$
3. Rechts von Gleichung ausrechnen $\bmod p$
4. Vergleichen

$$y^2 \equiv x^3 + x + 1 \bmod 19 \quad (1)$$

$$16^2 \equiv 15^3 + 15 + 1 \bmod 19 \quad (2)$$

$$16^2 \equiv 256 \bmod 19 \equiv 9 \bmod 19 \text{ und } \quad (3)$$

$$15^3 + 15 + 1 \equiv 3391 \bmod 19 \equiv 9 \bmod 19 \quad (4)$$

Da beide Seiten gleich sind, ist der Punkt auf der Kurve.

4.4 RSA

RSA braucht mehr Rechenleistung (ist sicherer?).

RSA Keys sind grösser als ECC Keys für gleiche Sicherheit.

Ablauf beim Signieren:

1. Hashwert berechnen (Integrität)
2. Hashwert mit privatem Schlüssel signieren (Authentizität)

4.4.1 Anzahl der Primzahlen 1-x

Ist eine Annäherungsformel, berechnet die Anzahl der Primzahlen in 1 bis x berechnen:

$$\pi(x) \approx \frac{x}{\ln(x)}$$

Nur ungerade Primzahlen von 1 bis x :

$$APZ(x) = \frac{Anz.PZ}{Anz.ungeradeZ.} \approx \frac{\frac{x}{\ln(x)}}{\frac{x}{2}} = \frac{2}{\ln(x)}$$

Beispiel:

$$\pi(10^{150}) \approx \frac{10^{150}}{\ln(10^{150})} = \frac{10^{150}}{150 \cdot \ln(10)} \approx \frac{10^{150}}{345} \approx \frac{10^{150}}{3.5 \cdot 10^2} = \frac{10 \cdot 10^{149}}{3.5 \cdot 10^2} = \frac{10 \cdot 10^{147}}{3.5} \approx 2.9 \cdot 10^{147} APZ(10^{150})$$

4.4.2 Anzahl n-stelligen Primzahlen

Ist die genaue Formel, nicht unbedingt nötig, Annäherungsformel ist genau genug.

Anzahl der n-stelligen Primzahlen:

$$\pi(10^n) - \pi(10^{n-1})$$

Anteil der n-stelligen PZ. in den n-stelligen ungeraden Zahlen:

$$APZ(n) = \frac{Anz.PZ}{Anz.ung.Z.} \approx \frac{\pi(10^n) - \pi(10^{n-1})}{\frac{0.9 \cdot 10^n}{2}} = \frac{\pi(10^n) - \pi(10^{n-1})}{0.45 \cdot 10^n}$$

Beispiel:

$$n = 150 \Rightarrow x = 10^{150}$$

$$\pi(10^{150}) - \pi(10^{149}) = \frac{10^{150}}{\ln(10^{150})} - \frac{10^{149}}{\ln(10^{149})} \approx 2.6 \cdot 10^{147}$$

4.4.3 Dezimalstellen ausrechnen:

Bits: $N = 1024$

$$D = N \cdot \log_2(10)$$

Dezimalstellen: $10^D \approx 10^{308}$

4.5 Diskreter Logarithmus bei ECC

Diffie-Hellman: bei $z \equiv g^x \pmod{p}$ ist auf x schwer zu schliessen.

Sofern p mindestens 600 stellige Primzahl und g ein Generator ist.

Bei EC ist Gleichung $Q = k \cdot P$ schwer auf k zu schliessen.

4.5.1 Key exchange

Diffie-Hellman (K = Secret):

$$\begin{aligned} A &= g^a \pmod{p} \\ B &= g^b \pmod{p} \\ K &= A^b = B^a = g^{ab} \pmod{p} \end{aligned}$$

EC Key exchange (K = Secret):

$$\begin{aligned} A &= a_{\text{Alice}} \cdot P = a_A \cdot P \\ B &= b_{\text{Bob}} \cdot P = b_B \cdot P \\ K &= b_B \cdot A = (b_B \cdot a_A) \cdot P = a_A \cdot B = (a_A \cdot b_B) \cdot P \end{aligned}$$

4.6 Verschlüsselung mit EC nach Volker Müller

1. Schlüsselgenerierung (Bob ist Empfänger)

- a) Elliptische Kurve wählen
- b) Mit Primzahl ca. 256, 384 oder 512 Bit gross
- c) Koeffizienten a und b wählen
- d) Einem Punkt P der eine zyklische Untergruppe der Primordnung q erzeugt.
- e) d wählen wobei $0 < d < q$ und 512 Bit lang
- f) Berechne $Q = d \cdot P$
- g) Damit sind die Beiden Schlüssel erzeugt:

$$K_{pub} = (p, a, b, q, P, Q)$$

$$K_{priv} = d$$

2. Verschlüsselung (Sender Alice)

- a) Wähle i mit $1 < i < q - 1$
- b) Berechne Einmal-Key $K_E = i \cdot P$
- c) Berechne Masking-Key $K_M = i \cdot Q$
- d) Verschlüsse Meldung T mit $Y = T \oplus x$ -Wert von K_M
- e) Meldung mit Einmal Key schicken also (Y, K_E)

3. Entschlüsselung (Empfänger Bob)

- a) Masking-Key berechnen $K_M = d \cdot K_E$
- b) Meldung entschlüsseln $T = Y \oplus x$ -Wert von K_M

5 Blinde Signaturen

Generelle Beschreibung: Anna weiß nicht WAS sie unterschreibt, wenn sie das Dokument später sieht, weiß sie aber DASS sie es unterschrieben hat.

Nutzen:

- Unverfälschbarkeit - kein falsches Geld erzeugen
- Anonymität - gegenüber Bank
- Unlinkbarkeit - keinen Zusammenhang zwischen s und m

Beispiel-Ablauf:

1. Kunde zieht Geld ab m ist Identifikationsnummer
2. Kunde bildet m' und fragt Bank um Unterschrift
3. Bank rechnet s' aus m' und bucht Geld ab und schickt s' zurück (Signatur)
4. Kunde berechnet s aus s' für den Wert m
5. Kunde bezahlt im Shop mit m und s
6. Shop kann mit Public-Key und Signatur kontrollieren ob von Bank signiert
7. Shop liefert Geld mit ID m und Signatur s bei Bank ein.
8. Bank prüft ebenfalls Signatur.
9. Bank weiß nun, dass dieses Geld abgehoben wurde, aber nicht von wem.

Allgemeiner mathematischer Ablauf:

1. Kunde kenn Public-Key N und e
2. Bank kenn Private-Key d
3. Kunde: Wahl der Nachricht $m \rightarrow$ eindeutige Seriennummer
4. Kunde: Nachricht "blinden":
 - a) Zufällige wahl von $r \in_R \mathbb{Z}_N^*$
 - b) $N = p \cdot q$ mit p, q Primzahlen
 - c) $ggT(r, N) = 1$
 - d) $ggT(r, N) \neq p$
 - e) $ggT(r, N) \neq q$
 - f) Berechnung von $m' \equiv r^e \cdot m \pmod{N}$
5. Kunde: "geblindete" Nachricht m' schicken
6. Bank: Nachricht m' signieren:
 $s' \equiv (m')^d \pmod{N}$
7. Bank: Signatur s' schicken
8. Kunde: Berechnung der Signatur s :
 $s \equiv s' \cdot r^{-1} \pmod{N}$
9. Kunde: Überprüfen der Signatur:
 $m \equiv s^e \pmod{N} \rightarrow \text{ok?}$
10. Kunde: Bezahlen mit m und s
11. Shop und Bank überprüfen Echtheit der Münze nach Schritt 7.

5.1 Beweis der Korrektheit

$$\frac{s'}{r} \equiv \frac{(m')^d}{r} \equiv \frac{(r^e \cdot m)^d}{r} \equiv \frac{r^{ed} \cdot m^d}{r} \equiv \frac{r \cdot m^d}{r} \equiv m^d \equiv s \pmod{N}$$

5.2 Beispiel mit Zahlen

Parameter:

- Öffentlicher Schlüssel: $(N, e) = (91, 5)$
- Privater Schlüssel: $(p, q, d) = (7, 13, 29)$

Kontrolle der Parameter:

$$\varphi(N) = \varphi(7 \cdot 13) = \varphi(7) \cdot \varphi(13) = 6 \cdot 12 = 72 = (2 \cdot 2 \cdot 2 \cdot 3 \cdot 3)$$

$e = 5$ ist teilerfremd zu 72.

$$d = 5^{-1} \bmod 72 \equiv 29, \text{ da } 5 \cdot 29 \equiv 145 \equiv 2 \cdot 72 + 1 \equiv 1 \bmod 72$$

Sei $m = 11$ die Nachricht.

Wir wählen zufällig $r = 16$

es muss gelten: $ggT(N, r) = ggT(91, 16) = 1$

Erwartete Signatur:

$$s \equiv m^d \bmod N \equiv 11^{29} \bmod 91 = 72$$

Kunde kennt Public Key (667, 9)	Meldung	Bank kennt Secret Exponent d = 137
1. Wahl der Nachricht m: $m = 38$		
2. Nachricht m „blinden“: $r = 63$ $m' \equiv 63^9 \cdot 38 \bmod 667 \equiv 36$		
3. geblindete Nachricht m' schicken:	$m' = 36$ ----->	
		4. Nachricht m' signieren: $s' \equiv (36)^{137} \bmod 667 \equiv 487$
	$s' = 487$ <-----	5. Signatur s' zurückschicken:
6. Signatur s aus s' extrahieren: $s \equiv 487 \cdot 63^{-1} \equiv 487 \cdot 180 \equiv 283 \bmod 667$		
7. Signatur s prüfen: $\underbrace{m \equiv s^e \equiv 283^9 \equiv 38 \bmod 667}_{OK}$		
8. Mit Münze (11, 72) bezahlen:		
9. resp. 10 Signatur s prüfen: Shop wie Bank überprüfen die Echtheit der Münze, cf. Schritt 7.		

Abbildung 8: Blinde Signaturen

5.3 Basis-Test

- geheimer Exponent muss aufgeteilt werden
- Man kann ihn additiv oder multiplikativ aufteilen
- Kann beliebig additiv aufgeteilt werden
- Kann NICHT beliebig multiplikativ aufgeteilt werden:
Der erste Teil der Aufteilung muss teilerfremd zu $\varphi(N)$ sein, also $ggT(d_1; \varphi(N)) = 1$.
- Ablauf der Erstellung einer Doppelsignatur ist bei additiv und multiplikativ nicht dieselbe
Bei multiplikativ muss z.B. die Reihenfolge eingehalten werden.
- Blinde Signaturen werden z.B. für anonymes, digitales Geld verwendet.

6 Einführung in die Public-Key Infrastruktur (PKI)

6.1 Verschlüsseln und Signieren (repetition)

6.1.1 Verschlüsseln

6.1.2 Signieren

Ablauf signieren:

1. Dokument von Alice ist Ausgangswert
2. Hash berechnen → Hashwert
3. chiffrieren (mit private key und Hash) → Signatur
4. Dokument & Signatur + Zertifikat → signiertes Dokument

Achtung: Schlüsselverwendung bei Signatur ist umgekehrt wie bei Verschlüsselung. Private Key wird für die Erstellung der Signatur verwendet und Public-Key um zu validieren.

Warum Zertifikat? → um sicherzustellen, dass der öffentliche Schlüssel auch wirklich von Alice ist.

Ablauf Signatur prüfen:

1. Dokument von Alice ist Ausgangswert
2. Dokument entpacken (Signatur und Dokument)
3. Signatur mit öffentlichem Schlüssel entschlüsseln → Hashwert
4. Hashwert von Dokument berechnen
5. Hashes vergleichen
6. Zertifikat Überprüfen
7. Wenn alles ok dann ist Signatur gültig.

6.2 Zertifikate

6.2.1 Herstellung eines Zertifikats

Allgemeiner Ablauf:

- Antragssteller identifiziert sich bei CA
- Aus dessen Informationen wird Datensatz gebildet (Zertifikatsinhalt)
- Datensatz wird mit privatem Schlüssel von CA signiert → Zertifikat
- CA veröffentlicht Zertifikat

Technisches vorgehen:

1. Zertifikatsinhalt
 - Version
 - Serial Number
 - Subject
 - Public Key
2. Inhalt hashen
3. Hash signieren
4. Signierter Hash + Zertifikatsinhalt → Zertifikat

6.2.2 Installation eines neues (Root-)Zertifikates

1. Root-CA-Zertifikat Echtheit überprüfen
2. Echtheitsprüfung wird mit Fingerprint gemacht
3. Lokal angezeigter Fingerprint wird mit vertrauenswürdiger Referenzquelle verglichen

6.2.3 Überprüfung der Echtheit eines Zertifikates vom Betriebssystem

1. Applikation überprüft Signatur auf Zertifikat mithilfe des Root-CA-Zertifikates.
2. Zertifizierungsstelle muss im System hinterlegt sein; sie wird zum Trust Anchor

6.2.4 Zertifikatsklassen

- **Klasse 1:** wenig Sicherheit, keine Identitätsprüfung
- **Klasse 2:** mittlere Sicherheit, schwache Identitätsprüfung
- **Klasse 3:** hohe Sicherheit, strenge Identitätsprüfung
- **Qualified Certificate:** höchste Stufe, werden nur für natürliche Personen ausgestellt

7 Protokolle

7.1 User Authentication

- Username / Password
- One-Time Password
- Symmetric Algorithms
- Public-Key Algorithms
- Biometric Authentication

7.2 False-rates

Es gibt zwei Arten von False-rates:

- False Acceptance Rate (FAR)
- False Rejection Rate (FRR)

Beide sollten so tief wie mögliche sein. Aber es ist ein Tradeoff zwischen den beiden. Senkt man die Eine erhöht sich die Andere.

7.3 Verifikationen

7.3.1 One to many

Handy: überprüfe ob ich derjenige bin, der ich vorzugeben behaupte.

7.3.2 Many to one

Bank: überprüfe ob ich derjenige bin, der ich vorzugeben behaupte.

7.4 Parallelsession Attacke

Zwei Sessions eröffnen, dann muss nichts gerechnet werden und Zufalls-/Chiffrierzahl kann kopiert werden.

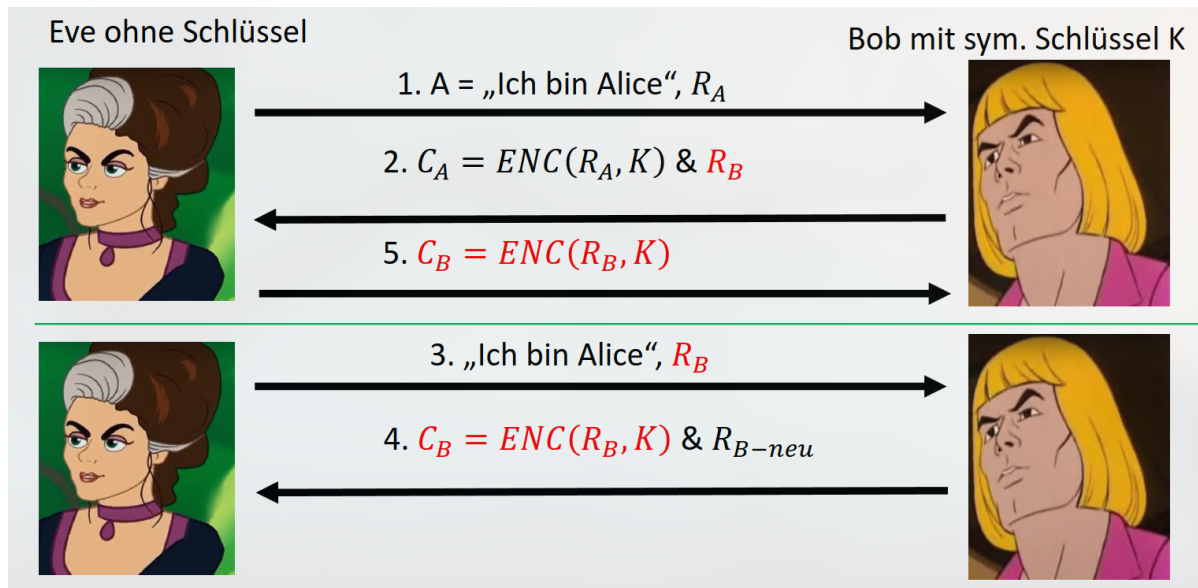


Abbildung 9: Parallelsession Attacke

8 Quantenkryptographie

Quantenkryptographie und Quantencomputer sind zwei verschiedene Dinge.

8.1 Polarization

4 Zustände, jedem muss 1 oder 0 zugewiesen werden.

- Senkrecht
- Wagrecht
- Schräg links
- Schräg rechts

Darauf basierend können Filter kreiert werden. Sollte filter nicht auf Teilchen passen, ist es 50/50 in welchem Zustand es durchgelassen wird.

8.2 Quantum Key Exchange

1. Sender Alice wählt zufällige Bits
2. Alice sendet Photonen mit gewählter Polarization
3. Empfänger Bob wählt zufällige Filter
4. Bob erhält gefilterte Photonen
5. Bob kennt die Kodierung und ordnet 1 oder 0 zu
6. Bob sendet zurück, welche Filter er benutzt hat
7. Alice sagt, welche Filter korrekt waren

Im statistischen Mittel werden in 50% die falschen Filter gewählt. Zudem werden die Hälfte davon aufgedeckt um zu prüfen ob man abgehört wurde. Es müssen also ca. 4 mal so viele geschickt werden.

Key-takeaway: immer die Bits verwenden, welche durch den Filter nicht verändert werden. Alle anderen haben einen nicht vorhersagbaren Zustand.

Aufgabe 1**7 Punkte****Aufgabe 1.1**

Sie setzen einen 128-Bit Blockchiffre-Algorithmus ein. Nun wird eine neue Version veröffentlicht; diese Version hat ebenfalls eine Schlüsselgrösse von 128 Bit, aber anstatt 128 Bit Input- und Outputgrösse hat sie neu 192 Bit Input- und Outputgrösse.

- a) [1 P.] Welche der zwei Brute-Force Attacken wird/werden nun aufwändiger?
- b) [3 P.] Um welchen Faktor wird/werden die Attacke(n) nun schwieriger?

Lösung:

- a) Es wird nur der Table Look up Angriff erschwert.
- b) Speicherbedarf allgemein: Sei n = Anzahl Schlüsselbits und m = die Anzahl der Input- resp. Outputbits

$$\text{Speicherplatz} = m \cdot 2^n$$

Somit ist der Erschwernisfaktor $= \frac{192 \cdot 2^{128}}{128 \cdot 2^{128}} = 1,5$; oder ein zusätzlicher Speicheraufwand von 50%.

Aufgabe 1.2

[3 P.] Ein CH-Sicherheitsexperte hat in einem Zeitungsinterview den folgenden Ratschlag zur Verschlüsselung von Bankdaten formuliert:

„.... die Bankdaten sollten in einzelne Gruppen zusammengefasst sein, die jede mit einem anderen Verfahren und einem unterschiedlichen digitalen Schlüssel verschlüsselt werden. Diese verschlüsselten Gruppen sollten dann komprimiert und nochmals verschlüsselt werden, wiederum mit einem anderen Verschlüsselungsverfahren und einem weiteren digitalen Schlüssel.....“.

Kommentieren Sie diesen Ratschlag.

Lösung:





- 1) Es gibt keinen Grund Gruppen zu bilden und mit verschiedenen Schlüsseln zu arbeiten.
- 2) Es ist ziemlich nutzlos, verschlüsselte Daten zu komprimieren, es gibt weder eine grössere Sicherheit, noch spart man Platz. (Bemerkung: Einzig vor dem Verschlüsseln würde eine Komprimierung eine Platzersparnis bringen).
- 3) Kaskaden von Verschlüsselungen sind mit AES nicht mehr nötig.
- 4) Key Management wird komplizierter.

Bewertungshinweis:

























Es genügen 3 von diesen 4 Argumenten.

Aufgabe 2**7 Punkte**

- a) [4 P.] Welche Bits sind nach dem untenstehenden Austausch bei Alice und Bob identisch?
Alice und Bob machen folgende Codierung miteinander ab:

			
0	1	1	0

Alice schickt folgende Sequenz und wählt dabei die untenstehenden Filter:

Zufälliges Photon								
Polarisierung von Alice								
Zwischenlinie für eigene Notizen.	1	1	0	1	1	0	1	0
Bobs Wahl der Filter								
Zwischenlinie für eigene Notizen.								
Gemeinsamer Schlüssel von Alice und Bob	-	1	-	1	1	0	-	0

Bewertungshinweis:

➤ Pro falsche Position, 1½ Pte Abzug.

- b) [3 P.] Sie müssen für einen SHA-384 einen 384 Bit Schlüssel mit dem obigen Verfahren austauschen. Erklären Sie wie viele Schlüsselbit Sie austauschen müssen, und wie, warum und wie viele Sie „verlieren“.

Lösung:

[1 P.] Ca. Faktor 4 Mal mehr, also ca. 1536 Bit.

[1 P.] Ca. die Hälfte verliert man, weil der falsche Filter verwendet wurde. Somit bleiben noch 768 übrig.

[1 P.] Davon braucht man die Hälfte, die man aufdeckt, um zu bemerken, ob man abgehört wurde.

Aufgabe 3**5 Punkte**

Folgendes Protokoll wird publiziert.

Abkürzungen:

ID_A = Identifikationsnummer von A (z.B. IP-Adresse).

ID_B = Identifikationsnummer von B (z.B. IP-Adresse).

N_A = 256 Bit Zufallszahl von A generiert.

N_B = 256 Zufallszahl von B generiert.

$Hash_{AB}$ = Hashwert mit SHA-256 über die Inhalte in der Klammer von A (für B).

$Hash_{BA}$ = Hashwert mit SHA-256 über die Inhalte in der Klammer von B (für A).

NR	Teilnehmer A	Meldung	Teilnehmer B
1)	Generiert zufällig N_A		
2)		ID_A, N_A ----->	
3)			Generiert zufällig N_B und schickt $Hash_{BA}(..)$
4)		$N_B, Hash_{BA}(N_A, N_B, ID_B)$ -----<	
5)	Verifiziert $Hash_{BA}(..)$ und gene- riert und schickt $Hash_{AB}(..)$.		
6)		$Hash_{AB}(N_A, N_B)$ ----->	
7)			Verifiziert $Hash_{AB}(..)$

a) [2 P.] Für was **sollte** dieses Verfahren am ehesten dienen? Kreuzen Sie entsprechend an. Falsches Ankreuzen gibt **Punktabzug**, die Summe kann aber nicht negativ werden.

- ☐ Einseitige Authentisierung gewähren
- ☐ Vertraulichkeit/Geheimhaltung gewähren
- ☐ Non repudiation of receipt erreichen
- ☐ Integrität gewähren
- ☐ Nicht-Abstreitbarkeit des Ursprungs erreichen
- ☒ Gegenseitige Authentisierung erwirken

b) [3 P.] Kommentieren Sie die Wirksamkeit gegen den/die geplanten Angriffe.

Lösung:

Im Hash ist kein Geheimnis drin, somit kann alles auch von Eve berechnet werden und ist somit absolut sinnlos.

Aufgabe 4**8 Punkte**

Eine Codebook Analyse hat nicht das Auffinden des unbekannten Schlüssels zum Ziel, sondern die direkte Rekonstruktion des Klartextes.

Der Ablauf geht wie folgt:

- Man berechnet eine (unter Umständen grosse) Anzahl von Chiffretextblöcken aus „sinnvollen“ Klartextblöcken, unter Anwendung des unbekannten Schlüssels.
- Speicherung der Klartext/Chiffretext-Paare (Aufbau des Codebuchs).
- Vergleich des abgehörten Chiffretextes mit den gespeicherten Paaren.

a) [2 P.] Um was für einen Typ von Attacke handelt es sich hier? Kreuzen Sie entsprechend an. Falsches Ankreuzen gibt **Punktabzug**, die Summe kann aber nicht negativ werden.

- ☐ Ciphertext-only Attacke
☐ Known-plaintext Attacke
☒ Chosen-plaintext Attacke
☐ Chosen-ciphertext Attacke

b) [6 P.] Gegeben ist ein 64-Bit PIN-Block der folgenden Form.

	PIN-Länge	PIN	Padding
1	6	PPPPPP	32 Zufallsbit

- Im ersten Halbbyte steht fix eine „1“
- Im zweiten Halbbyte steht fix eine „6“
- Im dritten bis achten Halbbyte stehen je eine Ziffer von 0, ..., 9
- In den letzten 8 Halbbytes stehen insgesamt 32 Zufallsbits.

Geben Sie die Grösse des Codebooks in Anzahl Harddisks von 10 TerraByte an.

Lösung:

Die Entropie der ersten zwei Halbbytes ist je Null: 1 P.

Die Entropie der 6 Ziffern ist $6 \cdot 3,3 \text{ Bits} = \text{ca. } 20 \text{ Bits}$. 1 P.

Die Entropie der 32 Zufallsbits beträgt 32 Bit. 1 P.

Somit müssen $2^{32+20} = 2^{52}$ Blöcke à 64 Bit gespeichert werden. 1 P.

$$\text{Anzahl Terabyte} = \frac{64 \cdot 2^{52}}{8 \cdot 2^{40}} = 8 \cdot 2^{12} = 2^{15} = 32'768 \quad 1 \text{ P.}$$

Resultat: Somit braucht es ca. 3'300 HD à 10 TByte. 1 P.

Aufgabe 5**5 Punkte**

Gegeben sind einige Algorithmen (Alg). Kreuzen Sie die richtige(n) Antwort(en) an. In der Spalte „Verfahren“ ist nur eine Antwort anzukreuzen. In der Spalte „Geeignet...“ sind mehrere Antworten möglich. Falsch angekreuzte Aussagen ergeben einen Punkteabzug, die Summe kann aber nicht negativ werden.

Alg.	Verfahren	Geeignet
RSA	<input type="checkbox"/> Symmetrisches <input checked="" type="checkbox"/> Asymmetrisches <input type="checkbox"/> Weder noch	<input type="checkbox"/> <u>nur</u> um Schlüssel auszutauschen <input type="checkbox"/> zum Berechnen eines CBC-MAC <input type="checkbox"/> <u>nur</u> zur gegenseitigen Authentisierung <input checked="" type="checkbox"/> zum Berechnen einer digitalen Signatur <input checked="" type="checkbox"/> zum Verschlüsseln <input type="checkbox"/> passt nicht in dieses Schema
AES	<input checked="" type="checkbox"/> Symmetrisches <input type="checkbox"/> Asymmetrisches <input type="checkbox"/> Weder noch	<input type="checkbox"/> <u>nur</u> um Schlüssel auszutauschen <input checked="" type="checkbox"/> zum Berechnen eines CBC-MAC <input type="checkbox"/> <u>nur</u> zur gegenseitigen Authentisierung <input type="checkbox"/> zum Berechnen einer digitalen Signatur <input checked="" type="checkbox"/> zum Verschlüsseln <input type="checkbox"/> passt nicht in dieses Schema
Diffie-Hellman	<input type="checkbox"/> Symmetrisches <input checked="" type="checkbox"/> Asymmetrisches <input type="checkbox"/> Weder noch	<input checked="" type="checkbox"/> <u>nur</u> um Schlüssel auszutauschen <input type="checkbox"/> zum Berechnen eines CBC-MAC <input type="checkbox"/> <u>nur</u> zur gegenseitigen Authentisierung <input type="checkbox"/> zum Berechnen einer digitalen Signatur <input type="checkbox"/> zum Verschlüsseln <input type="checkbox"/> passt nicht in dieses Schema
SHA-1	<input type="checkbox"/> Symmetrisches <input type="checkbox"/> Asymmetrisch <input checked="" type="checkbox"/> Weder noch	<input type="checkbox"/> <u>nur</u> um Schlüssel auszutauschen <input type="checkbox"/> zum Berechnen eines CBC-MAC <input type="checkbox"/> <u>nur</u> zur gegenseitigen Authentisierung <input type="checkbox"/> zum Berechnen einer digitalen Signatur <input type="checkbox"/> zum Verschlüsseln <input checked="" type="checkbox"/> passt nicht in dieses Schema

Bewertungshinweis:

- Pro richtiges Kreuz ½ Punkte, pro falsches ¼ Punkte Abzug.

Aufgabe 6**7 Punkte**

Um eine Doppelunterschrift zu implementieren wird der geheime Exponent $d = 59$ und der dazugehörige öffentliche Schlüssel ($e = 11$, $N = 91$) erzeugt.

Der aufzuteilende Exponent wird in die zwei Teile $T_1 = 126$ und $T_2 = x$ additiv aufgeteilt.

Die zu signierende Nachricht $m = 12345678$, die Hashfunktion sei die Quersumme der Nachricht mod 10.

Wie lautet die Signatur mit dem Exponenten T_2 , und wie lautet die vollständige Signatur der Nachricht m resp. $h(m)$, wenn die Signatur mit dem Exponenten T_1 den Wert 64 hat?

Lösung:

Berechnung von $\varphi(N) = \varphi(91) = \varphi(7 \cdot 13) = (7-1)(13-1) = 6 \cdot 12 = 72$

1 P.

Berechnung von $T_2 = x$: $d = T_1 + T_2 \bmod \varphi(N)$, also $59 = 126 + x \bmod 72$, also $x = 59 - 126 \bmod 72 = -67 \bmod 72 = 5$.

3 P.

Berechnung von $h(m)$: Quersumme von $m = 36$, somit $h(m) = 36 \bmod 10 = 6$.

1 P.

Berechnung der zweiten Hälfte der Signatur: $(h(m))^{T_2} \bmod N = 6^5 \bmod 91 = 41$

1 P.

Berechnung der Signatur von $h(m) = 64 \cdot 41 \bmod 91 = 76$

1 P.

Resultat: Die Signatur lautet $s = 76$

Bemerkung:

- Die Verifikation der Signatur: $s^e \bmod 91 = 76^{11} \bmod 91 = 6$
- Die direkte Berechnung der Signatur lautet: $6^{59} \bmod 91 = 76$

Bewertungshinweis:

Die direkte Berechnung der Signatur mit einem Rechner gibt 1 Punkt.

Aufgabe 7**14 Punkte**

Gegeben ist die elliptische Kurve $E: y^2 \equiv x^3 + x + 1$ über \mathbb{Z}_{19}

- [2 P.] Überprüfen Sie, ob die Kurve eine elliptische Kurve ist.
- [2 P.] Liegt der Punkt $P(15; 16)$ auf der Kurve?
- [7 P.] Der Punkt $Q(5; 13)$ liegt auf der Kurve. Berechnen Sie nun die Koordinaten des Punktes $S = 3 \cdot Q$
- [1 P.] Der Punkt $T(7; 3)$ liegt auf der Kurve. Bestimmen Sie die Koordinaten des Punktes $U = -T$.
- [2 P.] Die gegebene elliptische Kurve hat 21 Punkte. Überprüfen Sie, ob das stimmen kann.

Falls es Ihnen hilft, dürfen Sie die Kehrwerttabelle mod 19 im Folgenden verwenden.

x	1	2	3	4	5	6	7	8	9	10
$x^{-1} \bmod 19$	1	10	13	5	4	16	11	12	17	2

x	11	12	13	14	15	16	17	18
$x^{-1} \bmod 19$	7	8	3	15	14	6	9	18

Lösungen

- a) Die Nichtsingularitätsbedingung für $y^2 \equiv x^3 + a \cdot x + b \bmod p$ lautet:
 $4a^3 + 27b^2 \not\equiv 0 \bmod p$ ½ P.
 Für $a = 1$ und $b = 1$ bedeutet das: $4 \cdot 1^3 + 27 \cdot 1^2 = 31 \equiv 12 \bmod 19 \not\equiv 0 \bmod 19$ 1½ P.

- b) $P(15; 16)$ in die Gleichung einsetzen.

Der Punkt $P(15; 16)$ in $y^2 \equiv x^3 + x + 1 \bmod 19$ eingesetzt: $16^2 \equiv 15^3 + 1 \cdot 15 + 1 \bmod 19$

$$16^2 \equiv 256 \bmod 19 \equiv 9 \bmod 19 \text{ und } 15^3 + 1 \cdot 15 + 1 \equiv 3391 \bmod 19 \equiv 9 \bmod 19$$

1 P.

Resultat: Der Punkt $P(15; 16)$ liegt demnach auf der Kurve. 1 P.

- c) Koordinaten von Punkt $S = 3 \cdot Q = 2 \cdot Q + Q$. 1 P.

Für $2 \cdot Q(5; 13)$ gilt:

$$s \equiv \frac{3 \cdot x_1^2 + a}{2 \cdot y_1} \bmod p \equiv \frac{3 \cdot 5^2 + 1}{2 \cdot 13} \bmod 19 \equiv \frac{76}{26} \bmod 19 \equiv 76 \cdot 26^{-1} \bmod 19$$

$$\equiv (75 \bmod 19 \cdot 26^{-1} \bmod 19) \bmod 19 \equiv (0 \cdot (26^{-1} \bmod 19)) \bmod 19 \equiv 0$$

$$x_3 \equiv s^2 - x_1 - x_2 \bmod p \equiv 0^2 - 5 - 5 \bmod 19 \equiv -10 \bmod 19 = 9$$

$$y_3 \equiv s(x_1 - x_3) - y_1 \bmod p \equiv 0(5 - 9) - 13 \bmod 19 \equiv -13 \bmod 19 = 6$$

Also: $2 \cdot (5; 13) = (9; 6)$ 3 P.

Für $2 \cdot Q(5; 13) + Q(5; 13) = (9; 6) + (5; 13)$ gilt:

Detailberechnungen:

$$s \equiv \frac{y_2 - y_1}{x_2 - x_1} \mod p \equiv \frac{13 - 6}{5 - 9} \mod 19 \equiv \frac{7}{-4} \mod 19 \equiv \frac{7}{15} \mod 19 \equiv$$

$$\equiv (7 \cdot 15^{-1}) \mod 19 \equiv (7 \cdot 14) \mod 19 \equiv 98 \mod 19 = 3$$

$$x_3 \equiv s^2 - x_1 - x_2 \mod p \equiv 3^2 - 9 - 5 \mod 19 \equiv -5 \mod 19 \equiv 14$$

$$y_3 \equiv s(x_1 - x_3) - y_1 \mod p \equiv 3(9 - 14) - 6 \mod 19 \equiv -21 \mod 19 = 17$$

Resultat: $3Q(5; 13) = (14; 17)$

3 P.

d) $(7; 3) \rightarrow U = -T(7; -3)$

$$U(7; -3) = U(7; -3 \mod 19) = U(7; 16)$$

e) Mit dem Theorem von Hasse, kann die Anzahl der Kurvenpunkte abgeschätzt werden:

$$p + 1 - 2\sqrt{p} \leq |E| \leq p + 1 + 2\sqrt{p}$$

Für $p = 19$ ist das somit: $19 + 1 - 2\sqrt{19} \leq |E| \leq 19 + 1 + 2\sqrt{19}$, also: $12 \leq |E| \leq 28$

Resultat:

Eine Kurve mit mod 19 kann zwischen 12 und 28 Punkte haben, somit kann diese Anzahl (21 Kurvenpunkte) stimmen.

Bewertungshinweise:

- Rechenfehler werden mit 1 Punkt Abzug bewertet.
- Fehlende Interpretation, ½ P. Abzug.

Aufgabe 8:**4 Punkte**

Die Erstellung von digitalen Signaturen mithilfe von RSA erfolgt in zwei Schritten (=mathematische Operationen). Nennen Sie diese beiden Schritte (je 1 Punkt) und benennen Sie das Sicherheitsziel, das durch die Anwendung des jeweiligen Schrittes erreicht wird (je 1 Punkt).

	Mathematische Operation	Sicherheitsziel
1. Schritt		
2. Schritt		

Lösung:

	Mathematische Operation	Sicherheitsziel
1. Schritt	Berechnung des Hashwerts der Daten	Integrität
2. Schritt	Signatur des Hashwerts mit dem privaten Schlüssel des Signierers	Authentizität

Aufgabe 9:**3 Punkte**

Alice und Bob möchten untereinander verschlüsselte und signierte E-Mails austauschen. Beide haben sich deshalb Zertifikate einer öffentlichen Zertifizierungsstelle (z.B. SwissSign AG, QuoVadis Trustlink Schweiz AG etc.) beschafft, die sie nun einmalig gegenseitig als E-Mail-Anhang austauschen wollen. Es bieten sich ihnen hierzu 4 E-Mail-Versandoptionen entsprechend den möglichen Sicherheitszielen an. Wählen Sie die *minimal notwendige* Versandoption (1 Punkt) und begründen Sie, warum diese E-Mail-Versandoption ausreicht. (2 Punkte).

Versandoption	
«gewöhnliche» E-Mail	
Verschlüsselt	
Signiert	
Signiert und verschlüsselt	

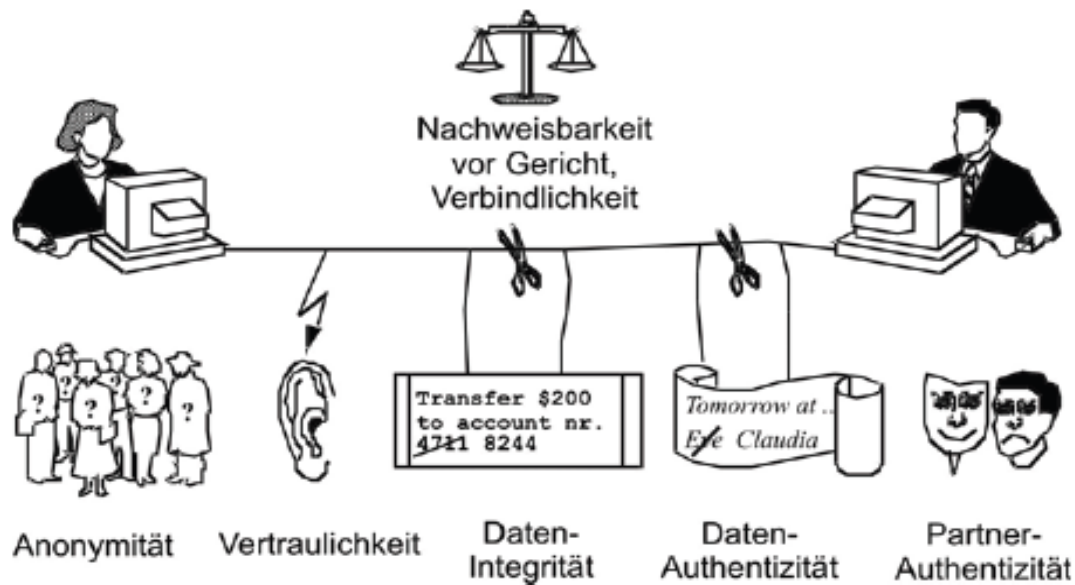
Lösung:

Versandoption	
«gewöhnliche» E-Mail	x
Verschlüsselt	
Signiert	
Signiert und verschlüsselt	

Begründung: Zertifikate müssen nicht verschlüsselt verteilt werden, da sie öffentliche Schlüssel beinhalten und somit selber auch öffentlich sind (1 Punkt). Zertifikate müssen zudem auch nicht signiert verteilt werden, da sie von der Zertifizierungsstelle signiert sind, wodurch deren Authentizität sichergestellt ist (1 Punkt).

Aufgabe 1:**8 Punkte**

In einem Buch werden die folgenden Sicherheitsdienste in einer Zeichnung dargestellt. Leider wurden die Kryptographischen Mechanismen nicht mit eingezeichnet.

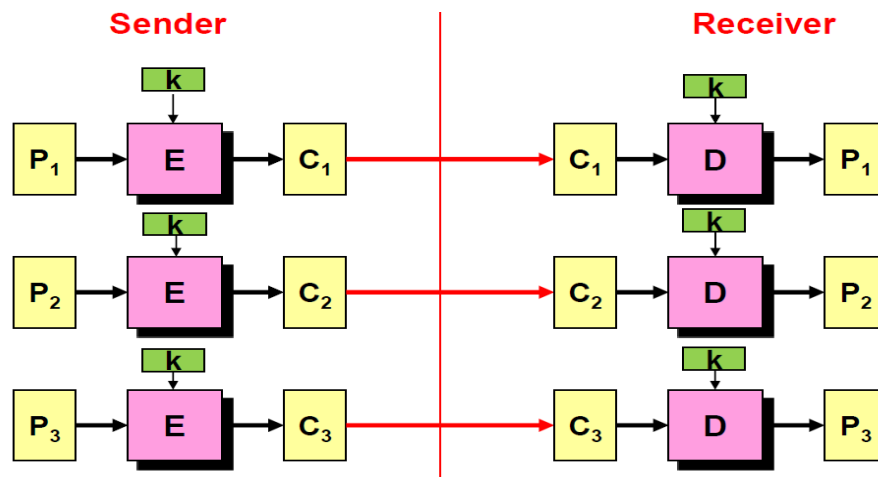


[8 P.] Kreuzen Sie alle korrekten Aussagen der Form „der gegebene Kryptographische Sicherheitsmechanismus kann Basis für den angekreuzten Sicherheitsdienst sein.“ **Falsches Ankreuzen wird mit Punktabzug versehen, die Summe kann nicht negativ werden.**

Sicherheitsdienst \ Krypt. Mechanismus	Vertraulichkeit	Datenintegrität	Partnerauthentizität	Keiner dieser Dienste
Sym. Verschlüsselung	X			
Asym. Verschlüsselung	X			
Diffie-Hellman Schlüsselaustausch Protokoll				X
Hybride Verschlüsselung	X			
MAC- Berechnung		X		
Digitale Signatur		X		
C-R Protokoll mit MAC			X	
C-R Protokoll mit digitaler Signatur			X	

Aufgabe 2**5 Punkte**

Im Folgenden ist auf der Senderseite die Verschlüsselung der Klartextblöcke P_1 , P_2 und P_3 mit dem Schlüssel k in die Chiffretextblöcke C_1 , C_2 und C_3 und beim Receiver die Entschlüsselung abgebildet.



Kreuzen Sie nun alle richtigen Antworten an. **Falsches Ankreuzen wird mit Punktabzug versehen.** Die Summe kann aber nicht negativ werden.

5 P.

NR	Aufgabe	Auswahl
a)	Bei diesem Verfahren handelt es sich um ein...	<input checked="" type="checkbox"/> ... symmetrisches Verfahren <input type="checkbox"/> ... asymmetrisches Verfahren <input type="checkbox"/> ... hybrides Verfahren <input type="checkbox"/> Keine der Angaben ist zutreffend.
b)	Es handelt sich dabei um eine ...	<input type="checkbox"/> ... Hashfunktion <input checked="" type="checkbox"/> ... Blockchiffre <input type="checkbox"/> ... Stromchiffre <input type="checkbox"/> ... Public Key Verfahren <input type="checkbox"/> Keine der Angaben ist zutreffend.
c)	Als Algorithmen kommen z.B. folgende infrage.	<input type="checkbox"/> Diffie-Hellman <input type="checkbox"/> ECC <input checked="" type="checkbox"/> 3-DES <input type="checkbox"/> RSA <input type="checkbox"/> Hashfunktionen wie SHA-1 <input type="checkbox"/> Elliptische Kurven <input checked="" type="checkbox"/> AES <input type="checkbox"/> Keine der Angaben ist zutreffend.
d)	Das abgebildete Verfahren zeigt den folgenden Modus.	<input type="checkbox"/> CBC <input type="checkbox"/> OFB <input type="checkbox"/> CTR <input checked="" type="checkbox"/> ECB <input type="checkbox"/> Keine der Angaben ist zutreffend.

Aufgabe 3**6 Punkte**

Sie wollen einen 3072 Bit RSA einsetzen.

- a) [3 P.] Wie viele Dezimalstellen müssen die zu wählenden Primzahlen haben?
 b) [3 P.] Angenommen Sie bräuchten 400-stellige Primzahlen, wie viele davon gibt es?

Lösung:

a) 3072 Bit bedeutet, dass $N = pq$ eine Zahl mit 3072 Bit ist, d.h. sie hat die Grösse 2^{3072}

$$2^{3072} = 2^{10 \cdot 307,2} = (2^{10})^{307,2} \approx (10^3)^{307,2} = 10^{3 \cdot 307,2} = 10^{921,6} \approx (10^{460})^2$$

1 P.

$$= 10^{460} \cdot 10^{460}$$

Somit muss p wie q je ungefähr 460-stellig sein.

2 P.

b) Ungefähr wie viele 400-stellige Primzahlen gibt es?

Die Eulersch'e Pi-Funktion $\pi(n) = \frac{n}{\ln(n)}$ gibt eine Abschätzung wie viele Primzahlen es von 1, bis n hat.

Somit gibt es ungefähr

$$\pi(n = 400) - \pi(n = 399) = \frac{10^{400}}{\ln(10^{400})} - \frac{10^{399}}{\ln(10^{399})} = \frac{10 \cdot 10^{399}}{400 \cdot \ln(10)} - \frac{10^{399}}{399 \cdot \ln(10)}$$

$$= 10^{399} \cdot \underbrace{\frac{1}{\ln(10)} \cdot \left(\frac{10}{400} - \frac{1}{399} \right)}_{\approx 1 \cdot 10^{-2}} \approx 10^{399} \cdot 10^{-2} = 10^{397}$$

Aufgabe 4**4 Punkte**

Sie berechnen d^{116} mittels Square and Multiply (SaM). Schreiben Sie detailliert die einzelnen Schritte auf. Es ist **nur die korrekte Reihenfolge** der resultierenden Exponenten aufzuschreiben.

Lösung:

$$116_{10} = 1110100_2$$





1 P.

Im Detail:

























Zahl											
1	Nichts										
1		2	3								
1				6	7						
0						14					
1							28	29			
0									58		
0										116	

Aufgabe 5**4 + 4 = 8 Punkte****Aufgabe 5.1****4 Punkte**

[4 P.] Welchen Schlüssel können Alice und Bob nach dem untenstehenden Austausch verwenden? Alice und Bob machen folgende Codierung miteinander ab:

			
1	0	1	0

Alice schickt folgende Sequenz und wählt dabei die untenstehenden Filter:

Zufälliges Photon								
Polarisierung von Alice								
Zwischenlinie für eigene Notizen.	0	0	1	0	1	1	0	1
Bobs Wahl der Filter								
Zwischenlinie für eigene Notizen.								
Gemeinsamer Schlüssel von Alice und Bob	-	0	1	0	-	1	-	-

Bewertungshinweis: Pro falsche Position, 1½ Pte Abzug.

Aufgabe 5.2**4 Punkte**

- a) [1 P.] Vergleichen Sie die klassische Sicherheit von einem 2048 Bit RSA und einer 384 Bit ECC.
- b) [3 P.] Vergleichen Sie die Sicherheit von einem 2048 Bit RSA und einer 384 Bit ECC, wenn es Quantencomputer mit genügend vielen Qubits gäbe.

Lösung:

- a) Ein 2048 Bit RSA hat in etwa die klassische Sicherheit eines 224 Bit ECC, resp. umgekehrt eine 384 Bit ECC hat in etwa die klassische Sicherheit eines 7680 Bit RSA.

b) Anzahl Qubit für das Faktorisieren: $K \approx 2k = 2 \cdot 2048 = 4096$ ½ P.

Anzahl Qubit für das ECC-Problem: 2 P.

$$K \approx 5k + 8\sqrt{k} + 5 \cdot \log_2 k = 5 \cdot 384 + 8 \cdot \sqrt{384} + 5 \cdot \log_2 384 \approx 2120$$

In Bezug auf die Quantencomputer ist ein 2048 Bit RSA sicherer als eine 384 Bit ECC. ½ P.

Aufgabe 6**6 Punkte**

Im Folgenden ist das Protokoll einer blinden Signatur mit dem RSA gegeben.
 Füllen Sie die offenen Stellen aus.

Werte: (i) $p = 3, q = 11 \Rightarrow N = pq = 33$ und $\varphi(N) = (p - 1)(q - 1) = 20$
 (ii) $e = 3$, und damit ist $d = e^{-1} \bmod \varphi(N) = 3^{-1} \bmod 20 = 7$.

Kunde kennt den Public Key (_____)	Meldung	Bank kennt Secret Key (_____)
<u>1. Wahl der Nachricht m:</u> $m = 2$		
<u>2. Nachricht m „blinden“:</u> $r = 5$ $m' \equiv$ _____		
<u>3. geblindete Nachricht m' schicken:</u>	$m' =$ _____ ----->	
<u>4. Nachricht m' signieren:</u>		$s' \equiv$ _____
<u>5. Signatur s' zurückschicken:</u>	$s' =$ _____ <-----	
<u>6. Signatur s aus s' extrahieren:</u> $s \equiv$ _____		

7. Kontrolle mit Signatur s direkt rechnen:

Platz für Nebenrechnungen:

Lösung:

Kunde kennt den Public Key (3, 33)	Meldung	Bank kennt Secret Key (3, 11, 7)
1. Wahl der Nachricht m: $m = 2$		
2. Nachricht m „blinden“: $r = 5$ $m' \equiv 5^3 \cdot 2 \bmod 33 \equiv 19$		
3. geblindete Nachricht m' schicken:	$m' = 19$ ----->	
4. Nachricht m' signieren:		$s' \equiv (19)^7 \bmod 33 \equiv 13$
5. Signatur s' zurückschicken:	$s' = 13$ <-----	
6. Signatur s aus s' extrahieren: $s \equiv \frac{13}{5} \equiv 13 \cdot 5^{-1} \equiv 13 \cdot 20 \equiv 29 \bmod 33$		

7. Kontrolle mit Signatur s direkt rechnen:

$$\underbrace{m^7 \equiv 2^7 \equiv 29 \bmod 33 = s}_{OK}$$

Bemerkung:

Der Kunde kann nun mit $(m, s) = (2, 29)$ im Shop einkaufen. Der Shopbesitzer will natürlich kontrollieren, ob das Geld echt ist und überprüft die Signatur

$$\underbrace{s^3 \equiv 29^3 \equiv 2 \bmod 33 = m}_{OK}$$

Bei der Einlieferung des Geldes bei der Bank, macht die Bank natürlich die gleiche Überprüfung der Signatur wie der Shopbesitzer.

Aufgabe 7**14 Punkte****Aufgabe 7.1****4 Punkte**

Gegeben sind drei Gleichungen:

a) $E: y^2 \equiv x^3 + 3x + 6 \text{ über } \mathbb{Z}_{21}$

b) $E: y^2 \equiv x^3 + 3x + 10 \text{ über } \mathbb{Z}_{13}$

c) $E: y^2 \equiv x^3 + 4x + 2 \text{ über } \mathbb{Z}_{29}$

Begründen Sie welche dieser Gleichungen als Gleichung für eine elliptische Kurve dienen kann und welche nicht.

Lösung:

a) Kann keine Gleichung für eine elliptische Kurve sein, da der Modulus keine Primzahl ist.

[1 P.]

b) Die Nichtsingularitätsbedingung $4 \cdot 3^3 + 27 \cdot 10^2 \equiv 2808 \equiv 0 \text{ mod } 13$ ergibt Null, daher kann es keine Gleichung für eine elliptische Kurve sein.

[1½ P.]

c) Die Nichtsingularitätsbedingung $4 \cdot 4^3 + 27 \cdot 2^2 \equiv 364 \equiv 16 \text{ mod } 29 \not\equiv 0 \text{ mod } 29$, also kann sie Gleichung für eine elliptische Kurve sein.

[1½ P.]

Aufgabe 7.2**10 Punkte**

Gegeben ist die elliptische Kurve $E: y^2 \equiv x^3 + 3x + 9$ über \mathbb{Z}_{19}

- a) [2 P.] Liegt der Punkt $B(15; 8)$ auf der Kurve?
 b) [7 P.] Von einem Punkt P , der auf der Kurve liegt kennt man die Koordinaten von $6 \cdot P = Q(15; 16)$ und $4 \cdot P = R(4; 16)$. Bestimmen Sie die Koordinaten des Punktes $S = 14 \cdot P$.
 c) [1 P.] Sie bestimmen alle Punkte der Kurve und kommen auf die Anzahl 32. Kann diese Zahl stimmen?

Falls es Ihnen hilft, dürfen Sie die Kehrwerttabelle mod 19 im Folgenden verwenden.

x	1	2	3	4	5	6	7	8	9	10
$x^{-1} \bmod 19$	1	10	13	5	4	16	11	12	17	2

x	11	12	13	14	15	16	17	18
$x^{-1} \bmod 19$	7	8	3	15	14	6	9	18

Lösungen

- a) $P(15; 8)$ in die Gleichung einsetzen.

Der Punkt $P(15; 8)$ in $y^2 \equiv x^3 + 3x + 9 \bmod 19$ eingesetzt: $8^2 \equiv 15^3 + 3 \cdot 15 + 9 \bmod 19$

$$8^2 \equiv 64 \equiv 7 \bmod 19 \quad \text{und} \quad 15^3 + 3 \cdot 15 + 9 \equiv 3429 \equiv 9 \bmod 19 \quad 1\frac{1}{2} \text{ P.}$$

Resultat: Der Punkt $P(15; 8)$ liegt demnach nicht auf der Kurve. $\frac{1}{2}$ P.

- b) Koordinaten von Punkt $S = 14 \cdot P = 8 \cdot P + 6 \cdot P = 2R + Q$. 1 P.

Für $2 \cdot R(4; 16)$ gilt:

$$s \equiv \frac{3 \cdot x_1^2 + a}{2 \cdot y_1} \bmod p \equiv \frac{3 \cdot 4^2 + 3}{2 \cdot 16} \bmod 19 \equiv \frac{51}{32} \bmod 19 \equiv \frac{13}{13} \bmod 19 \equiv 1$$

$$x_3 \equiv s^2 - x_1 - x_2 \bmod p \equiv 1^2 - 4 - 4 \bmod 19 \equiv -7 \bmod 19 \equiv 12$$

$$y_3 \equiv s(x_1 - x_3) - y_1 \bmod p \equiv 1(4 - 12) - 16 \bmod 19 \equiv 14$$

Also: $2 \cdot (4; 16) = (12; 14)$ 3 P.

Für $2 \cdot R(4; 16) + Q(15; 16) = (12; 14) + (15; 16)$ gilt

Detailberechnungen:

$$s \equiv \frac{y_2 - y_1}{x_2 - x_1} \bmod p \equiv \frac{16 - 14}{15 - 12} \bmod 19 \equiv \frac{2}{3} \bmod 19 \equiv (2 \cdot 3^{-1}) \bmod 19$$

$$\equiv (2 \bmod 19 \cdot 3^{-1} \bmod 19) \bmod 19 \equiv (2 \cdot 13) \bmod 19 \equiv 7 \quad (\text{siehe Tabelle})$$

$$x_3 \equiv s^2 - x_1 - x_2 \bmod p \equiv 7^2 - 15 - 12 \bmod 19 \equiv 22 \bmod 19 \equiv 3$$

$$y_3 \equiv s(x_1 - x_3) - y_1 \bmod p \equiv 7(12 - 3) - 14 \bmod 19 \equiv 49 \bmod 19 \equiv 11$$

Resultat: $(12; 14) + (15; 16) = 14 * P = (3; 11)$

3 P.

c) Mit dem Theorem von Hasse, kann die Anzahl der Kurvenpunkte abgeschätzt werden:

$$p + 1 - 2\sqrt{p} \leq |E| \leq p + 1 + 2\sqrt{p}$$

Für $p = 23$ ist das somit: $19 + 1 - 2\sqrt{19} \leq |E| \leq 19 + 1 + 2\sqrt{19}$, also: $11 \leq |E| \leq 28$

Resultat:

Eine Kurve mit mod 23 kann zwischen 11 und 28 Punkte haben, somit kann diese Anzahl **nicht** stimmen.

Aufgabe 8**4 Punkte**

Im folgenden Protokoll wird ein symmetrischer Schlüssel mittels einem Kurier auf zwei Rechenzentren verteilt (RZ₁ und RZ₂). Die Operation \oplus bedeutet die XOR-Operation.

RZ ₁	Kurier	RZ ₂
Erzeugt Schlüsselteil T ₁ = 6C	T ₁ in verschlossenem Couvert ----->	Unterschreibt, das Couvert in unbeschädigtem Zustand erhalten zu haben. Erzeugt Schlüsselteil T ₂ = 15
	Bringt die Bestätigung zurück und T ₂ in verschl. Couvert <-----	
Erzeugt Schlüsselteil T ₃ = A9	T ₃ in verschlossenem Couvert ----->	Unterschreibt, das Couvert in unbeschädigtem Zustand erhalten zu haben. Berechnet: Masterkey = T ₁ \oplus T ₂ \oplus T ₃
	Bringt die Bestätigung zurück <-----	
Berechnet: Masterkey = T ₁ \oplus T ₂ \oplus T ₃		

- [2 P.] Berechnen Sie den ausgetauschten Schlüssel = Masterkey = T₁ \oplus T₂ \oplus T₃
- [2 P.] Es gelang dem Kurier ein Couvert zu öffnen, den Teilschlüssel zu betrachten und das Couvert wieder so zu verschliessen, dass es wie unversehrt aussah. Den Teilschlüssel verkaufte er für viel Geld an die Mafia. Kann die Mafia mit diesem Teilschlüssel etwas über den Masterkey erfahren? Wenn ja, was und wie schlimm ist dieser Angriff?

Lösung:

a) Masterkey = T₁ \oplus T₂ \oplus T₃ = 6C \oplus 15 \oplus A9 = D0

- b) Es passiert nichts. Selbst wenn er zwei Couverts hätte öffnen können, hätte er keine Information über den Schlüssel. Dies selbst wenn er unendlich viele Computerressourcen zur Verfügung hätte. Er (oder die Mafia) hätte nicht Informationen, als sie durch raten erhalten würden, resp. nicht mehr als sie zum vorneherein schon wissen, nämlich der Masterkey **00** bis **FF** lauten muss.

Aufgabe 9**5 Punkte****Aufgabe 9.1:****2 Punkte**

Asymmetrische Schlüsselpaare können auf Smart-Cards berechnet werden. Technisch ist es dabei möglich sicherzustellen, dass der private Schlüssel nicht aus der Karte ausgelesen werden kann. Der private Schlüssel existiert in diesem Fall also ausschliesslich auf der Karte selber. Beantworten Sie zu diesem Szenario die folgende Frage:

Für welche kryptografische Operation (Verschlüsseln oder Signieren) sollten Sie das dem beschriebenen Szenario zugrundeliegende Schlüsselpaar *nicht* verwenden? (1 Punkt) Begründen Sie Ihre Antwort. (1 Punkt)

Lösung:

Das Schlüsselpaar sollte *nicht* für Verschlüsselungsoperationen verwendet werden. (1 Punkt)
Begründung: Wenn das Schlüsselpaar nur auf der Smart-Card existiert, dann gibt es kein Backup des privaten Schlüssels. Bei einem Schlüsselverlust sind deshalb die verschlüsselten Daten verloren. (1 Punkt)

Aufgabe 9.2:**3 Punkte**

Welches grundsätzliche Problem ergibt sich bei der Anwendung von PKI-Verfahren, wenn aufgrund eines Software-Problems keine CRLs mehr ausgestellt werden können? (1 Punkt) Welche Sicherheitsprobleme entstehen dabei bei der Verschlüsselung von Daten? (1 Punkt) Und welche bei der Verifikation von Signaturen? (1 Punkt)

Lösung:

Das grundsätzliche Problem besteht darin, dass Zertifikate vor deren Verwendung von den Applikationen nicht mehr auf Gültigkeit hin geprüft werden können. (1 Punkt)

Wenn Zertifikate vor der Verschlüsselung von Daten nicht mehr auf Gültigkeit hin geprüft werden können, werden Daten möglicherweise für einen Angreifer verschlüsselt. (1 Punkt)

Wenn Zertifikate vor der Verifikation von Signaturen nicht mehr auf Gültigkeit hin geprüft werden können, können Signaturen, die von einem Angreifer erstellt worden sind, nicht mehr als gefälscht erkannt werden. (1 Punkt)

Aufgabe 1

9 Punkte

Das linke Klartext-Bild ist mit einer Blockchiffre verschlüsselt worden. Das rechte Bild ist die verschlüsselte Version.



a) [1 P.] Geben Sie einen geeigneten, aktuellen & standardisierten Algorithmus an. AES

b) [1 P.] Welche minimale Schlüsselgrösse muss verwendet werden? 128 Bit

c) [1 P.] In welchem Modus wurde das obige Bild verschlüsselt? ECB-Modus

d) [1 P.] Begründen Sie Ihre Angabe des Modus in Aufgabe c).

Gleiche Klartextblöcke werden in gleiche Chiffratblöcke verschlüsselt, daher ist in diesem Bild die Grundstruktur nach wie vor gut sichtbar.

e) [2 P.] Geben Sie je einen (weiteren) Vor- und Nachteil des obigen Modus an.

Vorteil: Parallelisierung resp. Teilverschlüsselung

Nachteil: Vertauschen von Blöcken wird ev. n. entdeckt.

f) [1 P.] Geben Sie einen Modus an, der das Bild besser verschlüsselt.

CBC- oder CTR-Modus (Angaben von weiteren Modi wie OFB, GCM o.a. werden auch akzeptiert.)

g) [1 P.] Das Klartext-Bild soll auf dem Übertragungsweg gegen Verändern geschützt werden. Wie machen Sie das, wenn der oben erwähnte Blockchiffrieralgorithmus benutzt werden soll?

CBC-MAC-Berechnung (oder kombinierte Modi wie GCM)

h) [1 P.] Gibt es für f) eine (symmetrische) Alternative ohne Verwendung eines Blockchiffrierers? Was braucht es dazu? Wie heisst die Konstruktion?

Ja, mit einer Stromchiffre, dazu braucht es einen Pseudorandom-generator.

Aufgabe 2

6 Punkte

Voraussetzungen:

- Alice besitzt den Schlüssel K_1
- Bob besitzt den Schlüssel K_2

Alice möchte Bob die Meldung M verschlüsselt zuschicken. Dabei verwenden Sie das untenstehende Protokoll, das ohne vorgängigen Schlüsselaustausch auskommt. Als Verschlüsselungsoperation wird die Stromchiffre benutzt.

Alice mit K_1	unsichere Leitung	Bob K_2
Alice verschlüsselt die Nachricht M mit ihrem geheimen Schlüssel K_1 .		
	$C_1 = M \oplus K_1$ ----->	
		Bob verschlüsselt die Nachricht C_1 mit seinem geheimen Schlüssel K_2 .
	$C_2 = C_1 \oplus K_2$ <-----	
Alice entschlüsselt die Nachricht C_2 mit ihrem geheimen Schlüssel K_1 .		
	$C_3 = C_2 \oplus K_1$ ----->	
		Bob entschlüsselt die Nachricht C_3 mit seinem geheimen Schlüssel K_2 . $Y = C_3 \oplus K_2 \stackrel{?}{=} M$

- a) [3 P.] Beweisen Sie nun, dass der von Bob zu Letzt berechnete Wert $Y = C_3 \oplus K_2$ tatsächlich gleich der Nachricht M ist.
- b) [3 P.] Es ist nun offensichtlich, dass man nun drei Meldungen statt nur eine Meldung über die Leitung schicken muss. Aufgrund dessen, dass dieses Protokoll nicht implementiert wurde, muss ja noch irgendwo ein anderer Hacken sein. Finden Sie diesen Hacken. **Tipp:** Versetzen Sie sich in Eve, die alle Meldungen abhören kann. Eve hat die Idee, dass sie einfach einmal alle drei über die Leitung geschickten Chiffre miteinander XOR'ed. Berechnen Sie nun, was dann rauskommt.

Lösung:

a)

- $C_3 = (C_1 \oplus K_2) \oplus K_1 = ((M \oplus K_1) \oplus K_2) \oplus K_1 = ((M \oplus K_1) \oplus K_1) \oplus K_2 = M \oplus K_2$
- Und daher ist $Y = (M \oplus K_2) \oplus K_2 = M$

- b) Eve sieht ja alle 3 chiffrierten Meldungen; sie muss nur alle 3 Meldungen miteinander XOR'en.

$$C_1 \oplus C_2 \oplus C_3 = C_1 \oplus C_2 \oplus (C_2 \oplus K_1) = C_1 \oplus K_1 = (M \oplus K_1) \oplus K_1 = M$$

Aufgabe 3**7 Punkte**

Sie setzen für eine PIN-Block-Verschlüsselung zwei Typen von Blockchiffren ein:

- Typ A = 256 Bit AES.
- Typ B = Doppel-AES mit je 128 Bit Schlüsselgrösse (Analog einem Doppel-DES).

Die zwei Typen unterziehen Sie nun einer kryptoanalytischen Betrachtung und kommen zu den folgenden Ergebnissen, die Sie nun entsprechend in der Tabelle auswählen und ausfüllen.

Falsches Ankreuzen gibt Punktabzug, die Summe kann aber nicht negativ werden!

Angriff	Typ von Attacke
Table look up	<input checked="" type="checkbox"/> Für Typ A und B muss ungefähr der gleiche Aufwand betrieben werden. <input type="checkbox"/> Für Typ A muss der kleinere Aufwand betrieben werden nämlich Faktor _____ anstatt _____. <input type="checkbox"/> Für Typ B muss der kleinere Aufwand betrieben werden nämlich Faktor _____ anstatt _____. <input type="checkbox"/> Man kann die Typen bei diesem Angriff gar nicht vergleichen.
Exhaustive Key Search	<input checked="" type="checkbox"/> Für Typ A und B muss ungefähr der gleiche Aufwand betrieben werden. <input type="checkbox"/> Für Typ A muss der kleinere Aufwand betrieben werden nämlich Faktor _____ anstatt _____. <input type="checkbox"/> Für Typ B muss der kleinere Aufwand betrieben werden nämlich Faktor _____ anstatt _____. <input type="checkbox"/> Man kann die Typen bei diesem Angriff gar nicht vergleichen.
Time-memory-Trade off "meet-in-the-middle"	<input type="checkbox"/> Für Typ A und B muss ungefähr der gleiche Aufwand betrieben werden. <input type="checkbox"/> Für Typ A muss der kleinere Aufwand betrieben werden nämlich Faktor _____ anstatt _____. <input checked="" type="checkbox"/> Für Typ B muss der kleinere Aufwand betrieben werden nämlich Faktor $2 \cdot 2^{128} = 2^{129}$ anstatt 2^{256} . <input type="checkbox"/> Man kann die Typen bei diesem Angriff gar nicht vergleichen.


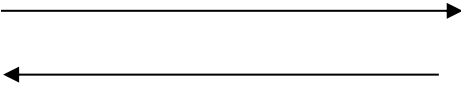

Aufgabe 4


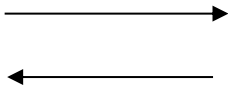

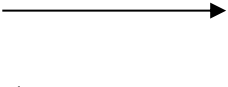

2 + 4 = 6 Punkte

Aufgabe 4.1

2 Punkte

Im Folgenden sind die Abläufe eines nicht angegriffenen und eines angegriffenen Schlüsselaustausch Protokolls gegeben.

Alice		Bob
		
Super, Bob und ich haben nun den gleichen Schlüssel K ausgetauscht!		Super, Alice und ich haben nun den gleichen Schlüssel K ausgetauscht!

Alice		Eve		Bob
				
Super, Bob und ich haben nun den gleichen Schlüssel K ausgetauscht!??				Super, Alice und ich haben nun den gleichen Schlüssel K ausgetauscht!??

Beantworten Sie die folgenden Fragen (je ½ P.)

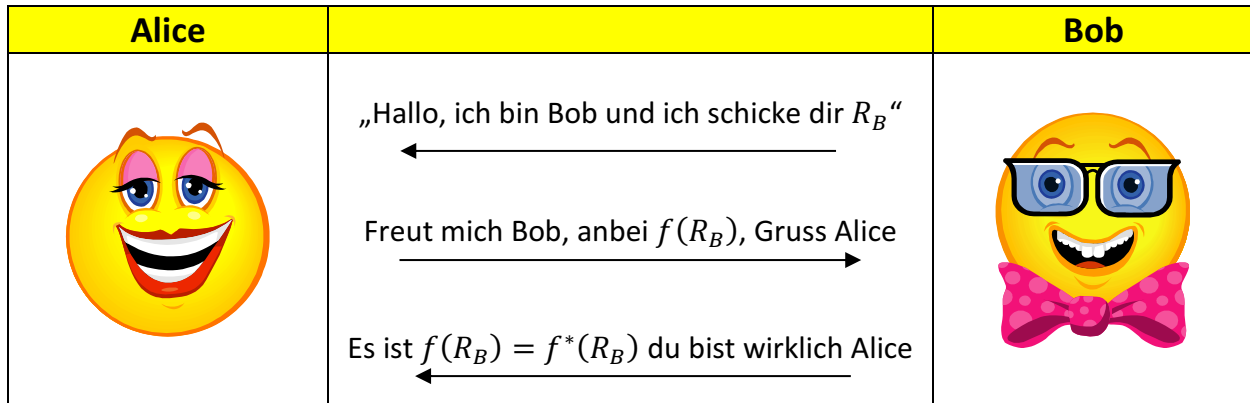
- Wie heisst der dargestellte Angriff? Man-in-the middle
- Bei welchem Schlüsselaustausch Protokoll ist der Angriff erfolgreich? Diffie-Hellman
- Warum ist der Angriff erfolgreich? Protokoll hat keine Benutzerauthentizität
- Korrigieren Sie im **angegriffenen Protokoll** die Aussage „...**den gleichen Schlüssel K ausgetauscht!??**“, d.h. ersetzen Sie ..., so dass die beiden Aussagen stimmen.

Alice hat mit Eve einen K1 und Bob hat mit Eve einen K2 ausgetauscht.

Aufgabe 4.2**4 Punkte**

Im Folgenden ist der Ablauf bei einer Authentisierung gegeben. Im Protokoll werden die folgenden Abkürzungen verwendet:

- R_B = Zufallswert von Bob gewählt; «R» steht für Random.
- $f(R_B)$ der Hashwert von R_B , wobei f eine kryptographisch sichere Hashfunktion (z.B. SHA-2 oder SHA-3) ist. Das ist der Wert den Alice berechnet.
- $f^*(R_B)$ der Hashwert von R_B , wobei f^* die gleiche kryptographisch sichere Hashfunktion wie f ist, aber der Hashwert wird von Bob gerechnet.



Beantworten Sie die folgenden Fragen (je 1 P.)

- Ist das Protokoll „mutual“, wenn JA, dann begründen Sie warum, wenn NEIN, wer authentifiziert sich gegenüber wem?

Nein, nur Alice authentifiziert sich gegenüber Bob.

- Ist die gewählte Funktion (kryptographisch sichere Hashfunktion, z.B. SHA-2 oder SHA-3) ein geeigneter Mechanismus? Wenn JA, warum, wenn NEIN warum nicht, und geben Sie einen geeigneteren an.

Nein, jedermann könnte diesen Hash rechnen; CBC-MAC oder HMAC.

- Für eine erfolgreiche einseitige oder gegenseitige Authentisierung fehlt ein wichtiges Element, was für eines?

**Es fehlt ein gemeinsamer symmetrischer Schlüssel K
(Resp. zwei Public Key Schlüsselpaare)**

- Ein wesentliches Element für eine erfolgreiche einseitige oder gegenseitige Authentisierung ist vorhanden, was für eines?

Die Anfrage mittels einem Randomwert.

Aufgabe 5**2 + 1 + 3 = 6 Punkte**

Gegeben ist ein RSA System mit den Primzahlen $p = 23$ und $q = 41$ sowie dem öffentlichen Exponenten $e = 9$ und dem geheimen Exponenten $d = 489$.

- a) [2 P.] Zeigen Sie, dass es der öffentliche Exponent $e = 9$ legitim ist.
- b) [1 P.] Zeigen Sie, dass der Exponent d zum Exponent e passt.
- c) [3 P.] Nun verschlüsselt Alice die Meldung $m = ?$ mit dem gegebenen RSA-System und schickt die verschlüsselte Meldung $c = 492$ über die Leitung. Die Angreiferin Eve will die verschlüsselte Meldung c so verändern, dass bei Bob nach dem Entschlüsseln die Meldung $5 \cdot m$ erscheint. Berechnen Sie das neue Chiffre c' , welches Eve erzeugen und an Bob weiterschicken muss.

Lösung:

a) $N = p \cdot q \Rightarrow \varphi(N) = \varphi(p) \cdot \varphi(q) = (p - 1) \cdot (q - 1) = 22 \cdot 40 = 880$.

Es muss gelten: $\text{ggT}(e; \varphi(N)) = \text{ggT}(9; 880) = 1$

1 P.

Das kann man auf ohne TR z.B. auf die folgenden zwei Arten machen:

- i) Primfaktorzerlegung von $880 = 2^4 \cdot 5 \cdot 11$: Sie enthält den Teiler 3 nicht, und damit ist der geforderte $\text{ggT}(9; 880) = \text{ggT}(3^2; 2^4 \cdot 5 \cdot 11) = 1$
- ii) $e = 9 = 3^2$. Nun muss ich nur noch zeigen, dass 880 den Teiler 3 nicht enthält. Das kann einfach gezeigt werden mit $\frac{880}{3} = 293,33 \dots$

1 P.

b) Es ist zu zeigen, dass $e \cdot d \bmod \varphi(N) \equiv 1$ ist.

$$9 \cdot 489 \bmod 880 \equiv 4401 \bmod 880 \equiv 5 \cdot 880 + 1 \bmod 880 \equiv 1.$$

1 P.

c) Eve muss $c' \equiv 5^9 \cdot c \bmod N \equiv 5^9 \cdot 492 \bmod 943 \equiv 1'953'125 \cdot 492 \bmod 943$
 $\equiv 172 \cdot 492 \bmod 943 \equiv 84'624 \bmod 943 \equiv 697$

Resultat: Eve muss den Wert 697 an Bob zuschicken.

3 P.

Kontrolle (wird in der Prüfung nicht verlangt):

$$697^{489} \bmod 943 \equiv 615 = 5 \cdot 123$$

Aufgabe 6**5 Punkte**

Die 3-stellige Zahl x wird mit der Formel $y \equiv (a \cdot x + b) \bmod N$ verschlüsselt. Die Entschlüsselungsfunktion lautet: $x \equiv a^{-1} \cdot (y - b) \bmod N$. Dabei ist $N = 11 \cdot 23 \cdot 41$ ein Produkt von drei Primzahlen; der Wert N ist öffentlich bekannt. Die Werte a und b bilden in der Form $(a; b)$ den geheimen Schlüssel. Die Werte a und b sind für die Verschlüsselung und Entschlüsselung geeignete Werte aus der Menge $\{2; 3; \dots; N - 1\}$

- a) [1 P.] Begründen Sie, ob es sich hier um eine symmetrische oder asymmetrische Verschlüsselung handelt. **Achtung:** Die Angabe „symmetrisch“ oder „asymmetrisch“ ohne stichhaltige Begründung gibt keine Punkte!

Lösung:

Es handelt sich um eine symmetrische Verschlüsselung, da Sender und Empfänger den gleichen Schlüssel $(a; b)$ haben müssen. (Dies ungeachtet dessen, dass der Sender noch auf $a^{-1} \bmod N$ umrechnen muss).

- b) [4 P.] Aus wie vielen möglichen Schlüsseln der Form $(a; b)$ können bei dieser Verschlüsselung ausgewählt werden? Es ist die exakte Zahl anzugeben.

Lösung allgemein:

- Für b sind alle möglichen Werte aus der Menge $\{2; 3; \dots; N - 1\}$ möglich, daher gibt es für b total $N - 2$ mögliche Werte.
- Für a sind aus der Menge $\{2; 3; \dots; N - 1\}$ nur diejenigen möglich, die teilerfremd zu N sind. Da die Zahl 1 aber nicht drin sein darf lautet die Anzahl der möglichen Werte für a somit:
$$\varphi(N) - 1 = \varphi(r \cdot s \cdot t) - 1 = \varphi(r) \cdot \varphi(s) \cdot \varphi(t) - 1 = (r - 1) \cdot (s - 1) \cdot (t - 1) - 1.$$
- Somit gibt es total $(N - 2) \cdot [(r - 1) \cdot (s - 1) \cdot (t - 1) - 1]$ mögliche Schlüssel.

Lösung mit den konkreten Zahlen:

$$r = 11; s = 23; t = 41 \Rightarrow N = r \cdot s \cdot t = 11 \cdot 23 \cdot 41 = 10373$$

$$\Rightarrow N - 2 = 10371$$

$$\Rightarrow \varphi(N) - 1 = \varphi(r \cdot s \cdot t) - 1 = (11 - 1) \cdot (23 - 1) \cdot (41 - 1) - 1 = 10 \cdot 22 \cdot 40 - 1 = 8799$$

Somit gibt es total $10371 \cdot 8799 = 91'254'429$ mögliche Schlüssel.

Aufgabe 7**8 Punkte**

Gegeben ist die elliptische Kurve $E: y^2 \equiv x^3 + x + 1$ über \mathbb{Z}_{23} .

- a) [2 P.] Überprüfen Sie, ob der Punkt (11; 19) auf der Kurve liegt.
 b) [3 P.] Der Punkt $P(3; 10)$ liegt auf der Kurve. Berechnen Sie die Koordinaten des Punktes $R = 2P$.
 c) [3 P.] Die Punkte $P(3; 10)$ und $Q(9; 7)$ liegen auf der Kurve. Berechnen Sie die Koordinaten des Punktes $T = P + Q$.

Falls es Ihnen hilft, dürfen Sie zudem die Kehrwerttabelle mod 23 verwenden.

x	1	2	3	4	5	6	7	8	9	10	11
$x^{-1} \bmod 23$	1	12	8	6	14	4	10	3	18	7	21

x	12	13	14	15	16	17	18	19	20	21	22
$x^{-1} \bmod 23$	2	16	5	20	13	19	9	17	15	11	22

Lösung:

a) $y^2 \equiv 19^2 \equiv 361 \equiv 16 \bmod 23$

$$x^3 + x + 1 \equiv 11^3 + 11 + 1 \equiv 1331 + 11 + 1 \equiv 1343 \equiv 9 \bmod 23$$

Resultat: Da die Werte nicht gleich sind, folgt dass der Punkt $P(11; 19)$ nicht auf der Kurve liegt.

b) Berechnung von $2 \cdot P = 2 \cdot (3; 10)$

$$s \equiv \frac{3 \cdot x_1^2 + a}{2 \cdot y_1} \bmod p \equiv \frac{3 \cdot 3^2 + 1}{2 \cdot 10} \equiv \frac{28}{20} \equiv 28 \cdot 20^{-1} \equiv 5 \cdot 20^{-1} \equiv 5 \cdot 15 \equiv 75 \equiv 6 \bmod 23$$

Bemerkung: In diesem Fall dürfte man kürzen, Grund: Die Kürzungsregel darf angewandt werden.

$$\frac{xa}{xb} \equiv xa \cdot (xb)^{-1} \equiv a \cdot (b)^{-1} \bmod m \Leftrightarrow \text{ggT}(x, m) = 1$$

$$\text{ggT}(4, 23) = 1 \Rightarrow \frac{28}{20} \equiv \frac{4 \cdot 7}{4 \cdot 5} \equiv 7 \cdot 5^{-1} \equiv 7 \cdot 14 \equiv 98 \equiv 6 \bmod 23$$

Oder einen Schritt weiter:

$$\text{ggT}(5, 23) = 1 \Rightarrow \frac{5}{20} \equiv \frac{5 \cdot 1}{5 \cdot 4} \equiv 1 \cdot 4^{-1} \equiv 1 \cdot 6 \equiv 6 \bmod 23$$

$$x_3 \equiv s^2 - x_1 - x_2 \bmod p \equiv 6^2 - 3 - 3 \bmod 23 \equiv 30 \bmod 23 \equiv 7$$

$$y_3 \equiv s(x_1 - x_3) - y_1 \bmod p \equiv 6(3 - 7) - 10 \bmod 23 \equiv -34 \equiv 12 \bmod 23$$

Also: $2 \cdot (3; 10) = (7; 12)$

3 P.

c) Berechnung von $P + Q = (3; 10) + (9; 7)$

Detailberechnungen:

$$s \equiv \frac{y_2 - y_1}{x_2 - x_1} \bmod p \equiv \frac{7 - 10}{9 - 3} \bmod 23 \equiv \frac{-3}{6} \equiv (-3) \cdot 6^{-1} \equiv 20 \cdot 4 \equiv 80 \equiv 11 \bmod 23$$

$$x_3 \equiv s^2 - x_1 - x_2 \bmod p \equiv 11^2 - 3 - 9 \equiv 209 \equiv 17 \bmod 23$$

$$y_3 \equiv s(x_1 - x_3) - y_1 \bmod p \equiv 11(3 - 17) - 10 \equiv -164 \equiv 20 \bmod 23$$

Resultat: $(3; 10) + (9; 7) = (17; 20)$

3 P.

Aufgabe 8**8 Punkte**

Alice schickt eine Meldung $M \in \mathbb{Z}_{19}$ verschlüsselt an Bob. Es wird die Verschlüsselungsmethode von Volker-Müller mit Elliptischen Kurven eingesetzt. Aus gewissen Gründen wird die eigentliche Verschlüsselungsoperation ersetzt. Anstatt der XOR-Operation \oplus wird die Addition mod 19 verwendet. Bob hat die folgenden Elemente gewählt...

- ... die elliptische Kurve $E: y^2 \equiv x^3 + 3x + 9$ über \mathbb{Z}_{19} .
- ... den Basispunkt $P(2; 17)$.
- ... den Secret Key $d = 21$.

Alice verschlüsselt die Nachricht $M = 12$. Sollte Alice einen zufälligen Wert generieren müssen, dann können Sie annehmen, dass der Wert 15 gewählt wird.

Alice beginnt den Datenaustausch mit der Meldung (*) = „Hallo Bob, ich möchte dir eine verschlüsselte Meldung schicken.“

Füllen Sie nun die folgende Tabelle aus. Sollten es in der Tabelle zu wenig Platz für allfällige Berechnungen haben, so führen Sie diese bitte auf der nächsten Seite durch.

Sämtliche Operationen mit den Punkten können in der separat ausgeteilten Tabelle nachgeschaut werden.

Alice	unsichere Leitung	Bob
	(*) ----->	
	$K_{\text{pub}} = (p, a, b, q, P, Q)$ <-----	Bob schickt $K_{\text{pub}} = (p, a, b, q, P, Q)$
K_{pub} in der vorgegebenen Reihenfolge angeben. $= (p, a, b, q, P, Q)$		
Verschlüsselt Meldung $M = 12$		
	Meldung notieren ----->	
		Entschlüsselung

Lösung:

Alice	unsichere Leitung	Bob
	(*) ----->	
	$K_{\text{pub}} = (p, a, b, q, P, Q)$ <-----	Bob schickt $K_{\text{pub}} = (p, a, b, q, P, Q)$
K_{pub} in der vorgegebenen Reihenfolge angeben. $= (p, a, b, q, P, Q)$ $(19, 3, 9, 23, (2; 17), (16; 12))$		
Verschlüsselt Meldung $M = 12$ $i = 15$ $K_E = i \cdot P = 15 \cdot (2; 17) = (12; 5)$ $K_M = i \cdot Q = 15 \cdot (16; 12) = (11; 9)$ $Y = M + x\text{Koord von } K_M \text{ mod } 19$ $= 12 + 11 \equiv 23 \equiv 4 \text{ mod } 19$		
	Alle nötigen Werte $(Y, K_E) = (4, (12; 5))$ ----->	
		$K_M = d \cdot K_E$ $= 21 \cdot (12; 5) = (11; 9)$ $M = Y - x\text{Koord } K_M \text{ mod } 19$ $= 4 - 11 \equiv -7 \equiv 12 \text{ mod } 19$

Bewertungshinweise:

- Q berechnen, 1 P.
- Angabe des korr. K_{pub} 1 P.
- Verschlüsselung $3 \cdot 1 = 3$ P.
- Korr. Meldung an Bob, 1 P.
- Entschlüsselung $2 \cdot 1 = 2$ P.

Aufgabe 9**5 Punkte**

Sie nutzen eine Sicherheitsapplikation, die Zertifikate für die Authentifizierung des Kommunikationspartners verwendet. Beim Versuch, sich mit einem neuen Partner zu verbinden, erhalten Sie eine Fehlermeldung, wonach das Root-CA-Zertifikat, das dem Zertifikat des Verbindungspartners zugrunde liegt, nicht vertrauenswürdig sei. Sie haben die Möglichkeit, den Verbindungsaufbau abubrechen oder das Root-CA-Zertifikat zu installieren. Die Sicherheitsapplikation zeigt Ihnen den Inhalt und weitere Eigenschaften des Root-CA-Zertifikats an. Sie entscheiden sich, das Root-CA-Zertifikat installieren.

- a) Was müssen Sie tun, bevor Sie das Zertifikat installieren. Beschreiben Sie den Vorgang stichwortartig (3 Punkte).
- b) Die oben beschriebene Fehlermeldung tritt glücklicherweise nur selten auf, weil die Prüfung des Zertifikats des Verbindungspartners in der Regel ohne Fehler durchgeführt werden kann. Beschreiben Sie in wenigen Worten, wie die Sicherheitsapplikation (oder das Betriebssystem) die Echtheit des Zertifikats des Verbindungspartners überprüft (1 Punkt). Welche Rolle spielt dabei der so genannte Trust Anchor (1 Punkt).

Lösung:

- a) Das Root-CA-Zertifikat muss auf Echtheit geprüft werden (1 Punkt). Die Echtheitsprüfung wird mithilfe des Fingerprints durchgeführt (1 Punkt). Bei der Überprüfung des Fingerprints wird der lokal angezeigte Fingerprint mit dem Fingerprint aus einer vertrauenswürdigen Referenzquelle verglichen (1 Punkt).
- b) Die Sicherheitsapplikation (oder das Betriebssystem) überprüft die Signatur auf dem Zertifikat des Verbindungspartners mithilfe des Zertifikats der herausgebenden Zertifizierungsstelle (Root-CA-Zertifikat) (1 Punkt). Die herausgebende Zertifizierungsstelle muss im System als vertrauenswürdige Root-CA hinterlegt sein; sie wird so zum Trust Anchor (1 Punkt).