1321 EINFÜHRUNG IN DIE KRYPTOGRAFIE – FRAGENKATALOG

KE1

(1) Worum geht es bei der Kryptografie?

- **Vertraulichkeit**: gewährleisten, dass vertrauliche Nachrichten nur von Befugten gelesen werden können (sichere Kommunikation über einen unsicheren Kanal).
- aber auch:
  - **Authentikation** (stelle sicher, dass eine Nachricht von X wirklich von X ist)
  - **Integrität** (stelle sicher, dass eine Nachricht im Zuge der Übermittlung nicht verändert wurde)

(2) Def. Kryptosystem und mathematische Beschreibung als 5-tupel (*P, C, K, E, D*)

- $\boldsymbol{P}$:= endliche Menge von Klartexten (plain texts)
- $\boldsymbol{C}$:= endliche Menge von Geheimtexten (cipher texts)
- $\boldsymbol{K}$:= endliche Menge von Schlüsseln (plain texts)
  Für jedes $K \in \boldsymbol{K}$ gibt es ein $e_K : \boldsymbol{P} \rightarrow \boldsymbol{C} \in \boldsymbol{E}$ und $d_K : \boldsymbol{C} \rightarrow \boldsymbol{P} \in \boldsymbol{D}$, sodass $d_K(e_K(x))=x$
- also muss e_K injektiv sein, d.h. jedes Bild hat genau 1 Urbild

(3) symmetrische vs. asymmetrische Verschlüsselung erklären und Beispiele

- **symmetrisch**:
  - aus e_k kann schnell und einfach d_k berechnet werden, und umgekehrt, d.h. jeder der chiffrieren kann, kann auch entschlüsseln
  - Vorteil: geht schnell
  - Nachteil: Schlüssel sind oft zu tauschen, damit Oscar sie nicht herausfinden kann
- **asymmetrisch**: ist ein KS, wenn es folgende Eigenschaften hat:
  - $e_K(x)$ lässt sich für alle $K \in \boldsymbol{K}$ (die **öffentlichen Schlüssel**) und alle Klartexte $x \in \boldsymbol{P}$ sehr schnell berechnen
  - Ohne eine geheime Zusatzinformation (**privater Schlüssel**) lässt sich das Urbild eines Geheimtextes $y \in \boldsymbol{C}$ nicht in angemessener Zeit berechnen, selbst wenn K und e_K bekannt sind
  - Mit der geheimen Zusatzinformation lässt sich $d_K(y)$ für alle $y \in \boldsymbol{C}$ sehr schnell berechnen.
  - Dazu Befugte können aus dem öff. Schlüssel K den privaten Schlüssel schnell herleiten
  - → e_k und K müssen nicht geheimgehalten werden (K = öff. Schlüssel), sondern nur die geheime Zusatzinformation (geheimer Schlüssel)
  - → Vorteil: Viele Leute können sicher miteinander kommunizieren ohne komplizierte Schlüsselabsprachen – Veröffentlichung der öff. Schlüssel mit zugehörigen Chiffrierungsregeln wie in einem Telefonbuch

(4) mono- vs. polyalphabetische Verschlüsselung erklären und Beispiele

- **Monoalphabetisches KS:** jeder Klartextbuchstabe wird auf genau einen Geheimtextbuchtaben abgebildet
  - wenn $P=C=\{a,\ldots,z\}$ weil e_k injektiv und P endlich $\Rightarrow$ e_k ist surjektiv, also eine bijektive Permutation $\Rightarrow$ man erhält das **Permutations-KS** $|K| = 26!$
    - **affines KS $e_k(x) = (ax+b) \bmod 26$** $|K|=|(\mathbf{Z}/26\mathbf{Z})^x \times \mathbf{Z}/26\mathbf{Z}| = 12*26=312$
      - **Verschiebe-KS: a=1** $|K| = 26$
        - **Cäsar-KS: b=23; ROT13: b=13**
- **Polyalphabetische KS**
  - **Blockchiffren:** Klartext in gleich lange Blöcke zerlegen und diese unabhängig voneinander mit dem gleichen Schlüssel chiffrieren.
    - Vigenère-KS, Hill-KS
  - **Stromchiffren**.
    - **periodische** SC: mit Periode d, fall $z_{i+d} = z_i$ für alle i>=1.
    - **synchrone** Stromchiffre: Schlüsselstromist unabh. vom Klartext
      - **Vigenère**-KS: $|K| = 26^m$ (m = Länge Schlüsselwort) (kann auch als synchrone, periodische Stromchiffre der Periode m aufgefasst werden)
      - **One-Time-Pad**
    - **asynchrone** SC, z.B. Selbstschlüssel-KS $|K| = 26$

(5) Permutations-KS

- $\boldsymbol{P} = \boldsymbol{C} = \{a,\ldots,z\}$, $\boldsymbol{K}$ = alle möglichen Permuatationen der Buchstaben a, …, z. Für jede Permutation $\pi \in \boldsymbol{K}$ sei $e_\pi = \pi$ und $d_\pi = \pi^{-1}$.
  - $\Rightarrow |K| = 26!$
  - Vorgehensweise zum Finden einer Schlüsselpermutation:
  - Vereinbarung Schlüsselwort z.B. *sonne* und Schlüsselbuchst. z.B. *e*
  - Jeden Buchstaben ab dem 2. Vorkommen löschen: *sone*
  - Dann alphabetisch mit den noch fehlenden Buchst. auffüllen, das Schlüsselwort beginnt bei dem Schlüsselbuchst.: - fertig ist $e_k = \pi$:

    a b c d e f g h i j k l m n o p q r s t u v w x y z
    w x y z **s o n e** a b c d f g h I j k l m p q r t u v

(6) Berechnen des Inversen einer Zahl a **in Z/mZ:**

- Voraussetzung: a invertierbar $\Leftrightarrow aa^{-1} \bmod m = 1 \Leftrightarrow ggT(a,m) = 1$.
- Berechnung: Bestimme s und t, so dass sa+tm=1 (erweiterter Eukl. Alg.). Dann ist $a^{-1} = s \bmod m$.

(7) Affines KS

- f: $\mathbf{Z}/m\mathbf{Z} \rightarrow \mathbf{Z}/m\mathbf{Z}$ mit $f(x) = (ax+b) \bmod m$ ist injektiv $\Leftrightarrow ggT(a,m)=1$
  - $\Rightarrow$ so ein f ist bijektiv
- $\boldsymbol{P} = \boldsymbol{C} = \mathbf{Z}/26\mathbf{Z}$, $\boldsymbol{K} = (\mathbf{Z}/26\mathbf{Z})^\times \times (\mathbf{Z}/26\mathbf{Z})$. Für ein $K=(a,b) \in \boldsymbol{K}$ sei
  - $e_K : \mathbf{Z}/26\mathbf{Z} \rightarrow \mathbf{Z}/26\mathbf{Z}$ $\quad e_K(x) = (ax + b) \bmod 26$
  - $d_K : \mathbf{Z}/26\mathbf{Z} \rightarrow \mathbf{Z}/26\mathbf{Z}$ $\quad d_K(x) = a^{-1}(y-b) \bmod 26$
- ist nicht mit einem known-Ciphertext, aber mit known-Plaintext-Angriff zu brechen.

(8) Verschiebe-KS:

- $\boldsymbol{P} = \boldsymbol{C} = \boldsymbol{K} = \mathbf{Z}/26\mathbf{Z}$.
- Für alle $K \in \boldsymbol{K}$ sind $e_K : \boldsymbol{P} \rightarrow \boldsymbol{C}$ $\quad e_K(x) = x + K$ und $d_K : \boldsymbol{C} \rightarrow \boldsymbol{P}$ $\quad d_K(x) = y - K$.

(9) Euklidischer Algorithmus:

- $a,b \in \mathbf{Z}$, nicht beide 0. Ein $d \in \mathbf{N}$ heißt ggT(a,b), falls folgende Bedingungen beide erfüllt sind:
  (i) $d|a$ und $d|b$ (ii) Jede ganze Zahl, die a und b teilt, teilt auch d.
- o.B.d.A. sei $a \neq 0$.
  - $b = q_1 a + r_1$
  - $a = q_2 r_1 + r_2$
  - $r_1 = q_3 r_2 + r_3$
  - …
  - $r_{n-2} = q_n r_{n-1} + \boxed{r_n} \rightarrow d$
  - $r_{n-1} = q_{n+1} r_n$

(10)Erweiterter Euklidischer Algorithmus: Gleichungen umstellen nach den Resten:

  - $b - q_1 a = r_1$
  - $a - q_2 r_1 = r_2$
  - $r_1 - q_3 r_2 = r_3$
  - …
  - $r_{n-2} - q_n r_{n-1} = \boxed{r_n} \rightarrow d$
- in die umgestellte Gleichung r_{i+i} setzt man r_i und r_{i-1} ein. Schließlich erhält man s und t für $r_n = d = sa+tb$.

(11) Kerckhoff-Prinzip

- Die Sicherheit eines Kryptosystems darf nicht von der Geheimhaltung der Chiffrierungsregel abhängen, sondern darf nur auf der Geheimhaltung des Schlüssels beruhen. Begründung:
  - Es ist viel schwieriger, einen Algorithmus geheim zu halten als einen Schlüssel.
  - Es ist schwieriger, einen kompromittierten Algorithmus durch einen anderen zu ersetzen als einen kompromittierten Schlüssel.
  - Geheime Algorithmen können durch Reverse-Engineering aus Software- oder Hardware-Implementierungen rekonstruiert werden.
  - Fehler in öffentlichen Algorithmen werden i.A. leichter entdeckt (Prüfung durch möglichst viele Fachleute).
  - Bei der Nutzung von „geheimen“ Verschlüsselungsverfahren muss mit dem Vorhandensein einer Hintertür gerechnet werden.

(12) mögliche Attacken auf KS:

- **1. Known-ciphertext-attack**: Oscar kennt ein relativ großes Stück des Geheimtextes.
- **2. Known-plaintext-attack:** Oscar kennt ein vergleichsweise kleines Stück von zusammengehörigem Klar- und Geheimtext., z.B. Standardgrußformeln zu Beginn/Ende der abgefangenen Nachricht.
- **3. Chosen-plaintext-attack**: Oscar hat Zugang zu einem Chiffrieralgorithmus und möchte die sich dahinter verbergende Chiffrierungsregel und den Schlüssel erkennen.

(13) Kryptoanalyse monoalphabetischer KS

- Häufigkeiten der Einzelbuchstaben ermitteln → kann e und n lokalisieren und die Geheimtextbuchstaben, die der Menge {i,s,r,a,t} entsprechen
- Häufigkeiten der Bigramme ermitteln → kann {i,s,r,a,t,c,h} ermitteln
  - z.B. ei und ie treten etwa gleich oft auf, ea ist selten (und liefert a)
  - c und h treten einzeln selten, aber gemeinsam oft auf
- hat nun schon 2/3 des Textes. Rest durch geschicktes Probieren

(14) Vigenère-KS

- blockweise Addition des Schlüsselwortes:

3.1.1 Definition Sei m eine feste, positive Zahl. Im **Vigenère-Kryptosystem** seien $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}/26\mathbb{Z})^m = \underbrace{\mathbb{Z}/26\mathbb{Z} \times \cdots \times \mathbb{Z}/26\mathbb{Z}}_{\text{m mal}}$. Für einen Schlüssel $K = (k_1, \ldots, k_m)$ in $\mathcal{K}$ definieren wir

$$e_K : \mathcal{P} \to \mathcal{C} \text{ durch } e_k(x_1, \ldots, x_m) = (x_1 + k_1, x_2 + k_2, \ldots, x_m + k_m)$$

und

$$d_K : \mathcal{C} \to \mathcal{P} \text{ durch } d_k(y_1, \ldots, y_m) = (y_1 - k_1, y_2 - k_2, \ldots, y_m - k_m).$$

Dabei finden alle Rechnungen in $(\mathbb{Z}/26\mathbb{Z})^m$ statt.

Offenbar gilt für alle $x = (x_1, \ldots, x_m) \in (\mathbb{Z}/26\mathbb{Z})^m$, dass $d_K(e_K(x)) = x$ ist.

(15) Kasiski-Test

- Eigenschaft Vigenère: Kurze, häufig vorkommende Wörter werden u.U. mit demselben Teil des Schlüsselwortes mehrmals verschlüsselt
- sucht Wiederholungen kurzer Buchstaben-Folgen im Geheimtext → Schlüsselwortlänge ist ein Teiler des Abstandes
- → der KT liefert die Schlüsselwortlänge bis auf Vielfache

(16) Friedman-Test

- als Indiz, ob mono- oder polyalphabetisches KS.

3.2.3 Definition Die Zahl $I(\mathbf{x}) = \dfrac{\sum_{i=0}^{25} n_i(n_i - 1)}{n(n-1)}$ wird der **(Friedmansche) Koinzidenzindex** von $\mathbf{x}$ genannt.

- = $\dfrac{\text{\# Buchstaben-Paschs}}{\text{\# beliebige Buchstabenpaare}}$ = Wahrsch., dass bei 2mal ziehen 2mal derselbe Buchstabe gezogen wird
- mit p_i := Wahrscheinlichkeit für den i-ten Buchstaben im Alphabet folgt:
- Der Koinzidenzindex I(**x**) eines Textes **x**, der in etwa $\sum_{0 \le i \le 25} p_i^2$ beträgt, ist mind. 0,0385 und wird größer, wenn die Buchstaben unregelmäßig verteilt sind:
  - monoalphabetisch in deutscher Sprache: ca. 0,0762
  - monoalphabetisch in englischer Sprache: ca. 0,065
  - weicht er deutlich davon ab → Vermutung: polyalphabetisch verschlüsselt
- **Friedman-Test:** Umformen der Formel für I(x) nach *l* (Schlüsselwortlänge) liefert einen Näherungswert für l.
- Koinzidenzindex dazu benutzen, die Schlüsselwortlänge zu schätzen.
- Paare gleicher Buchstaben noch einmal anders zählen, und dabei die Schätzung aus dem Kasiski-Test einzubeziehen:

- Text schreibt man in Spalten, soviele Spalten wie mit dem KT-Test vermutet
- Dann ist jede Spalte monoalphabetisch verschlüsselt
- innerhalb einer Spalte erwartet man 7,62% Buchstabenpaschs von insgesamt n(n-l)/(2l) Paaren von Buchstaben bei denen die Buchstaben aus derselben Spalte genommen werden
- auf 2 versch. Spalten bezogen erwartet man nur unwesentlich mehr als 3,85% Buchstabenpaschs (s.oben) von insg. $n^2(l-1)/(2l)$ Paaren von Buchstaben bei denen die Buchstaben aus verschiedenen Spalten genommen werden

näherungsweise Berechnung von $I(\mathbf{x})$:

$$I(\mathbf{x}) \approx \frac{\left(\frac{n(n-l)}{2l} \cdot 0,0762 + \frac{n^2(l-1)}{2l} \cdot 0,0385\right)}{\frac{n(n-1)}{2}} = \frac{1}{l(n-1)}(0,0377n + l(0,0385n - 0,0762)).$$

-
- umformen nach l liefert den Schätzwert

(17) Hill-KS

- **Adjunktensatz**: Sei $A = (a_{ij}) \in M_{mm}(R)$. $A^{Ad} = (a'_{ij}) \in M_{mm}(R)$, wobei für alle $1 \le i,j \le m$ gilt:
  - $a'_{ij} = (-1)^{i+j} \cdot \det A_{ji}$ mit $A_{ji} \in M_{m-1,m-1}(R)$ = j-te Zeile und i-te Spalte aus A gestrichen.
  - **Satz**: Sei $A \in M_{mm}(R)$. Dann gilt $A \cdot A^{Ad} = A^{Ad} \cdot A = \det A \cdot I_m$.
  - Sei $A \in M_{mm}(R)$ invertierbar. Dann ist mit dem Adjunktensatz $A^{-1} = \det(A)^{-1} \cdot A^{Ad}$
- **Hill-KS:** Sei $m \in \mathbf{N}$.
  - $\boldsymbol{P} = \boldsymbol{C} = (\mathbf{Z}/26\mathbf{Z})^m$ Zeilenvektoren
  - $\boldsymbol{K} = Gl_m(\mathbf{Z}/26\mathbf{Z}) = \{ M_{mm}(\mathbf{Z}/26\mathbf{Z}) \mid M \text{ ist invertierbar} \}$
  - Für ein $K \in \boldsymbol{K}$ sei $e_K : (\mathbf{Z}/26\mathbf{Z})^m \to (\mathbf{Z}/26\mathbf{Z})^m$ $e_K(x) = xK$
    und $d_K : (\mathbf{Z}/26\mathbf{Z})^m \to (\mathbf{Z}/26\mathbf{Z})^m$ $d_K(x) = yK^{-1}$
- ist nicht mit einem known-Ciphertext, aber mit known-Plaintext-Angriff zu knacken.
- Oscar kennt m und hat mind. m versch.m-Tupel Geheimtext (Y) und entspr. Klartext (X)
- also XK=Y mit K = unbekannte Schlüsselmatrix.
  - X invertierbar $\Rightarrow$ $K=X^{-1}Y$ direkt berechenbar
  - X nicht invertierbar → kann Zeilen in X und Y solange mit weiteren bekannten Klartext-Geheimtext-Tupeln austauschen, bis er ein invertierbares X' findet und K berechnen kann.
  - oder dann zur Not zumindest soviele Informationen zu ermitteln, dass er K mit etwas experimentieren findet.

(18) Stromchiffren

- Schlüsselstrom $z_1\, z_2\, z_3 \ldots$ erzeugen, mit dem eine Folge $\mathbf{x} = x_1\, x_2\, x_3 \ldots$ Klartextsymbole verschlüsselt wird: $\mathbf{y} = y_1\, y_2\, y_3 \ldots = e_{z1}(x_1)\, e_{z2}(x_2)\, e_{z3}(x_3)$
- Schlüssel z_i wird mit einer Fkt. f_i erzeugt: $z_i = f_i(K, x_1, \ldots, x_{i-1})$
- Verschlüsselungsvorgang: nacheinander berechnen: $z_1, y_1, z_2, y_2, \ldots$
- Entschlüsselungsvorgang: nacheinander berechnen: $z_1, x_1, z_2, x_2, \ldots$
- **Definition:** Eine **Stromchiffre** ist ein Tupel (***P, C, K, L, F, E, D***), so dass gilt:
  - (i) ***P*** ist eine endliche Menge von **Klartexten**.
  - (ii) ***C*** ist eine endliche Menge von **Geheimtexten**.
  - (iii) ***K*** ist eine endliche Menge von **Schlüsseln**.
  - (iv) ***L*** ist eine endliche Menge, genannt das **Schlüsselstromalphabet**.
  - (v) $\boldsymbol{F} = (f_1, f_2, \ldots)$ ist der **Schlüsselstrom-Erzeuger**. Für $i \ge 1$ ist $f_i : \boldsymbol{K} \times \boldsymbol{P}^{i-1} \to \boldsymbol{L}$ eine Abbildung
  - (vi) Fü jedes $z \in \boldsymbol{L}$ gibt es
    - eine Chiffrierungsregel $e_z \in \boldsymbol{E}$, $e_z : \boldsymbol{P} \to \boldsymbol{L}$ und
    - eine Dechiffrierungsregel $d_z \in \boldsymbol{D}$, $d_z : \boldsymbol{C} \to \boldsymbol{P}$, so dass $\forall x \in \boldsymbol{P}$ gilt: $d_z(e_z(x)) = x$.
- **Selbstschlüssel-KS:** $\boldsymbol{P}=\boldsymbol{C}=\boldsymbol{K}=\boldsymbol{L}=\mathbf{Z}/26\mathbf{Z}$. Es ist $z_1=\boldsymbol{K}$ und $z_i = x_{i-1}$ für $i \ge 2$. Für $0 \le z \le 25$ wird definiert:
  $e_K(x) = x+z \bmod 26$ und $d_K(y) = y-z \bmod 26$

(19) One-Time Pad

- Sei $n \ge 1$ und $\boldsymbol{P} = \boldsymbol{C} = \boldsymbol{K} = (\mathbf{Z}/2\mathbf{Z})^n$. Sei $K = (K_1, \ldots, K_n) \in \boldsymbol{K}$ und sei $\mathbf{x} = (x_1, \ldots, x_n) \in \boldsymbol{P}$. Dann ist
  $e_K(x) = (x_1 + K_1, \ldots, x_n + K_n) \bmod 2$ und $d_K(x) = (y_1 + K_1, \ldots, y_n + K_n) \bmod 2$
- Schlüssel
  - muss mindestens so lang wie die zu verschlüsselnde Nachricht sein

- muss wirklich absolut zufällig sein, also alle Einträge in *K* mit derselben Wahrscheinlichkeit
- darf nur einmal benutzt werden

- Verabredung eines Schlüssels der Länge *n*, anstelle der Übertragung einer Nachricht der Länge *n*.
  - Vorteil, weil Zeitpunkt der Schlüsselverabredung frei wählbar, wenn sie sich sicher fühlen

---

EXTRA: ÜBERSICHT ALGEBRAISCHE STRUKTUREN

Verknüpfung: eine Abbildung $M \times M \to M$.
Mengen mit einer Verknüpfung:

- **Halbgruppe**: Verknüpfung ist assoziativ: (a*b)*c = a*(b*c)
  Beispiele: $(\mathbb{N},+)$
- **Halbgruppe mit neutralem Element**: $\exists$ e$\in$M sodass e*a=a*e= a für alle a$\in$M
  Beispiele: $(\mathbb{N},+)$, $(M_{nn}(\mathbb{K}), *_{\text{Matrizenmult.}})$
  - **Gruppe**: alle a$\in$M sind invertierbar
    - **Abelsche Gruppe**: Die Verknüpfung ist kommutativ: a*b=b*a
      Bsp: $(\mathbb{Z},+)$ – neutrales Element: 0; zu *a* ist –*a* invers
      $(\mathbb{Z}/n\mathbb{Z},+)$ – neutrales Element: 0; zu *a* ist *n*–*a* invers

Menge R mit 2 Verknüpfungen, i.d.R. (R,+,*):

- **Ring**: (R,+) ist abelsche Gruppe mit neutralem Element 0 und
  (R,*) ist Halbgruppe mit neutralem Element *e* oder 1
  es gelten die Distributivgesetze, also ein- und ausklammern geht beliebig.
  Bsp: $(M_{nn}(\mathbb{K}), +, *_{\text{Matrizenmult.}})$, *R*={0}, für jeden Ring *R* ist der Polynomring *R*[*T*] ein Ring.
  - **kommutativer Ring:** * ist kommutativ: a*b=b*a
    Bsp: $(\mathbb{Z},+,*)$, $(\mathbb{Z}/n\mathbb{Z},+,*)$, *R*[*T*] ist kommutativ, falls R es ist
    nicht aber $M_{nn}(\mathbb{K})$ für n≥2
    - **Integritätsbereich:** wenn folgende Aussage gilt: a*b=0 $\Rightarrow$ a=0 oder b=0
      Bsp: $(\mathbb{Z},+,*)$, $\mathbb{K}(T)$ mit Polynomaddition und –multiplikation
      i.d.R. kein Integritätsbereich: $\mathbb{Z}/n\mathbb{Z}$
      - **Körper**: mit mind. 2 Elementen 0 (neutrales El. +) und 1 (neutrales Elem. *), sodass *R*\{0} invertierbar bezüglich * ist. (d.h. *R*\{0} ist Gruppe)
        Bsp.: jeder endliche Integritätsbereich ist Körper, z.B. $\mathbb{Z}/p\mathbb{Z}$ mit p Primzahl; $\mathbb{Q}$, $\mathbb{R}$.
  - (R,+,*) ist **Schiefkörper**, falls (*R*\{0},*) eine Gruppe ist.
    - **Körper**: sind alle **kommutativen** Schiefkörper

KE2

(20) Begriffe:

- **Ordnung** einer Gruppe = # Elemente
- **Untergruppe**: enthält das neutr. Elem. *e* und ist mit * wieder Gruppe. Schreibweise: H < G
- **Untergruppenkriterium**: H < G $\Leftrightarrow$ $ab^{-1} \in H$ f. alle a,b $\in$H (multpl.) $\Leftrightarrow$ a – b $\in$H (add.)
- von a erzeugte Untergruppe <a> = { a^i | i $\in$**Z** } (mult.) = { ai | i $\in$**Z** } (add.).
  - ord(a) = |<a>|
  - falls ord(a) = n < ∞, dann ist <a> = { a^0, …, a^{n-1} }.
- Sei H eine UG von G.
  - Seien a,b$\in$G. Ha = Hb $\Leftrightarrow$ $ab^{-1} \in H$. aH = bH $\Leftrightarrow$ $a^{-1}b \in H$.
  - wenn Ha = aH für alle a$\in$G, heißt H **Normalteiler** von G, Schreibweise H $\lhd$ G.
    - $\Leftrightarrow$ für alle a$\in$G: aHa^{-1} = H.
  - **Linkskongruenz** modulo H: (a,b) $\in R_H \Leftrightarrow$ a= bh für ein h $\in$H. (mult.)
  - [Links-]**Äquivalenzklassen** bezüglich R_H sind: aH = { ah | h $\in$H }, a $\in$G
    bzw. a + H = { a+h | h $\in$H }, a $\in$G
  - [G : H] := Anzahl der Linksnebenklassen modulo H := der „Index“ von H in G
- Satz von **Lagrange**: Sei Gruppe G endlich mit Untergruppe H von G. $\Rightarrow$ |G| = [G : H] · | H |
- Ring R. R^x = { bez. * invertierbare Elem. in R } := **Einheitengruppe**

- Sei $m \in \mathbf{N}$, $m>1$. Die Ordnung der Gruppe $(\mathbf{Z}/m\mathbf{Z})^x$ wird mit $\varphi(m)$ bezeichnet, zusätzlich sei $\varphi(1)=1$. Diese Funktion φ: $\mathbf{N} \to \mathbf{N}$ heißt die **Eulersche φ-Funktion.**
  - sie ist **multiplikativ**, d.h. $\varphi(mn) = \varphi(m)\,\varphi(n)$ für alle m,n.

(21) Satz von Euler

- Seien $x,n \in \mathbf{N}$ mit $ggT(x,n)=1$. Dann gilt $x^{\varphi(n)} \equiv 1 \pmod n$, also $n \mid (x^{\varphi(n)} - 1)$

(22) kleiner Satz von Fermat und seine Erweiterung

- Primzahl p; $a \in \mathbf{N}$ mit $ggT(a,p) = 1$ $\Rightarrow$ $p \mid (a^{p-1} - 1)$ bzw. $a^{p-1} \equiv 1 \pmod p$
- Primzahlen p,q; $c \in \mathbf{N}$ mit $c \bmod \varphi(pq) = 1$ $\Rightarrow \forall a \in \mathbf{Z}$: $pq \mid (a^c - a)$ bzw. $a^c \equiv a \pmod{pq}$

(23) Zyklische Gruppen

- Gruppe G heißt **zyklisch**, wenn es ein a in G gibt, so dass $\langle a \rangle = G$ ist. So ein a heißt **erzeugendes Element** von G genannt
- Zyklische Grupen sind abelsch.
- Jede Gruppe G mit ord(G)=prim ist zyklisch.
- G zyklisch. Wenn $|G| = \infty \Rightarrow G \simeq (\mathbf{Z},+)$. Wenn $|G| = n < \infty \Rightarrow G \simeq (\mathbf{Z}/n\mathbf{Z},+)$.
- G zyklisch und endlich. Es gibt $\varphi(|G|)$ erzeugende Elemente von G.
- **K** ein Körper.
  - Jede endl. Untergruppe G von $\mathbf{K}^x$ ist zyklisch .

(24) Endlicher Körper, primitives Element

- **F** ein endlicher Körper mit q Elementen.
  - $\mathbf{F}^x = \mathbf{F}\backslash\{0\}$ [1] ist zyklisch und hat [erwartungsgemäß] $\varphi(|q-1|)$ erzeugende Elemente.
  - Ein erzeugendes Element von $\mathbf{F}^x$ heißt **primitives Element** in **F**.

KE3

(25) Ideale und Faktorringe

- $(R,+,*)$ ein Ring. **Ideal** ist eine Untergruppe von $(R,+)$, sodass für alle $a \in I$ und $b \in R$: $ab, ba \in I$. Schreibweise: $I \lhd R$
  - Bsp. $n\mathbf{Z} \lhd \mathbf{Z}$
  - $(f) \lhd \mathbf{K}[T]$ mit $f = \sum_{i=0}^{n} a_i T^i$
  - **Z** ist zwar ein Unterring, aber kein Ideal in **Q**.
  - Falls $1 \in I \Rightarrow I = R$.
  - Ideale sind Normalteiler von $(R,+)$.
- $(R,+,*)$ ein Ring mit Ideal $I \lhd R$. Klasseneinteilung mit Nebenklassen modulo I: $[r] = r + I = \{ r+s \mid s \in I \}$
  - $a,b \in R$ heißen **kongruent modulo** I $\Leftrightarrow a \equiv b \pmod I \Leftrightarrow [a] = [b] \Leftrightarrow a-b \in I$.
  - Die Menge der Nkl. heißt **Faktor-** bzw. **Restklassenring von R modulo I:** $R/I = \{ [r] \mid r \in R \}$

(26) Primring

- Im folgenden beizeichnet e das neutr. Element der Multiplikation.
- Sei ϕ: $\mathbf{Z} \to R$ mit $\phi(n) = ne$. Dann gilt:
  - $\phi(m+n) = (m+n)e = me + ne = \phi(m) + \phi(n)$
  - $\phi(mn) = (mn)e = (mn)e^2 = \phi(m)\phi(n)$
  - $\phi(1) = 1e = e$
  - ϕ ist ein Ringhomomorphismus[2] mit $Bild(\phi) = \mathbf{Z}e = \{ ne \mid n \in \mathbf{Z} \}$
- Der Unterring **Z**e von R heißt **Primring** von R und wird mit P(R) bezeichnet.
  - ist der kleinste Unterring von R, in allen anderen Unterringen enthalten.
  - $P(R) = \mathbf{Z}e \simeq \mathbf{Z}/Kern(\phi)$. (Kern = Menge der Nullstellen)

(27) Charakteristik char(R):

- Es gilt stets entweder (a) oder (b):
  - (a) $P(R) = \mathbf{Z}$. Dann gilt $ne \neq 0$ für alle $n \neq 0$. Wir sagen: $char(R)=0$.
  - (b) $P(R) = \mathbf{Z}/m\mathbf{Z}$ für ein $m>0$. Dann ist m die kleinste positive Zahl mit $me=0$. $char(R)=m$.

---

[1] Denn $F\backslash\{0\}$ ist bezüglich * invertierbar, also eine Gruppe $\Rightarrow F\backslash\{0\} = F^x$.

[2] d.h. für alle $r_1, r_2 \in R$ gilt: $\phi(r_1+r_2) = \phi(r_1) + \phi(r_2)$ und $\phi(r_1 r_2) = \phi(r_1)\phi(r_2)$ und $\phi(1) = 1$.

- gibt an, wie oft man das multiplikative neutrale Element (1) eines Körpers/Rings addieren muss, um das additive neutrale Element (0) zu erhalten. Ist dies nicht möglich, so ist die Charakteristik 0.
- Körper haben stets Charakteristik 0 oder eine Primzahl (beides ist auch für unendl. Körper möglich).

(28) Endliche Körper

- Charakteristik ist eine Primzahl p.
- haben stets die Ordnung p^n (diese Primzahl p =Charakteristik dieses Körpers)

(29) Primelement

- $a,b \in R$ heißen **assoziiert**, wenn es eine Einheit $z \in R^x$ gibt, sodass a=bz.
- $c \in R$ heißt **Primelement**, wenn es keine Einheit ist und nur durch Einheiten und die zu c assoziierten Elemente teilbar ist.
    - Bsp: in **Z** sind die Einheiten +1, -1; Primelemente sind die Primzahlen und deren Negatives.

(30) RSA Vorgehensweise

- A wählt große Primzahlen p und q.
- A bildet m=pq und $\varphi(m) = (p-1)(q-1)$ und wählt ein e mit $ggT(e, \varphi(m))=1$. → e invertierbar in $\mathbf{Z}/\varphi(m)\mathbf{Z}$
    - **Öffentlicher Schlüssel**: (m,e) zum Verschlüsseln
    - **Privater Schlüssel**: p, q und d, sodass ed=1 in $\mathbf{Z}/\varphi(m)\mathbf{Z}$ ($\Rightarrow$ es gilt $ed = r\,\varphi(m)+1$ für ein $r \in \mathbf{Z}$) zum Entschlüsseln
    - **Klartextmenge und Geheimtextmenge: $P = C = \mathbf{Z}/m\mathbf{Z}$**
- **Chiffrierung** der Nachricht $x \in \mathbf{Z}/m\mathbf{Z}$: $y = x^e$ in $\mathbf{Z}/m\mathbf{Z}$.
- **Dechiffrierung**: bilde $y^d = (x^e)^d = x^{ed} = x$ (→ Erweiterung kleiner Satz Fermat nachvollziehen).

(31) effiziente Berechnung der einzelnen Komponenten des RSA-Kryptosystems

- Bestimmung p und q: effiziente Primzahltests
- Bestimmung e: e wird durch einen Zufallsgenerator vorgeschlagen und solange um 1 erhöht, bis $ggT(\varphi(m))=1$ erfüllt ist (Euklid).
- d ist das Inverse zu e und wird mit dem effizienten [erweiterten] eukl. Alg. berechnet.
- Der Euklidische Algorthmus und seine Erweiterung sind effizient.
- Chiffrierung und Dechiffrierung erfolgt effizient mittels → wiederholtem Quadrieren.

(32) warum ist RSA sicher?

- Oscar kennt m=pq, aber nicht die Faktoren.
- Es gibt keine effizienten Faktorisierungsalgorithmen, die eine natürl. Zahl in ihre Primfaktoren zerlegen.
    - es gibt aber schlechtere Alg., die das tun, daher muss die Schlüssellänge groß genug gewählt werden (BSI: mind. 1024Bit).
- Vermutung: Außer der Faktorisierung von m gibt es keinen anderen Weg, um das RSA-KS zu brechen

KE4

(33) Effizienter Algorithmus

- **Algorithmus** := Anleitung, wie eine Aufgabe oder ein Problem in endlicher Zeit zu lösen ist.
- **Laufzeit**, die ein Algorithmus benötigt, ist die Anzahl der Schritte, die ausgeführt werden müssen. (meistens abhängig von der Eingabegröße).
- Kryptologie: als Eingabegröße dient i.d.R. die Anzahl der Bits in der Eingabe benutzt.
- Als Schritte in einem Algorithmus werden Berechnungen mit einzelnen Bits gezählt und keine Vergleiche, weil die sehr schnell gehen.
- **Addition** von 2 n-Bit-Zahlen: dauert 2n+1 Schritte, also $\boldsymbol{O}(n)$.
- **Multiplikation** einer n-Bit-Zahl mit einer m-Bit-Zahl, $n \geq m$: höchstens $m(4n+1)$ Schritte, also $\boldsymbol{O}(mn)$.
- **Substraktion** einer m-Bit-Zahl von einer n-Bit-Zahl, $1 \leq m < n$: 2n-1 Schritte, also $\boldsymbol{O}(n)$.
- **O-Notation**:
    - Seien $f,g : \mathbf{N}_0^r \to \mathbf{R}$. Gibt es $B,C \in \mathbf{N}_0$, sodass für alle $(n_1,\ldots,n_r) \in \mathbf{N}_0^r$ mit $n_i \geq B$ für alle $1 \leq i \leq r$ gilt:
        - $f(n_1,\ldots,n_r) > 0$ und $g(n_1,\ldots,n_r) > 0$ und
        - $f(n_1,\ldots,n_r) < C \cdot g(n_1,\ldots,n_r)$
    - dann sagen wir, „f ist durch g beschränkt“ und schreiben $f=\boldsymbol{O}(g)$ („f ist gleich Groß-O von g).
    - Jenachdem, wie groß B und C sind, kann es sein, dass 2 Algorithmen zwar für kleine Eingaben ähnlich schnell sind, sich aber asymptotisch (d.h. für sehr große Eingaben) deutlich unterscheiden.

- ein Alg. heißt **effizient**, wenn er polynomial ist, d.h., wenn die Anzahl der benötigten Bitoperationen auch bei großen Eingaben durch ein Polynom beschränkt ist.
- **Division mit Rest** einer n- und einer m-Bit Zahl, m<n: Es sind max. n Substraktionen nötig $\Rightarrow$ $\boldsymbol{O}(n^2)$.

(34) Wie geht wiederholtes Quadrieren? b^n mod m

- b, n, m haben die Bitlängen[3] $[\log_2 b]+1$, $[\log_2 n]+1 = k+1$ und $[\log_2 m]+1$
- naiv: (n-1) Multiplikationen, d.h. $\boldsymbol{O}(2^{[\log n]+1})$ und ist nicht mehr effizient.
- Besser: n in 2erpotenzen zerlegen: $n = (\alpha_k \dots \alpha_0)_2 = \sum_{1\le i\le k} \alpha_i 2^i$. Dann ist $b^n \bmod m = \prod_{1\le i\le k} b\text{^}(\alpha_i 2^i)$
- Algorithmus: $a_0 := 1$; $b_0 := b$
  - für i = 0 … k:
    - wenn a_i ==1: $a_{i+1} := a_i b_i \bmod m$
    - sonst: $a_{i+1} := a_i \bmod m$
  - übergebe a_{k+1}
- Es ist $a_{k+1} \equiv b^n \pmod m$, denn es gilt für alle $1\le i\le k$: $a_{i+1} = b\text{^}(a_0 2^0 + \dots + a_i 2^i) \bmod m$
  und $b_{i+1} = b\text{^}(2^{i+1}) \bmod m$
- Weil Multiplikation und Reduktion modulo m (=Division mit Rest) effizient sind und es nur k+1 Schleifendurchläufe gibt, ist auch das Wdh. Quadrieren effizient.

(35) Carmichael-Zahl und Pseudoprimzahl

- **Fermat:** Für Primzahl p und ein $a\in\mathbf{N}$ mit ggT(a,p) = 1 $\Rightarrow$ $p|(a^{p-1} -1)$ bzw. $a^{p-1} \equiv 1 \pmod p$
- Umkehrung gilt nicht! Sei $a\in(\mathbf{Z}/\tilde{p}\mathbf{Z})^x$. Eine <u>zusammengesetzte</u> Zahl $\tilde{p}\in\mathbf{N}$ mit $a^{\tilde{p}-1} \bmod \tilde{p} = 1$ heißt **Pseudoprimzahl zur Basis** a.
  - Entweder : $\tilde{p}\in\mathbf{N}$ ist eine PPZ für <u>alle</u> Basen $a\in(\mathbf{Z}/\tilde{p}\mathbf{Z})^x$: so ein $\tilde{p}$ heißt **Carmichael-Zahl**.
  - oder : $\tilde{p}\in\mathbf{N}$ ist eine PPZ für höchstens die Hälfte aller $a\in(\mathbf{Z}/\tilde{p}\mathbf{Z})^x$.
- Satz v. **Korselt**: sei $n\in\mathbf{N}$ ungerade und zusammengesetzt. n ist eine Carmichael-Zahl $\Leftrightarrow$
  - n nicht durch eine Quadratzahl teilbar
  - und: für alle Primteiler p von n gilt: $p - 1|n - 1$.
- Carmichael-Zahlen sind das Produkt von mind. 3 Primzahlen

(36) Fermat-Test

- Sei $n\in\mathbf{N}$. Wähle zufällig $a\in\mathbf{N}$ mit $0 < a < n$.
  - ggT(a,n)≠1 $\Rightarrow$ a ist Teiler von n, und n ist zusammengesetzt.
  - ggT(a,n)=1 $\Rightarrow$ dann berechne $a^{n-1} \bmod n$
  - $a^{n-1} \bmod n \neq 1 \Rightarrow$ n ist zusammengesetzt.
  - $a^{n-1} \bmod n = 1 \Rightarrow$ n „wahrscheinlich prim“. D.h. n ist entweder
    - prim
    - oder eine Carmichael-Zahl [n ist PPZ für alle Basen]
    - oder zusammengesetzt, und die Wahrscheinlichkeit ein solches a zu wählen, war höchstens ½. [PPZ höchstens die Hälfte aller a, auch das gewählte].
- Gäbe es keine Carmichael-Zahlen, so wäre die Wahrsch., das der Test ausgibt, eine Zahl sei prim, obwohl sie es nicht ist, höchstens ½.
  - wenn man den Test k-mal mit jeweils zufällig gewähltem a wiederholt, kann man die Wahrsch. reduzieren auf höchstens $(½)^k$. I.d.R. wählt man $20\le k\le 50$.
- Test ist effizient, denn er berechnet den ggT und eine Potenz in $(\mathbf{Z}/n\mathbf{Z})^x$.

(37) starke Pseudoprimzahl

- Für Primzahlen gilt: $p=s*2^r + 1$ (mit s ungerade). Die Umkehrung gilt nicht!
- Eine ungerade, zusammengesetzte Zahl $\tilde{p}\in\mathbf{N}$ heißt **starke PPZ zur Basis a**$\in(\mathbf{Z}/\tilde{p}\mathbf{Z})^x$, wenn $\tilde{p}=2^r s+1$ (s ungerade) und wenn entweder:
  - $a^s \equiv 1 \pmod{\tilde{p}}$ gilt, $\Rightarrow (x_0, \dots, x_r) = (1,\dots,1)$
  - oder es gibt ein i mit $0\le i<r$, sodass $a\text{^}(2^i s) \equiv -1 \pmod{\tilde{p}}$ bzw. $a\text{^}(2^i s) \equiv \tilde{p}-1 \pmod{\tilde{p}}$ gilt.
    $\Rightarrow a\text{^}(2^j s)\equiv 1 \pmod{\tilde{p}}$ für j>i, d.h. $(x_0, \dots, x_r) = (*,\dots,*,\tilde{p}-1,1,\dots,1)$
- Ein ungerades und zusammengesetztes $\tilde{p}\in\mathbf{N}$ ist starke PPZ für höchstens die Hälfte aller $a\in(\mathbf{Z}/\tilde{p}\mathbf{Z})^x$.

(38) Rabin-Miller-Test

- Sei $n \in\mathbf{N}$ ungerade. Wähle zufällig $a\in\mathbf{N}$ mit $1 < a < n$.
  - ggT(a, n)≠1 $\Rightarrow$ n ist zusammengesetzt.

---

[3] [] ist die Gaußklammer, die den eingeschlossenen Wert zur nächstkleineren ganzen Zahl rundet.

- ggT(a, n)=1 ⇒ dann berechne r,s ∈**N** so, dass $n-1=2^r s$ und s ungerade ist.
- Berechne die Folge $(x_0, \ldots, x_r)$ mit $x_i = a^{(2^i s)} \bmod n$
- Falls $(x_0, \ldots, x_r)$ = (*,…,*) oder (*,… ,*,1,… ,1) oder (*,…,*,n-1) wobei *≠1 und *≠n-1 ⇒ n ist zusammengesetzt
- Falls $(x_0, \ldots, x_r)$ = (1,… ,1) oder (*, … , *, n-1, 1, … , 1): n ist „wahrscheinlich prim“, d.h.
  - entweder: n ist prim
  - oder: n ist zusammengesetzt [eine starke PPZ] und die Wahrscheinlichkeit ein solches a zu wählen, war höchstens ½ [eigentlich sogar ¼].

- d.h. nach *k* Wiederholungen des Tests mit unabhängig voneinander gewählten *a* ist die W., dass *n* zusammengesetzt ist und dies nicht erkannt wird, höchstens $(½)^k$ [eigentlich sogar $(¼)^k$].
  - es gibt kein Analogon zu den Carmichael-Zahlen und das ist super.
  - das ist der in der Praxis am häufigsten benutzte Test (MuPad, Maple)
- fkt. ähnlich wie Solovay-Strassen, ist aber schneller, und Irrtumswahrscheinlichkeit geringer.
- Test ist effizient, denn:
  - es werden r+1 Werte x_i effizient durch wdh. Quadrieren berechnet und $r \leq \log_2 n$.
  - r wird mit Division durch 2 berechnet bis das Ergebnis ungerade ist. Dazu benötigt man r Divisionen durch 2 ⇒ r lässt sich effizient berechnen.

(39) Legendre- und Jacobi-Symbol, Eulersche Pseudoprimzahl

- p > 2 eine Primzahl. Eine Quadratzahl in $(Z/pZ)^x$ heißt **quadratischer Rest** modulo p. Eine Zahl, die keine Quadratzahl ist, wird **quadratischer Nichtrest** modulo p genannt.
  - Die Hälfte der Zahlen $(Z/pZ)^x$ sind quadratische Reste.
- **Legendre-Symbol** für a∈**Z** und Primzahl p>2:

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & \text{falls } p \mid a \text{ gilt.} \\ 1, & \text{falls } a \bmod p \text{ quadratischer Rest modulo } p \text{ ist.} \\ -1, & \text{falls } a \bmod p \text{ quadratischer Nichtrest modulo } p \text{ ist.} \end{cases}$$

  - [Für Primzahlen p] gilt (a|p) ≡ a^((p-1)/2) mod p
- Erweiterung: **Jacobi-Symbol** für a∈**Z** und ungerades n∈**N**, n>2 mit Primzahlzerlegung $n=p_1\alpha^1 \ldots p_r\alpha^r$:

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1} \cdot \ldots \cdot \left(\frac{a}{p_r}\right)^{\alpha_r}.$$

  - stimmt für primes n mit Legendre überein
  - für zusammengesetztes n keine Aussage mehr möglich, ob a in $(Z/nZ)^x$ quadratischer Rest ist.
- ungerades, zusammengesetztes p̃∈**N**, p̃>2 und a∈(Z/ p̃Z)^x. p̃ heißt **Eulersche PPZ** zur Basis a, wenn (a|p̃) ≡ a^((p̃-1)/2) mod p̃
- Ein ungerades, zusammengesetztes p̃∈**N** ist Eulersche PPZ für höchstens die Hälfte aller a∈(**Z**/p̃**Z**)^x.

(40) Solovay-Strassen-Test

- war erste effiziente probabilistische Primzahltest
- Sei n∈**N**, n>2 ungerade. Wähle zufällig a∈**N** mit 1 < a < n.
  - ggT(a, n)≠1 ⇒ n ist zusammengesetzt.
  - ggT(a, n)=1 ⇒ dann berechne a^((n-1)/2) mod n und (a|n).
  - a^((n-1)/2) ≢ (a|n) $_{(\text{mod } n)}$ ⇒ n ist zusammengesetzt
  - a^((n-1)/2) ≡ (a|n) $_{(\text{mod } n)}$ ⇒ n ist „wahrscheinlich prim“, d.h.
    - entweder: n ist prim
    - oder: n ist zusammengesetzt [Eulersche PPZ] und die Wahrscheinlichkeit ein solches a zu wählen, war höchstens ½.
- d.h. nach *k* Wiederholungen des Tests mit unabhängig voneinander gewählten *a* ist die W., dass *n* zusammengesetzt ist und dies nicht erkannt wird, höchstens (½).
- Test ist effizient.

KE5

(41) Endliche Körper

- Bsp. für Körper K, die nicht von der Form $\mathbf{F}_p$ (p Primzahl) sind:
  - $\mathbf{K}[T]/(f)$ ist Körper, wenn f irreduzibel in $\mathbf{K}[T]$ ist und enthält alle Polynome mit Grad kleiner als f: $\mathbf{K}[T]/(f) = \{ r \in \mathbf{K}[T] \mid \mathrm{Grad}(r) < \mathrm{Grad}(f) \}$
  - es gibt p^n Polynome mit Grad<n in $\mathbf{F}_p[T]$, denn für jeden Koeffizienten a_i gibt es p Möglichkeiten, z.B. $\mathbf{F}_2[T]/(T^3+T+1)$ hat 8 Elemente

(42) Primkörper := ein Körper **K**, der keine Unterkörper mehr enthält, die echte Teilmengen von **K** sind.

(43) Grad einer Körpererweiterung L : K

- es gilt char(**K**) = char(**L**) und **P**(**K**) = **P**(**L**) (die zu **K** bzw. **L** gehörenden primkörper sind gleich).
- **L** ist ein **K**-Vektorraum, dessen Dimension ist der **Grad** der Körpererweiterung ist $[\mathbf{L}:\mathbf{K}] = \dim_{\mathbf{K}}\mathbf{L}$
  - z.B. $[\mathbf{C}:\mathbf{R}] = 2$, denn $(1,i)$ ist eine Basis von **C** über **R**.
  - $[\ \mathbf{F}_2[T]/(T^2+T+1) : \mathbf{F}_2\ ] = 2$ mit Basis $([T], [1])$.
- a heißt **algebraisch** über **K**, wenn a Nullstelle eines Polynoms $f \in \mathbf{K}[T]$ ist.
- Körpererweiterung **L : K** heißt **algebraisch**, wenn jedes $a \in \mathbf{L}$ algebraisch über **K** ist.

(44) Zusammenfassung Eigenschaften endlicher Körper

- Zu jeder Primzahl p gibt es den Körper $\mathbf{F}_p$ mit p Elementen.
- Charakteristik eines endlichen Körpers ist immer eine Primzahl p.
- Jeder endliche Körper ist eine endliche Erweiterung seines Primkörpers.
- Ist K ein endlicher Körper mit char(K) = p, dann ist die Abbildung $a \longmapsto a^p$ für alle $a \in \mathbf{K}$ ein Automorphismus.
- ein endl. Körper hat immer p^n Elemente, wobei p := Charakteristik von **K** und n := Grad der Körpererweiterung von **K** über seinem Primkörper.
- **K** ein endlicher Körper $\Rightarrow (K^x, *)$ ist zyklisch $\Rightarrow$ K ist eine einfache algebraische Erweiterung seines Primkörpers **P**, nämlich **P**(a),[4] wobei a ein primitives Element von $\mathbf{K}^x$ ist.
- $\mathrm{char}(\mathbf{K}) = p \Rightarrow \forall a,b \in \mathbf{K}: \ (a + b)^p = a^p + b^p$

(45) Definition diskreter Logarithmus

- Sei **K** endlicher Körper und $g \in \mathbf{K}^x$ sei ein primitives Element von $\mathbf{K}^x$. Gegeben sei $a \in \mathbf{K}^x$. Bestimme $x \in \mathbf{Z}$ mit $a=g^x$. x ist der **diskrete Logarithmus** von a zur Basis g.
- Wikipedia: der diskrete Log. ist die kleinste Lösung x der Gleichung $g^x \equiv a \bmod p$ [→ Spezialfall $\mathbf{K}=\mathbf{F}_p$]
- Bsp: 2 ist ein primitives Element in $\mathbf{F}_{37}$, gesucht ist $\log_2 15$, also x für $15=2^x$.
  - man kann nur ausprobieren, also man muss $2^1, 2^2, 2^3, \ldots$ in $\mathbf{F}_{37}$ berechnen.
- **DLP**: Vermutung: es gibt keinen effizienten Algorithmus zur Berechnung des diskreten Log. über endlichen Körpern.
- $\Rightarrow$ diskrete Exponentiation als „Einwegfunktion", effizient berechenbar mit wdh. Quadrieren]

(46) Diffie-Hellman-Schlüsselaustausch-Verfahren

- Dient zum Austausch eines Schlüssels für ein symmetrisches KS.
- öffentlich bekannt: ein großer endlicher Körper **K** und ein primitives Element $g \in \mathbf{K}^x$
- Bildung des Schlüssels:
  - A wählt ein zufälliges $a \in \mathbf{N}$ mit $1 \le a \le |\mathbf{K}^x|$. A berechnet und veröffentlicht dann g^a.
  - B wählt ein zufälliges $b \in \mathbf{N}$ mit $1 \le b \le |\mathbf{K}^x|$. B berechnet und veröffentlicht dann g^b.
  - Der **Schlüssel** ist dann g^{ab}, welchen A berechnen kann mit $(g^b)^a$ und B berechnet $(g^a)^b$.
- Vermutung: es gibt keine Möglichkeit, das Diffie-Hellman-Verfahren zu brechen, ohne diskrete Logarithmen zu berechnen.
- Könnte Oscar das DLP in **K** lösen, so könnte er aus g^a und g^b die Zahlen a und b und somit auch g^{ab} ermitteln.

(47) Massey-Omura-Verfahren

- öffentlich bekannt: ein großer endlicher Körper **K** mit q Elementen.
- Ziel: es soll ein $N \in \mathbf{K}$ übermittelt werden.
  - A wählt ein zufälliges $a \in \mathbf{N}$ mit $1 \le a \le q-1$ und ggT(a,q-1)=1.

---

[4] **P**(a) ist der kleinste Unterkörper von **K**, der sowohl den Primkörper **P**, als auch a enthält.

- A berechnet dann a‘ sodass $a'a \equiv 1_{(\text{mod } q-1)}$ (Erweiterter Euklid) und schickt N^a an B (was sie effizient mit wdh. Quadrieren berechnen kann).
- B wählt ein zufälliges $b \in \mathbf{N}$ mit $1 \leq b \leq q-1$ und ggT(b,q-1)=1.
- B berechnet dann b‘ mit $b'b \equiv 1_{(\text{mod } q-1)}$ (Erweiterter Euklid) und schickt $(N^a)^b$ an A zurück(was er effizient mit wdh. Quadrieren berechnen kann).
- A bildet $(N^{ab})^{a'} = (N^{aa'})^b = N^b$ (wdh. Quadrieren)und schickt N^b zurück an B.
- Nun kann B $(N^b)^{b'} = N$ berechnen (wdh. Quadrieren).

- Vermutung: es gibt keine Möglichkeit, das Massey-Omura-Verfahren zu brechen, ohne diskrete Logarithmen zu berechnen.
- Alle benötigten Daten sind effizient berechenbar.
- Könnte Oscar das DLP in **K** lösen, dann könnte er aus N^{ab} und N^a den Wert von $b = \log_{(N^a)}(N^{ab})$ ermitteln, somit auch b‘. Dann kann er, wenn er N^b kennt, auch N ermitteln.

(48) ElGamal-KS

- öffentlich bekannt: ein großer endlicher Körper **K** mit q Elementen und ein primitives Element $g \in \mathbf{K}^x$
- Ziel: es soll ein $N \in \mathbf{K}^x$ übermittelt werden.
  - B wählt ein zufälliges $a \in \mathbf{N}$ mit $1 \leq a \leq |\mathbf{K}^x| = q-1$. B berechnet und veröffentlicht dann g^a.
  - A wählt ein zufälliges $k \in \mathbf{N}$ mit $1 \leq k \leq q-1$ und schickt B das Paar (g^k, Ng^{ak}) von Elementen aus $\mathbf{K}^x$.
  - B bildet $(g^k)^{q-1-a} = g^{k(q-1)-ka} = (g^{q-1})^k g^{-ka} = g^{-ka}$ [warum?]
  - B bildet $g^{-ka}Ng^{ak} = N$.
- Vermutung: es gibt keine Möglichkeit, das ElGamal-KS zu brechen, ohne diskrete Logarithmen zu berechnen.
- Alle benötigten Daten sind effizient berechenbar.
- Könnte Oscar das DLP in **K** lösen, dann könnte er anhand von g^a, g^k und Ng^{ak} den Wert von a und b und damit N ermitteln. [wie?]

KE6

(49) Was ist elliptische Kurve

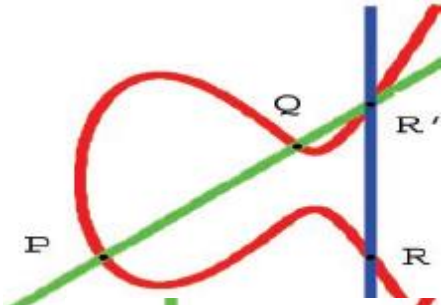
- Sei **K** endlich mit char(**K**)>3 (i.A. ist **K** ein $\mathbf{F}_p$ mit p=sehr große Primzahl) und $a,b \in \mathbf{K}$ mit $4a^3+27b^2 \neq 0$.
  Die Menge $E(a,b,\mathbf{K}) = \{ (x,y) \mid x,y \in \mathbf{K} \text{ und } y^2 = x^3 + ax + b \} \cup \{\boldsymbol{O}\}$ heißt elliptische Kurve über **K**.
- Sei **K** endlich mit char(**K**)=2 (i.A. ist dann **K** ein Körper $\mathbf{F}_{2^n}$) und $a,b \in \mathbf{K}$ mit $b \neq 0$.
  Die Menge $E(a,b,\mathbf{K}) = \{ (x,y) \mid x,y \in \mathbf{K} \text{ und } y^2 + xy = x^3 + ax^2 + b \} \cup \{\boldsymbol{O}\}$ heißt elliptische Kurve über **K**.
- $\boldsymbol{O}$ ist der „Punkt bei Unendlich“.
  - Er liegt auf jeder Geraden, die parallel zur y-Achse verläuft, und zwar im Unendlichen.
- Die Bedingung für a und b stellt sicher, dass die ell. Kurve keine Spitzen hat oder sich selbst kreuzt.

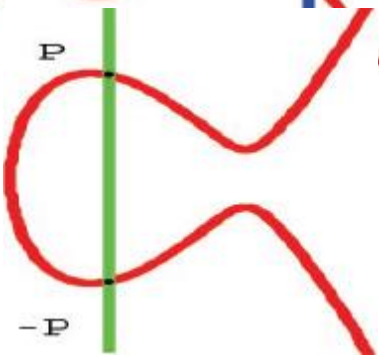
(50) Sonstige Eigenschaften elliptischer Kurven [relevant ??]

- **Satz von Helmut Hasse**: Sei $E(a,b,\mathbf{F}_q)$ eine elliptische Kurve mit N Punkten. $\Rightarrow |N-(q+1)| \leq 2\sqrt{q}$
- Sei $E(a,b,\mathbf{K})$ eine elliptische Kurve über einem endlichen Körper mit Charakteristik >3 oder =2. Dann ist $E(a,b,\mathbf{K}) \simeq (\mathbf{Z}/d_1\mathbf{Z}) \times (\mathbf{Z}/d_2\mathbf{Z})$, wobei $d_1 | d_2$. Der Fall $d_1=1$ (also $E(a,b,\mathbf{K})$ ist zyklisch) ist möglich.

(51) Addition auf elliptischen Kurven für char(K)>3

- **(R1)** $\boldsymbol{O}$ ist das neutrale El. der Addition, also $P+\boldsymbol{O} = \boldsymbol{O}+P = P$ für alle $P \in E$.
- **(R2)** zu $P=(x,y) \in E(a,b,\mathbf{K})$ (also $P \neq \boldsymbol{O}$) ist $-P=(x,-y) \in E(a,b,\mathbf{K})$ invers, also $P + (-P) = \boldsymbol{O}$.
- Addition zweier Punkte P und Q:
  - man verbindet diese beiden Punkte durch eine Gerade.
  - Diese Gerade schneidet die elliptische Kurve in einem weiteren Punkt R’
  - Die Summe P+Q ist dann der an der x-Achse gespiegelte Punkt R. Falls P=Q nimmt man stattdessen die Tangente durch P.
- Um zu einem Punkt P das Inverse –P zu erhalten, spiegelt man ihn an der x-Achse und erhält so den 2. Schnittpunkt der Kurve mit der zur y-Achse parallelen Geraden durch P





- Merkregel: Liegen 3 Punkte auf einer Geraden, gilt P+Q+R= -R+R = ***O***
- mit dieser Addition ist *E*(a,b,**K**) eine abelsche Gruppe.
- die Addition 2er Punkte ist effizient durchführbar.
- **(R3)** Sind $P = (x_1, y_1), Q = (x_2, y_2) \in E(a, b, \mathbb{K})$ mit $x_1 \neq x_2$ (also $Q \neq P$ und $Q \neq -P$), dann ist $P + Q = R = (x_3, y_3)$ mit
$$x_3 = (\frac{y_2 - y_1}{x_2 - x_1})^2 - x_1 - x_2 \text{ und}$$
$$y_3 = (\frac{y_2 - y_1}{x_2 - x_1})(x_1 - x_3) - y_1.$$
(R4) Ist $P = (x_1, y_1) \in E(a, b, \mathbb{K})$ und ist $y_1 = 0$, dann ist $2P = \mathcal{O}$. Ist $y_1 \neq 0$, dann ist $2P = R = (x_3, y_3)$ mit
$$x_3 = (\frac{3x_1^2 + a}{2y_1})^2 - 2x_1 \text{ und}$$
$$y_3 = -(\frac{3x_1^2 + a}{2y_1})x_3 + \frac{3x_1^2 + a}{2y_1}x_1 - y_1 = (\frac{3x_1^2 + a}{2y_1})(x_1 - x_3) - y_1.$$

(52) Addition auf elliptischen Kurven für char(K)=2

- Achtung! über K gilt y=-y.
- Das zu Punkt P=(x,y) Inverse ist –P =(x,x+y). Auch hier ist dies der 2. Schnittpunkt der Kurve mit der Gerade, die parallel zur y-Achse durch P verläuft.
- **(R1)**: wie oben
- **(R2)** zu P=(x,y)∈ *E*(a,b,**K**) (also P≠***O***) ist –P=(x,x+y))∈ *E*(a,b,**K**) invers, also P + (-P) = ***O***.
- **(R3)** Seien $P = (x_1, y_1), Q = (x_2, y_2) \in E(a, b, \mathbb{K})$, und es gelte $x_1 \neq x_2$. Dann ist $P + Q = R = (x_3, y_3)$ mit
$$x_3 = (\frac{y_2 + y_1}{x_2 + x_1})^2 + \frac{y_2 + y_1}{x_2 + x_1} + a + x_1 + x_2 \text{ und}$$
$$y_3 = (\frac{y_2 + y_1}{x_2 + x_1})(x_1 + x_3) + y_1 + x_3.$$
(R4) Ist $P = (x_1, y_1) \in E(a, b, \mathbb{K})$ und ist $x_1 = 0$, dann ist $2P = \mathcal{O}$. Ist $x_1 \neq 0$, dann ist $2P = (x_3, y_3)$ mit
$$x_3 = x_1^2 + \frac{b}{x_1^2} \text{ und } y_3 = (\frac{y_1}{x_1} + x_1 + 1)x_3 + x_1^2.$$
- mit dieser Addition ist *E*(a,b,**K**) eine abelsche Gruppe.
- die Addition ist effizient durchführbar.

(53) DLP auf elliptischen Kurven

- Gegeben sind ein Punkt P auf der elliptischen Kurve und ein Vielfaches aP von P mit a∈Z. Das Problem ist, a zu berechnen.
- Vermutung: Es gibt keinen effzienten Algorithmus, um das DLP füur elliptische Kurven zu lösen.
- Sogar: Vermutung: Das DLP für elliptische Kurven ist schwerer zu lösen als das für endliche Körper.
- Vorteile ggü. dem anderen DLP:
  - Körper kann hier viel kleiner gewählt werden (etwa zwischen 2^{173} und 2^{313} Elementen gegenüber sonst nötigen 2^{1024} bis 2^{4096} Elementen)
  - → weniger Speicherplatz nötig
  - → Berechnungen gehen schneller
  - Es gibt viele verschiedene elliptische Kurven über einem festen endlichen Körper → man kann den Körper fest wählen kann und die Körperarithmetik in Hardware zur Verfügung stellen kann, was nocheinmal wesentlich schneller ist.

(54) Diffie-Hellman-Schlüsselaustausch-Verfahren für ell. Kurven

- Ziel: es soll ein Punkt aus der von P erzeugten Untergruppe als Schlüssel für ein symm. KS vereinbahrt werden.
- öffentlich bekannt: ein endlicher Körper **K** mit q Elementen und char(**K**)=2 oder char(**K**)>3, eine ell. Kurve E(a,b,**K**) und ein zufälliger Punkt P∈ E(a,b,**K**).
    - P sollte so gewählt sein, dass die von P erzeugte Untergruppe in E möglichst groß ist, also etwa so groß wie E selbst
- Bildung des Schlüssels:
    - A wählt ein zufälliges a∈**N** in der Größenordnung von q (→ Satz von Hasse). A berechnet und veröffentlicht dann aP.
    - B wählt ein zufälliges b∈**N** in der Größenordnung von. B berechnet und veröffentlicht dann bP.
    - Der **Schlüssel** ist dann abP, welchen A berechnen kann mit a(bP) und B mit b(aP) mit dem Analogon zum Wiederholten Quadrieren für additive Gruppen.
- Vermutung: es gibt keine Möglichkeit, das Diffie-Hellman-Verfahren zu brechen, ohne diskrete Logarithmen für elliptische Kurven zu berechnen.
- Könnte Oscar das DLP in **K** lösen, so könnte er aus P, aP und bP den Schlüssel abP berechnen.

(55) Massey-Omura-Verfahren für ell. Kurven

- öffentlich bekannt: ein endlicher Körper **K** mit q Elementen und char(**K**)=2 oder char(**K**)>3, eine ell. Kurve E(a,b,**K**), die Anzahl N ihrer Punkte [eff. berechenbar mit „Verfahren von Schoof"]
- Ziel: es soll ein Punkt P∈ E(a,b,**K**) übertragen werden.
    - A wählt ein zufälliges a∈**N** mit $0 \leq a \leq N$ und ggT(a,N)=1.
    - A berechnet dann a' sodass $a'a \equiv 1_{(\text{mod } N)}$ (Erweiterter Euklid) und schickt aP an B.
    - B wählt ein zufälliges b∈**N** mit $0 \leq b \leq N$ und ggT(b,N)=1.
    - B berechnet dann b' mit $b'b \equiv 1_{(\text{mod } N)}$ (Erweiterter Euklid) und schickt b(aP) an A zurück.
    - A bildet a'(baP) = a'abP = bP und schickt bP zurück an B.
    - Nun kann B b'bP = P berechnen.
- Vermutung: es gibt keine Möglichkeit, das Massey-Omura-Verfahren zu brechen, ohne diskrete Logarithmen zu berechnen.
- Könnte Oscar das DLP in **K** lösen, dann könnte er aus aP, bP und abP den Wert von a und b ermitteln, daraus a' und b' (erw. Euklid) und schließlich P ermitteln.

(56) ElGamal-KS für ell. Kurven

- öffentlich bekannt: ein endlicher Körper **K** mit q Elementen und char(**K**)=2 oder char(**K**)>3, eine ell. Kurve E(a,b,**K**) und ein zufälliger Punkt S∈ E(a,b,**K**).
    - S sollte so gewählt sein, dass die von A erzeugte Untergruppe in E möglichst groß ist, also etwa so groß wie E selbst.
- Ziel: es soll ein Punkt P∈ E(a,b,**K**) übertragen werden.
    - B wählt ein zufälliges b∈**N** und veröffentlicht dann bS.
    - A wählt ein zufälliges a∈**N** und schickt B das Paar (aS, P+a(bS)).
    - B bildet b(aS) und dann P+a(bS) – b(aS) = P.
- Vermutung: es gibt keine Möglichkeit, das ElGamal-KS zu brechen, ohne diskrete Logarithmen zu berechnen.
- Könnte Oscar das DLP in **K** lösen, dann könnte er anhand von bS, aS und P+abS den Wert von a und b und damit P ermitteln. [wie?]

(57) Punkte auf einer elliptischen Kurve E(a,b,K) finden [relevant ??]

- kein effizienter deterministischer Algorithmus bekannt
- wähle zufällig x∈**K** und berechne $\alpha=x^3+ax+b$, falls char(**K**)>3, oder $\alpha=x^3+ax^2+b$, falls char(**K**)=2.
- Dann versucht man $y^2=\alpha$ bzw. $y^2+xy+\alpha=0$ zu lösen. Die Wahrsch., dass dies geht, ist etwa ½.
- Klappt das nicht, probiert man den nächsten x-Wert aus.

(58) Anforderungen an die gewählte Kurve

- Anzahl N der Punkte der Kurve muss mind. einen großen Primteiler haben, sonst ist der diskrete Logarithmus relativ leicht zu berechnen
- Ideal ist, wenn N=prim, denn dann ist jeder Punkt, der nicht $\boldsymbol{O}$ ist, ein Erzeuger der Gruppe.
    - man könnte z.b. solange zufällig Kurven erzeugen, bis man eine mit primer Ordnung findet.
    - Problem ist der hohe Rechenaufwand

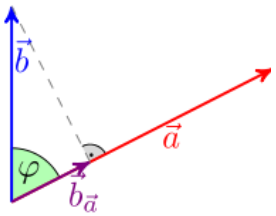
(59) Eindeutige Zuordnung einer Nachricht m und einem Punkt P_m

- wähle ein $k \in \mathbf{N}$, sodass die Irrtumswahrscheinlichkeit $(½)^k$ ist.
- Sei $0 \le m < M$ (z.B. mit M=26, wenn A,…,Z mit 0,…,25 identifiziert wird)
- **K** muss mehr als kM Elemente haben und $\varphi : \{0,\dots,kM\} \to \mathbf{K}$ muss eine injektive Abb. sein
- dann gibt es für jedes $j \in \{1,\dots,k\}$ ein $x \in \mathbf{K}$ mit $x=\varphi(mk+j)$
- für dieses x berechne $\alpha=x^3+ax+b$, falls char(**K**)>3, oder $\alpha=x^3+ax^2+b$, falls char(**K**)=2.
- Dann versucht man $y^2=\alpha$ bzw. $y^2+xy+\alpha=0$ zu lösen. Die Wahrsch., dass dies geht, ist etwa ½.
- Dann hat man entweder einen Punkt P_m gefunden, oder falls nicht, erhöht man j um 1 und probiert weiter.
- Der Empfänger kann aus $P_m=(x,y)$ wieder m berechnen, indem er das Urbild $\varphi^{-1}(x)$ berechnet, dies ist von der Form mk+j für ein $j \in \{1,\dots,k\}$. Man erhält dann $m = [\,(\varphi^{-1}(x) -1) / k\,]$.

KE7

(60) Def. Gitter

- Sei $(a_1,\dots,a_n)$ eine Basis von $\mathbf{R}^n$. Die Menge ganzzahliger Linearkombinationen der Basisvektoren $L(a_1,\dots,a_n) = \{ \sum_{1\le i\le n} \lambda_i a_i \mid \lambda_i \in \mathbf{Z} \}$ heißt **Gitter mit Basis** $(a_1,\dots,a_n)$.
- Eine Teilmenge T eines topolog. Raumes heißt **diskret**, wenn es zu jedem $t \in T$ eine Umgebung V so gibt, dass $V \cap T=\{t\}$.



(61) Determinante des Gitters

- $\det L = |\det(a_1|\dots|a_n)|$ (Matrix mit den Basisvektoren als Spalten)
- **Ungleichung von Hadamard**: $L(a_1|\dots|a_n)$. Es gilt: $|\det L| \le \prod_{1\le i\le n} ||a_i||$.
  - Falls die a_i eine Orthogonalbasis sind: $|\det L| = \prod_{1\le i\le n} ||a_i||$.

(62) Gram-Schmidt-Orthogonalbasis $(a^*_1,\dots,a^*_n)$ zu $(a_1,\dots,a_n)$

- Grundlage: Orthogonalprojektion: $b_a= b\cdot(a/||a||)\cdot(a/||a||) = \langle b,a\rangle\cdot a / \langle a,a\rangle$
- **GS-Orthogonalbasis:** Setze $a^*_1 := a_1$ und: $a^*_i := a_i - \sum_{1\le j<i} \mu_{ij}\cdot a^*_j$ wobei $\mu_{ij} := \langle a_i, a^*_j\rangle / \langle a^*_j, a^*_j\rangle$
  - $\mu_{ij}\cdot a^*_j$ ist die Orthogonalprojektion von a_i auf a_j. Dann ist $a_j \perp a_i-\mu_{ij}\cdot a^*_j$ (siehe Bild: $a \perp b-b_a$)
  - die μ_{ij} sind ein Maß für die Abweichung der Orthogonalität der Gittervektoren: $\mu_{ij} = 0 \Leftrightarrow \langle a_i, a^*_j\rangle = 0 \Leftrightarrow a_i \perp a^*_j$.

(63) Problem der kurzen Vektoren

- Problem: Für ein gegebenes Gitter L Basisvektoren zu finden, sodass $\prod_{1\le i\le n} ||a_i||$ minimal wird
- NP-hartes Problem, d.h. Vermutung: es gibt keinen polynomialen Algorithmus zur Lösung
  - sogar: Würde für ein NP-hartes Problem eine polynomiale Lösung gefunden werden, dann auch für alle anderen Probleme in NP
- Lösung: LLL-Algorithmus: bestimmt effizient eine kurze Gitterbasis, wenn auch nicht die kürzeste

(64) LLL-Algorithmus

- Der kürzeste Vektor a in einem Gitter $L(a_1,\dots,a_n)$ ist immer mindestens so lang wie ein Vektor in der zugehörigen GS-Orthogonalbasis $(a^*_1,\dots,a^*_n)$ (→ Hadamard). Problem : der kürzeste Vektor in $(a^*_1,\dots,a^*_n)$liegt i.d.R. gar nicht in L[5].
- Idee: eine „fast orthogonale" Basis zu $(a_1,\dots,a_n)$ konstruieren, die in L liegt. D.h., eine GS-Orthogonalisierung durchführen, wo die auftretenden Koeffizienten zur nächsten ganzen Zahl gerundet werden.
- Sei $(a_1,\dots,a_n)$ eine Basis von $\mathbf{R}^n$ und $(a^*_1,\dots,a^*_n)$ die zugehörige GS-Orthogonalbasis. $(a_1,\dots,a_n)$ heißt **reduzierte Basis**, wenn $||a^*_i||^2 \le 2||a^*_{i+1}||^2$ für alle $1 \le i < n$.
  - $\Rightarrow$ Sei $L(a_1,\dots,a_n)$ mit reduzierter Basis. Für alle $a \in L\setminus\{0\}$ gilt dann: $||a_1|| \le 2^{(n+1)/2} \cdot ||a||$
  - $\Rightarrow$ d.h. der erste Basisvektor einer reduzierten Basis eines Gitters L ist schlimmstenfalls um den Faktor $2^{(n-1)/2}$ größer als ein kürzester Vektor in L.
- Notation: $\lceil \mu \rfloor := [\mu + ½]$:= die ganze Zahl, die am nächsten an μ liegt.
- **Eingabe**: Linear unabhängige Spaltenvektoren $a_1,\dots,a_n \in \mathbf{Z}$
- **Ausgabe**: eine reduzierte Basis von $L(a_1,\dots,a_n)$
- **Algorithmus:**

[5] Die GS-Orthogonalbasis zu einer Gitterbasis ist i.A. keine Basis dieses Gitters.

- $(b_1,\ldots,b_n) := (a_1,\ldots,a_n)$.
- Berechne die GS-Orthogonalisierung $(b^*_1,\ldots,b^*_n)$ und M mit $M = \begin{pmatrix} 1 & \mu_{21} & \cdots & \mu_{n1} \\ 0 & 1 & \cdots & \mu_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$
- i := 2
- solange $i \leq n$:
  - für j := i-1,…,1: setze $b_i := b_i - \lceil \mu_{ij} \rfloor b_j$ und aktualisiere die GS-Orthogonalisierung
  - wenn i>1 und $||b^*_{i-1}||^2 > 2||b^*_i||^2$, dann
    - tausche b_i und b_{i-1} und aktualisiere die GS-Orthogonalisierung
    - i--
  - sonst: [die ersten i Vektoren sind jetzt fertig LLL-reduziert] i++
- übergebe $(b_1,\ldots,b_n)$.

- Berechnung LLL-reduzierter Basen von Gittern über **Q**: Die Basis von L mit einem gemeinsamen Vielfachen λ aller Nenner der Werte in a_i vervielfachen und diese vervielfachte Basis (die nun in **Z** liegt) LLL-reduzieren, das Ergebnis dann wieder durch λ teilen.

(65) Teilmengen-Summen-Problem

- seien $a_0,\ldots,a_n,s \in \mathbf{N}$. Gibt es $x_1,\ldots,x_n \in \{0,1\}$, sodass $\sum_i a_i x_i = s$ ist?
- dieses Problem ist NP-hart und es wird vermutet, dass es keine effizienten Lösungsalgorithmen gibt.
- Ist ein TMS-Problem lösbar, dann ist die Lösung i.d.R. nicht eindeutig.

(66) super-aufsteigende Folge

- ist eine Folge$(a_0,\ldots,a_n)$ natürlicher Zahlen, falls $a_i > \sum_{0\leq j<i} a_j$ für alle $1 \leq i \leq n$.
- Ein TMS-Problem mit superaufsteigender Zahlenfolge ist leicht zu lösen, denn falls, so ist es eindeutig lösbar: man packt solange das jeweils größtmögliche a_i in die Summe rein, bis man genau bei s landet (Lösung ist eindeutig) oder es reicht eben nicht und das Ergebnis ist < s.

(67) Knapsack-KS und das Problem dabei

- **privater Schlüssel**: A wählt
  - ein superaufsteigendes n-Tupel natürlicher Zahlen $(a_0,\ldots,a_{n-1})$,
  - ein $m \in \mathbf{N}$ mit $m > \sum_i a_i$ und
  - ein zu m teilerfremdes a
  - und berechnet ein $b \in \mathbf{Z}/m\mathbf{Z}$ mit $ab \equiv 1 \bmod m$.
- **öffentlicher Schlüssel**:
  - A bildet nun die [i.d.R. nicht mehr superaufsteigende] Folge $(aa_0 \bmod m,\ldots, aa_{n-1} \bmod m) = (w_0,\ldots,w_{n-1})$ und gibt sie bekannt.
- **chiffrieren**: B will A eine n-Bit-Zahl übermitteln, also ein $(x_{n-1},\ldots,x_0)$ mit $x_i \in \{0,1\}$.
  - B bildet $s = \sum_{0\leq i<n} x_i w_i$ und schickt s an A.
- **dechiffrieren**: A bildet in $\mathbf{Z}/m\mathbf{Z}$ $bs = b\cdot\sum_{0\leq i<n} x_i w_i = \sum_{0\leq i<n} x_i baa_i = \sum_{0\leq i<n} x_i a_i$
  - A muss nun ein TMS-Problem mit superaufsteigender Folge berechnen, was einfach ist.
- **Warum der Flop?** Die Vermutung, dass Oscar ein TMS-Problem auf einer nicht-superaufsteigenden Folge lösen muss, um das Knapsack-KS zu brechen, wurde wiederlegt:
  - Oscar kennt $s = \sum_{1\leq i\leq n} x_i a_i$ und die Folge $(a_0,\ldots,a_n)$
  - O bildet das Gitter $L(b_1,\ldots,b_{n+1})$ dessen Basis die Spalten der rechts abgebildeten Matrix sind $\begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ -a_1 & -a_2 & \cdots & -a_n & s \end{pmatrix}$
  - Angenommen, er findet x_i mit $x_i \in \{0,1\}$, sodass $s = \sum_{1\leq i\leq n} x_i a_i$. Dann ist $\sum_{1\leq i\leq n} x_i b_i + b_{n+1} = (x_1 \ \ldots \ x_n \ \ 0)^T$ ein Vektor in $L(b_1,\ldots,b_{n+1})$ mit sehr kurzer Länge, nämlich maximal $\sqrt{n}$.
  - zum Brechen des KS berechnet er eine LLL-reduzierte Basis von $L(b_1,\ldots,b_{n+1})$ und hofft, dass der erste Vektor der reduzierten Basis $(x_1 \ \ldots \ x_n \ \ 0)^T$ ist.
  - Das klappt so oft, dass das Knapsack-Kryptosystem als nicht sicher verworfen wurde.