



WESTFÄLISCHE  
WILHELMS-UNIVERSITÄT  
MÜNSTER



FACHBEREICH 10  
MATHEMATIK UND  
INFORMATIK

# **Elliptische Kurven und Kryptographie**

**gelesen von PD Dr. Karin Halupczok**

Zusammenfassung von Phil Steinhorst

Sommersemester 2015

Hier kommt bald ein Bild hin!

<http://wwwmath.uni-muenster.de/u/karin.halupczok/ellKKSoSe15/>

---

## Vorwort

Der vorliegende Text ist eine inhaltliche Aufbereitung zur Vorlesung Elliptische Kurven und Kryptographie, gelesen von PD Dr. Karin Halupczok an der WWU Münster im Sommersemester 2015. Der Inhalt entspricht weitestgehend dem handschriftlichen Skript, welches auf der Vorlesungswebsite bereitgestellt wird. Dieses Werk ist daher keine Eigenleistung des Autors und wird nicht von der Dozentin der Veranstaltung korrekturgelesen. Für die Korrektheit des Inhalts wird keinerlei Garantie übernommen. Bemerkungen, Korrekturen und Ergänzungen kann man folgenderweise loswerden:

- persönlich durch Überreichen von Notizen oder per E-Mail
- durch Abändern der entsprechenden TeX-Dateien und Versand per E-Mail an mich
- direktes Mitarbeiten via GitHub. Dieses Skript befindet sich im `latex-wwu`-Repository von Jannes Bantje:

<https://github.com/JaMeZ-B/latex-wwu>

## Literatur

- Blake, Seroussi, Smart: Elliptic curves in cryptography
- Menezes, van Oorschot, Vanstone: Handbook of applied cryptography
- Silverman: The arithmetic of elliptic curves
- Silverman: A friendly introduction to number theory, chap. 40-45
- Washington: Elliptic curves, number theory and cryptography
- Werner: Elliptische Kurven in der Kryptographie

## Kommentar der Dozentin

In der Vorlesung beschäftigen wir uns mit den arithmetischen und geometrischen Eigenschaften elliptischer Kurven sowie deren Anwendungen in der Kryptographie. Dabei werden wir auch einen Vergleich mit Anwendungen der elementaren Zahlentheorie in der Kryptographie ziehen. Wir verfolgen eine elementare Herangehensweise, d.h. Kenntnisse der algebraischen Geometrie und der Funktionen- oder Zahlentheorie werden nicht benötigt. Es genügen die Vorkenntnisse aus den Grundvorlesungen.

## Vorlesungswebsite

Das handgeschriebene Skript sowie weiteres Material findet man unter folgendem Link:

<http://wwwmath.uni-muenster.de/u/karin.halupczok/ellKKSoSe15/>

## Titelbild

Das fehlt noch. Über Ideen und Anregungen freue ich mich sehr!

Phil Steinhorst  
p.st@wwu.de

## Inhaltsverzeichnis

<b>0 Motivation und Einführung</b>	<b>4</b>
<b>1 Allgemeines über Kryptographieverfahren</b>	<b>7</b>
1.1 Grundlagen aus der elementaren Zahlentheorie und Gruppentheorie . . . . .	7
1.1.1 Zahlen, Darstellung von Zahlen . . . . .	7
1.1.2 Kongruenzenrechnen und die modulare Brille . . . . .	12
1.1.3 Gruppen . . . . .	16
1.2 Public-Key-Kryptographie . . . . .	21
1.2.1 RSA-Verfahren . . . . .	21
1.2.2 Diffie-Hellman-Verfahren . . . . .	22
1.2.3 ElGamal-Verschlüsselung . . . . .	24
1.3 Digitale Unterschriften . . . . .	25
1.3.1 DSA-Signatur . . . . .	25
<b>Index</b>	<b>27</b>

## 0 Motivation und Einführung

### Kryptologie

[1] Die **Kryptologie** besteht aus den folgenden beiden Gebieten:

**Kryptographie:** Studium mathematischer Techniken zur Verschlüsselung von Informationen oder geheimen Nachrichten und dem Schutz von Daten.

**Kryptoanalyse:** Beschreibung der Rückgewinnung von Informationen aus verschlüsselten Texten, der Entschlüsselung.

Oft meint man mit "Kryptographie" die Kryptologie.

Früher wurde die Kryptographie vor allem im militärischen oder diplomatischen Sektor verwendet, heutzutage steht in unserer vernetzten Welt vor allem auch der praktische Nutzen im Alltag im Vordergrund: im Internet einkaufen, Online-Banking, persönliche Daten geheimhalten bzw. Datenschutz, Nachrichten und Dokumente digital unterschreiben etc. Das Internet liefert schnelle Informationswege über öffentliche Kanäle, die leicht abgehört werden können, sodass die Verschlüsselung schützenswerter Daten unumgänglich wird. Auch die Möglichkeit zur Signierung wird nötig, weil sehr leicht Absenderangaben gefälscht werden können. Eventuell nicht abhörsichere Kanäle können außer dem Internet aber auch Briefe, Radio, Boten, etc. sein.

Bei der **symmetrischen Verschlüsselung** von Daten gibt es einen Sender  $S$  und einen Empfänger  $E$ , die sich beide auf einen gemeinsamen Schlüssel geeinigt haben, der zum Ver- und Entschlüsseln dient. Beim **Caesar-Code** z.B. ist dies die Vereinbarung, jeden Buchstaben durch den dritten nachfolgenden im Alphabet zu ersetzen, also  $A \mapsto D, B \mapsto E, C \mapsto F$ , usw. Die Entschlüsselung ist klar. Derartige **monoalphabetische Chiffrierungen**, bei der jeder Buchstabe des Alphabets stets durch denselben Geheimentextbuchstaben chiffriert wird, sind durch Häufigkeitsanalysen durch einen Angreifer, der die verschlüsselten Nachrichten abhört, sehr leicht zu entschlüsseln. Übrigens gibt es auch heutzutage PDF-Verschlüsselungsprogramme, die so arbeiten!

In dieser Vorlesung behandeln wir die heutzutage gängigen modernen Methoden, die als sicher gelten. Worauf diese starke Sicherheit beruht, hat mathematische Gründe, die wir besprechen möchten. Vor allem interessiert uns, wie und welche Mathematik in die Kryptologie kommt, sodass wir deren Verfahren verstehen können.

Die Anwendungen erfordern die Lösung folgender Probleme bei symmetrischen Verschlüsselungsverfahren:

- Schlüsselaustausch über öffentliche Kanäle (**öffentliche Schlüssel**)
- Verschlüsselung ohne vorherigen Schlüsselaustausch (mit **geheimen Schlüsseln**, die nicht versendet werden)
- Digitale Signierung und Authentifizierung

Dies können **asymmetrische Verfahren** leisten (auch **Public Key-Kryptographie** genannt) und gehen zurück auf Ideen von Diffie<sup>1</sup> und Hellman<sup>2</sup> aus den 70er Jahren:

Jeder Nutzer eines Kommunikationskanals hat einen privaten Schlüssel, den er geheim hält und niemand sonst kennt, sowie einen öffentlichen Schlüssel, den jeder einsehen kann. Eine Nachricht wird dann unter Ausnutzung einer Funktion  $x \mapsto f(x)$  verschlüsselt, die zwar leicht zu berechnen, aber praktisch nur mit Kenntnis des privaten Schlüssels des rechtmäßigen Empfängers entschlüsselt werden kann. Der Sender der Nachricht wird dafür den

---

<sup>1</sup>Whitfield Diffie, [http://de.wikipedia.org/wiki/Whitfield\\_Diffie](http://de.wikipedia.org/wiki/Whitfield_Diffie)

<sup>2</sup>Martin Hellman, [http://de.wikipedia.org/wiki/Martin\\_Hellman](http://de.wikipedia.org/wiki/Martin_Hellman)

öffentlichen Schlüssel des Empfängers zur Verschlüsselung benutzen. Eine derartige Funktion heißt **Einwegfunktion**.

### Beispiele

- **RSA-Verfahren:**  $(p, q) \mapsto p \cdot q$  mit  $p, q$  prim.
- **ECC-Verfahren:**  $x \mapsto mx$  in einer Gruppe auf einer elliptischen Kurve.

In einem ersten Teil der Vorlesung stellen wir gängige Verfahren dar, die leicht mit dem Zahlring  $\mathbb{Z}$  und Strukturen darin realisiert werden können. Dabei werden wir nur einige Hilfsmittel der elementaren Zahlentheorie entwickeln und dafür heranziehen. In einem zweiten Teil studieren wir die Eigenschaften elliptischer Kurven als interessante geometrische und arithmetische Objekte, die sich in der Praxis der Kryptographie als nützlich erwiesen haben. Wir besprechen dann auch die Sicherheit und Implementierung dieser Verfahren und vergleichen sie miteinander.

### Elliptische Kurven

Was sind elliptische Kurven? Jedenfalls sind elliptische Kurven **keine** Ellipsen. Ellipsen lassen sich durch Gleichungen der Form

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1 \text{ mit } a, b \in \mathbb{R} \setminus \{0\}$$

beschreiben. Durch die Parametrisierung  $x(t) = a \cdot \cos(t), y(t) = b \cdot \sin(t)$  ergibt sich für die Bogenlänge der Ellipse ein elliptisches Integral zweiter Art, nämlich

$$\int_0^{2\pi} \sqrt{\left(\frac{dx(t)}{dt}\right)^2 + \left(\frac{dy(t)}{dt}\right)^2} dt = 4 \int_0^{2\pi} \sqrt{a^2 \cos^2(t) + b^2 \cdot \sin^2(t)} dt$$

Im Allgemeinen lässt sich dies nicht elementar integrieren (außer natürlich, falls  $a = b$ , d.h. ein Kreis vorliegt). Mit Hilfe von elliptischen Kurven findet man jedoch nicht-elementare Stammfunktionen für diese Integrale ( $\Rightarrow$  Funktionentheorie). Aufgrund dieses Zusammenhangs haben elliptische Kurven ihren Namen, sie haben ansonsten nichts mit Ellipsen zutun.

Was sind nun elliptische Kurven? Es sind "abelsche Varietäten der Dimension 1". Elliptische Kurven sind spezielle algebraische Kurven über einem Körper  $k$ . Es handelt sich dabei um glatte kubische Kurven, deren definierende algebraische Gleichung sich meist in die Form

$$E: y^2 = x^3 + ax + b \text{ mit } a, b \in k$$

bringen lässt. Als Punktmenge haben wir dafür

$$E(k) := \{(x, y) \in k^2 : y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\},$$

die Kurve hängt nur von  $a, b$  ab. Die Rolle des zusätzlichen so genannten "unendlich fernen Punkts"  $\mathcal{O}$  werden wir dabei noch näher beleuchten.

Zwei typische Beispiele für elliptische Kurven:

- 1)  $E_1: y^2 = x^3 + 17$ , hier liegen sogar Punkte mit ganzzahligen Koordinaten auf  $E_1$ , nämlich  $(-2, 3), (-1, 4), (2, 5)$ . Die Kurve besteht aus einer Zusammenhangskomponente.
- 2)  $E_2: y^2 = x^3 + ax + b$ , wenn  $f(x) = x^3 + ax + b$  drei verschiedene Nullstellen hat, z.B.  $a = -3, b = -1$ . Die Kurve besteht dann aus zwei Zusammenhangskomponenten.

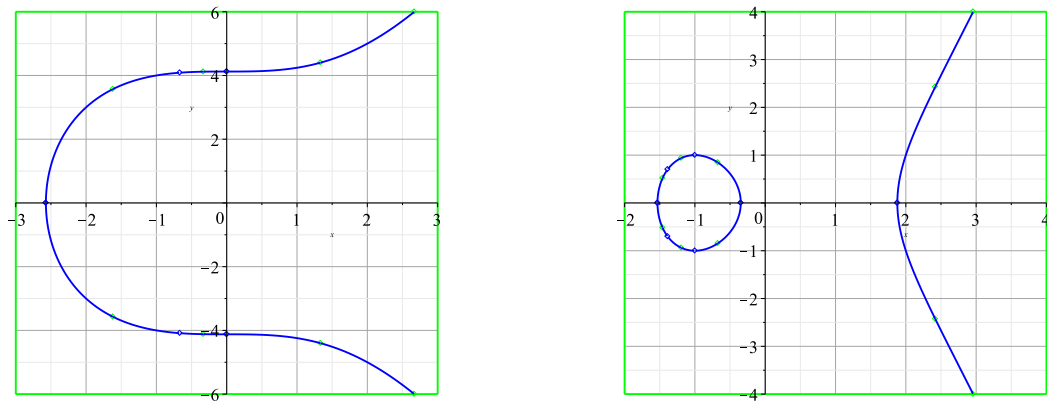


Abbildung 1: Die Kurven  $E_1$  (links) und  $E_2$  (rechts).

### Bemerkung

Die kubischen Kurven  $C_1: y^2 = x^3 - 3x + 2$  und  $C_2: y^2 = x^3$  z. B. sind jedoch keine elliptischen Kurven, weil diese nicht glatt sind.

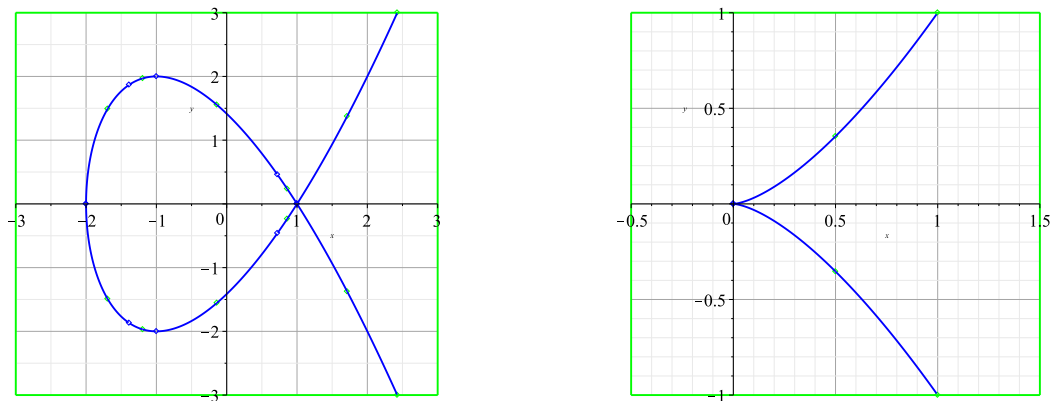


Abbildung 2: Die Kurven  $C_1$  (links) und  $C_2$  (rechts).  $C_1$  ist nicht glatt im Punkt  $(1, 1)$ ,  $C_2$  nicht im Punkt  $(0, 0)$ .

Für die Kryptographie sind elliptische Kurven interessant, weil sich eine Verknüpfung auf ihrer Punktmenge definieren lässt, mit der diese zu einer Gruppe wird. Dabei gerade auch endliche Körper  $k$  zuzulassen, macht diese Verknüpfung auf Rechnemaschinen realisierbar. Die Sicherheit der darauf beruhenden elliptic curve cryptography (ECC) beruht darauf, dass das Problem des diskreten Logarithmus auf einer elliptischen Kurve  $E$ , nämlich die Umkehrung der Funktion  $P \mapsto mP$  für  $m \in \mathbb{N}$  fest, nach heutigem Wissensstand rechnerisch im Allgemeinen extrem schwer realisierbar ist.

# 1 Allgemeines über Kryptographieverfahren

## 1.1 Grundlagen aus der elementaren Zahlentheorie und Gruppentheorie

### 1.1.1 Zahlen, Darstellung von Zahlen

Die Zahlbereiche  $\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R} \subseteq \mathbb{C}$  sind aus den Grundvorlesungen bekannt. Bezüglich den Verknüpfungen  $+$  und  $\cdot$  sind verschiedene Axiome erfüllt, die diese Zahlbereiche zu interessante algebraische Strukturen machen: [2]

Halbgruppe	Gruppe	Ring	Körper
$(\mathbb{N}, +), (\mathbb{N}, \cdot)$			
$(\mathbb{Z}, +), (\mathbb{Z}, \cdot)$	$(\mathbb{Z}, +, 0)$	$(\mathbb{Z}, +, \cdot)$	
$(\mathbb{Q}, +), (\mathbb{Q}, \cdot)$	$(\mathbb{Q}, +, 0), (\mathbb{Q} \setminus \{0\}, \cdot, 1)$	$(\mathbb{Q}, +, \cdot)$	$(\mathbb{Q}, +, \cdot)$
$(\mathbb{R}, +), (\mathbb{R}, \cdot)$	$(\mathbb{R}, +, 0), (\mathbb{R} \setminus \{0\}, \cdot, 1)$	$(\mathbb{R}, +, \cdot)$	$(\mathbb{R}, +, \cdot)$
$(\mathbb{C}, +), (\mathbb{C}, \cdot)$	$(\mathbb{C}, +, 0), (\mathbb{C} \setminus \{0\}, \cdot, 1)$	$(\mathbb{C}, +, \cdot)$	$(\mathbb{C}, +, \cdot)$

Weiter sind  $\mathbb{Q}$  und  $\mathbb{R}$  angeordnete Körper, d.h. es gibt eine Anordnungsrelation  $\leq$ , die sich mit  $+$  und  $\cdot$  verträgt. Für  $\mathbb{C}$  ist eine solche Anordnung nicht mehr möglich.

#### Definition 2.1 (Halbgruppe)

Eine Menge  $H \neq \emptyset$  mit Verknüpfung  $*$ :  $H \times H \rightarrow H$  heißt **Halbgruppe**, falls  $*$  assoziativ ist, d.h. für alle  $a, b, c \in H$  gilt  $a * (b * c) = (a * b) * c$ .

#### Definition 2.2 (Gruppe)

Eine Halbgruppe  $(G, *)$  heißt **Gruppe**, falls es ein neutrales Element  $e \in G$  gibt mit  $e * g = g * e = g$  für alle  $g \in G$ , und falls zu jedem  $g \in G$  ein inverses Element  $h \in G$  existiert mit  $h * g = g * h = e$ . Wir schreiben auch  $g^{-1}$ ,  $\frac{1}{g}$  oder  $-g$  für  $h$ .

#### Definition 2.3 (abelsche Gruppe)

Eine Gruppe  $(G, *, e)$  heißt **abelsch** bzw. **kommutativ**, falls für alle  $a, b \in G$  gilt:  $a * b = b * a$ .

#### Definition 2.4 (Ring)

Ein **Ring**  $(R, +, \cdot)$  ist eine Menge  $R \neq \emptyset$  und zwei Verknüpfungen  $+$  und  $\cdot$  so, dass  $(R, +, 0)$  eine Gruppe ist,  $(R, \cdot, 1)$  eine Halbgruppe mit neutralem Element 1, und so, dass die Distributivgesetze gelten, d.h.  $(a + b) \cdot c = a \cdot c + b \cdot c$  und  $c \cdot (a + b) = c \cdot a + c \cdot b$ .

Ring mit Eins

#### Bemerkung 2.5

Die Addition  $+$  ist in einem Ring stets kommutativ. Ein Ring heißt kommutativ, wenn die Multiplikation  $\cdot$  kommutativ ist. Soll der Nullring  $R = \{0\}$  mit  $1 = 0$  ausgeschlossen werden, fordert man zusätzlich noch  $1 \neq 0$  in den Ringaxiomen.

#### Definition 2.6 (Einheit, Einheitengruppe)

Die in einem Ring  $(R, +, \cdot)$  bezüglich  $\cdot$  invertierbaren Elemente heißen **Einheiten**. Die Menge der Einheiten in  $R$  wird mit  $R^*$  bezeichnet, d.h. also  $R^* := \{a \in R : \exists b \in R \text{ mit } a \cdot b = b \cdot a = 1\}$ . Damit ist  $(R^*, \cdot, 1)$  also eine Gruppe.

#### Definition 2.7 (Körper)

Ein **Körper**  $(K, +, \cdot)$  ist ein kommutativer Ring mit  $1 \neq 0$ , für den  $K^* = K \setminus \{0\}$  gilt.

Algebraische Strukturen dieser Art können wir auch in Teilmengen von  $\mathbb{Z}$  auffinden und diese für kryptographische Anwendungen ausnutzen. Darum geht es in §1 dieser Vorlesung. Dabei wird klar, dass die Anwendungen auch – teilweise – in beliebigen Gruppen, Ringen und Körpern möglich sind. Die Gruppen, die durch elliptische Kurven gegeben sind, haben sich in der Praxis dann als vorteilhaft herausgestellt.

Wenn wir Teilmengen von  $\mathbb{Z}$  auch praktisch untersuchen möchten, wird die Frage wichtig, wie man ganze Zahlen auf geschickte und kompakte Art darstellen kann. Dafür benutzen wir im Alltag das Dezimalsystem, für Rechenmaschinen ist auch das Binär- und das Hexadezimalsystem nützlich. Dabei werden die Ziffern  $0, 1, \dots, 9$  bzw.  $0, 1$  bzw.  $0, 1, \dots, 9, A, \dots, F$  verwendet. Allgemein erhalten wir die  $g$ -adische Darstellung von  $n \in \mathbb{N}$  so:

### Satz 2.8

Sei  $g \in \mathbb{N}, g \geq 2$  und  $n \in \mathbb{N}$ . Dann gibt es ein  $k \in \mathbb{N}_0$  und  $c_k, c_{k-1}, \dots, c_0 \in \{0, \dots, g-1\}$  (genannt "Ziffern"), sodass  $n = c_k g^k + c_{k-1} g^{k-1} + \dots + c_0 = \sum_{i=0}^k c_i g^i$ . Fordern wir  $c_k \neq 0$ , ist  $k$  und die Folge  $c_k, \dots, c_1, c_0$  eindeutig bestimmt.

### Beweis

**Existenz:** Sei  $k \in \mathbb{N}_0$  so, dass  $g^k \leq n < g^{k+1}$  gilt, das heißt wir setzen  $k := \left\lfloor \frac{\log(n)}{\log(g)} \right\rfloor$ . Zeige durch Induktion nach  $k$  die Existenz:

$k = 0$ : Setze  $c_0 := n$ .

$k \rightsquigarrow k+1$ : Sei  $g^{k+1} \leq n < g^{k+2}$ . Setze  $n' = n - \left\lfloor \frac{n}{g^{k+1}} \right\rfloor \cdot g^{k+1}$ . Es folgt  $0 \leq n' < g^{k+1}$ , d.h. auf  $n'$  ist die Induktionsvoraussetzung anwendbar. Nach dieser hat  $n'$  eine  $g$ -adische Zifferndarstellung  $n' = \sum_{i=0}^k c_i g^i$ .

Wegen  $1 \leq \frac{n}{g^{k+1}} < g$  ist  $1 \leq \left\lfloor \frac{n}{g^{k+1}} \right\rfloor < g$ , also setze  $c_{k+1} := \left\lfloor \frac{n}{g^{k+1}} \right\rfloor$ .

$$\Rightarrow n = c_{k+1} g^{k+1} + n' = \sum_{i=0}^{k+1} c_i g^i.$$

**Eindeutigkeit:** Sind  $\sum_{i=0}^k a_i g^i = m = \sum_{i=0}^r b_i g^i$  zwei verschiedene Darstellungen von  $m \in \mathbb{N}$ . Ist  $r > k$ , so sei  $a_{k+1} = \dots = a_r := 0$ , sonst sei  $b_{r+1} = \dots = b_k := 0$ , falls  $r < k$ . Dann sei  $l := \max\{i \in \mathbb{N}_0 : i \leq \max\{k, r\}, a_i \neq b_i\}$  die größte Stelle, an der sich die Darstellungen unterscheiden.

$$\Rightarrow 0 = \sum_{i=0}^l \underbrace{(a_i - b_i)}_{=0 \text{ für } i > l} g^i \Rightarrow \underbrace{|b_l - a_l|}_{\geq 1} g^l = \left| \sum_{i=0}^{l-1} (a_i - b_i) g^i \right|$$

$$\Rightarrow g^l \leq \sum_{i=0}^{l-1} |a_i - b_i| g^i \leq \sum_{i=0}^{l-1} (g-1) g^i = (g-1) \frac{g^l - 1}{g-1} = g^l - 1 \quad \not\leq$$

□

### Definition 2.9 ( $g$ -adische Darstellung)

Die Ziffernfolge  $c_k, c_{k-1}, \dots, c_0$  aus Satz 2.8 heißt  $g$ -**adische Darstellung** von  $n$ . Die Zahl  $c_k$  heißt **Leitziffer**, die Zahl  $c_0$  die **Endziffer**. Die Zahl  $k+1$  heißt **Stellenzahl** bzw. **Länge** der  $g$ -adischen Darstellung. Die Zahl  $g$  heißt auch **Basis** der Darstellung. Eine  $m$ -**Bit-Zahl** ist eine Zahl  $n \in \mathbb{N}$  der Länge  $\leq m$  zur Basis 2.

### Bemerkung 2.10

Wir können jede natürliche (und dann auch jede ganze) Zahl  $n$  also eindeutig schreiben als Linearkombination endlich vieler Potenzen von  $g$ .



**Beispiel 2.11**

$$\begin{aligned}
163_{(10)} &= 1 \cdot 10^2 + 6 \cdot 10^1 + 3 \cdot 10^0 \\
43_{(10)} &= 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 101011_{(2)} \\
&= 2 \cdot 16^1 + 11 \cdot 16^0 = 2B_{(16)}
\end{aligned}$$

Die bekannten schriftlichen Additions- und Multiplikationsrechnungen, die unter Beachtung von Überträgen ziffernweise geschehen, können in jeder Basis ausgeführt werden. Es gibt weiter für die Multiplikation großer Zahlen (d.h. mit großer Stellenzahl bis etwa  $2 \cdot 10^{10}$ ) schnelle Algorithmen, die wir hier aber nicht näher behandeln möchten; etwa mit der schnellen Fouriertransformation (FFT) nach Schönhage/Strassen<sup>3</sup>.

Der Beweis von Satz 2.8 zeigt, dass die Länge von  $n$  gleich  $\left\lfloor \frac{\log(n)}{\log(g)} \right\rfloor + 1$  ist, so viele Ziffern müssen zum Hinschreiben bzw. Eintippen von  $n$  angegeben werden. Bei verschiedenen Basen ändert sich hier nur der Faktor  $\frac{1}{\log(g)}$ . Deswegen sagt man, die Länge sei  $\mathcal{O}(\log(n))$  und meint damit die Aussage: Es existiert eine Konstante  $C > 0$ , sodass  $k + 1 \leq C \cdot \log(n)$ . (Landau-Symbolik<sup>4</sup>, "Groß-O-Notation")

Entscheidend für das Studium von  $\mathbb{Z}$  ist der Grundbegriff der Teilbarkeit.

**Definition 2.12 (Teilbarkeit)**

Für  $a, b \in \mathbb{Z}$  heißt  $a$  **Teiler** von  $b$  bzw.  $a$  **teilt**  $b$ , in Zeichen  $a \mid b$ , falls ein  $c \in \mathbb{Z}$  existiert mit  $ac = b$ . Ist  $a$  kein Teiler von  $b$ , schreibt man  $a \nmid b$ .

**Beispiel 2.13**

$3 \mid 12, 4 \mid 0, 0 \mid 0, 7 \nmid 12, 0 \nmid 4$ . Es kann 0 nur die 0 teilen.

**Definition 2.14 (Primzahl)**

Eine natürliche Zahl  $p \in \mathbb{N}$  heißt **Primzahl** bzw. **prim**, wenn sie genau zwei Teiler in  $\mathbb{N}$  besitzt (nämlich 1 und  $p$ ,  $1 \neq p$ ). Eine natürliche Zahl  $n > 1$  heißt **zusammengesetzt**, falls  $n$  keine Primzahl ist.

Primzahlen sind die "Bausteine" der natürlichen Zahlen:

**Satz 2.15 (Satz von der eindeutigen Primfaktorzerlegung, Hauptsatz der Arithmetik)**

Jede natürliche Zahl  $n > 1$  besitzt genau eine Darstellung

$$n = p_1^{e_1} \cdot p_r^{e_r} = \prod_{i=1}^r p_i^{e_i}$$

mit  $r \in \mathbb{N}$ , Primzahlen  $p_1, \dots, p_r$  mit  $e_1, \dots, e_r \in \mathbb{N}$  und  $p_1 < p_2 < \dots < p_r$ . Diese heißt die **Primfaktorzerlegung** (PFZ) von  $n$ .

**Bemerkung 2.16**

Lässt man die letzte Bedingung weg, ist die Darstellung eindeutig bis auf die Reihenfolge der Primpotenzen. Die Zahl  $e_i$  ist dabei die Vielfachheit (auch **Exponent** genannt), mit der  $p_i$  als Faktor in  $n$  auftritt, d.h.  $p_i^{e_i} \mid n$ , aber  $p_i^{e_i+1} \nmid n$ . Dafür gibt es das Symbol  $p^e \parallel n$ , und die Primfaktorzerlegung lässt sich kompakt auch schreiben als  $n = \prod_p p^{e(p)}$ , wobei  $e(p) := e$  mit  $p^e \parallel n$ , falls  $p \mid n$ , und  $e(p) := 0$ , falls  $p \nmid n$ . Weiter ist  $\omega(n) := r$  die Anzahl der verschiedenen Primteiler von  $n$ .

<sup>3</sup>siehe <http://de.wikipedia.org/wiki/Sch%C3%B6nhage-Strassen-Algorithmus>

<sup>4</sup>siehe <http://de.wikipedia.org/wiki/Landau-Symbole>

**Beweis**

**Existenz:** Ist  $n$  prim, ist nichts zu zeigen, und ist  $n$  nicht prim, gibt es  $k, l \in \mathbb{N} \setminus \{1\}$  mit  $n = kl$ . Da  $\min\{k, l\} > 1$ , folgt  $\max\{k, l\} < n$ . Nach Induktionsvoraussetzung sind also  $k, l$  Produkte von Potenzen von Primzahlen, also auch  $n = kl$ .

**Eindeutigkeit:** Sei  $n > 1$  minimal mit zwei verschiedenen Zerlegungen  $n = \prod_{i=1}^r p_i^{e_i} = \prod_{i=1}^s q_i^{f_i}$ , die  $p_i, q_i$  prim und angeordnet. Da  $p_1 \neq q_i$  für alle  $i$  gilt (sonst hätte  $\frac{n}{p_1} < n$  zwei verschiedene Zerlegungen), ist  $\text{ggT}(p_1, q_i) = 1$ , und mit den Zerlegungen folgt  $p_1 \mid q_1^{f_1-1}$  aus Lemma 2.21. Die Fortsetzung des Verfahrens zeigt schließlich  $p_1 \mid q_s$ , was wegen  $\text{ggT}(p_1, q_s) = 1$  ein Widerspruch ist.  $\square$  (Beachte: Zum Beweis von Lemma 2.21 wurde nie die Eindeutigkeit der Primfaktorzerlegung benutzt.)

Die Eindeutigkeit der Primfaktorzerlegung zeigt, dass auch diese eine Möglichkeit zur Darstellung natürlicher Zahlen ist. Diese ist jedoch unpraktisch, weil das folgende Problem im Allgemeinen schwer zu lösen ist, worauf einige kryptographische Verfahren (insb. RSA) beruhen.

**Definition 2.17 (Faktorisierungsproblem)**

Zu einer natürlichen zusammengesetzten Zahl  $n > 1$  bestimme man einen nichttrivialen Teiler  $t$  mit  $1 < t < n$ .

Klar: Ist das Faktorisierungsproblem rechnerisch leicht zu machen, kann auch (durch Iteration) die Primfaktorzerlegung von  $n$  leicht bestimmt werden. In der Praxis, wenn  $n$  nicht gerade schon von einer speziellen Form ist, können Teiler großer Zahlen  $n$  jedoch nur sehr schwer aufgefunden werden.

- Das derzeit schnellste algorithmische Verfahren zur Faktorisierung (auf einem klassischen Computer) ist das **Zahlkörpersieb** mit einer Laufzeit von nur  $\mathcal{O}(\exp(C(\log n)^{1/3}(\log \log n)^{2/3}))$ , d.h. es handelt sich um so genanntes **subexponential schnelles Verfahren**, weil  $(\log n)^B \ll \exp(C(\log n)^{1/3}(\log \log n)^{2/3}) \ll \exp(D \log n) = n^D$ .
- Peter Shor<sup>5</sup> entdeckte um 1994, dass das Faktorisierungsproblem auf einem Quantencomputer mit einer Laufzeit von (meist) nur  $\mathcal{O}((\log n)^3)$  sehr (d.h. polynomiell) schnell gelöst werden kann, was die Sicherheit gängiger Kryptoverfahren wie RSA untergräbt. Allerdings ist die Konstruktion solcher Quantencomputer (physikalisch) extrem schwierig, diverse Forschergruppen arbeiten daran. Am 2.1.2014 meldete die Washington Post unter Berufung auf Dokumente von Edward Snowden<sup>6</sup>, dass die NSA an der Entwicklung eines kryptographisch nützlichen Quantencomputers arbeitet<sup>7</sup>. Zum Begriff Quantencomputer siehe [Wikipedia](#).

Im Folgenden besprechen wir noch den ggT zweier natürlicher Zahlen, der sich in vielerlei Hinsicht als wichtig und nützlich erweist:

**Definition 2.18**

Seien  $a, b \in \mathbb{Z}$ . Der **größte gemeinsame Teiler** (ggT) von  $a$  und  $b$  in  $\mathbb{N}$  ist die Zahl  $d := \max\{t \in \mathbb{N} : t \mid a \wedge t \mid b\}$ .

Notation:  $\text{ggT}(a, b) := d$ . Ist  $\text{ggT}(a, b) = 1$ , heißen  $a$  und  $b$  **teilerfremd**.

Haben wir für  $a$  und  $b$  die Primfaktorzerlegungen  $a = \prod_p p^{e(p)}$  und  $b = \prod_p p^{f(p)}$  vorliegen, kann ihr ggT leicht bestimmt werden als  $\text{ggT}(a, b) = \prod_p p^{\min(e(p), f(p))}$ , z.B.  $\text{ggT}(2^3 \cdot 3^6 \cdot 5^4, 2^4 \cdot 3^5) = 2^3 \cdot 3^5$ . Wegen des Faktorisierungsproblems kann dies aber so nicht praktisch umgesetzt werden. Stattdessen benutzt man den (polynomiell) schnellen euklidischen Algorithmus, vgl. Übungsaufgabe.

<sup>5</sup>[http://de.wikipedia.org/wiki/Peter\\_Shor](http://de.wikipedia.org/wiki/Peter_Shor)

<sup>6</sup>[http://de.wikipedia.org/wiki/Edward\\_Snowden](http://de.wikipedia.org/wiki/Edward_Snowden)

<sup>7</sup>Link zum Artikel

**Satz 2.19 (Teilen mit Rest)**

Zu  $a \in \mathbb{Z}, b \in \mathbb{N}$  existieren eindeutigen  $q, r \in \mathbb{Z}, 0 \leq r < b$  mit  $a = qb + r$ , nämlich  $q = \lfloor \frac{a}{b} \rfloor = \max\{m \in \mathbb{Z} : m \leq \frac{a}{b}\}$  und  $r = a - qb$ . Dabei heißt  $r$  der **kleinste nichtnegative Rest**. Statt  $0 \leq r < b$  kann auch  $r \in \mathbb{Z}, |r| < \frac{b}{2}$ , erfüllt werden;  $r$  heißt dann der **absolut kleinste Rest** (bei Division durch  $b$ ).

**Satz 2.20 (Euklidischer Algorithmus)**

Seien  $a, b \in \mathbb{N}$ . Durch fortgesetztes Teilen mit Rest erhalten wir als letzten Rest  $\neq 0$  den  $\text{ggT}(a, b)$ , sowie  $x, y \in \mathbb{Z}$  mit  $\text{ggT}(a, b) = xa + yb$  (siehe Schema).

**Beschreibung des Rechenverfahrens**

Rechnen sukzessive mit  $r_{-1} := a, r_0 := b$ :

$$r_{-1} = q_0 r_0 + r_1$$

$$r_0 = q_1 r_1 + r_2$$

$$r_1 = q_2 r_2 + r_3$$

$$\vdots$$

Das Verfahren wird fortgeführt, bis erstmals ein Rest  $r_{m+1} = 0$  auftritt, was wegen  $r_0 > r_1 > r_2 > \dots$  nach höchstens  $b + 1$  vielen Schritten der Fall sein wird. Sind die Quotienten  $q_0, \dots, q_m$  bekannt, können mit den Rekursionen

$$c_{-2} = 0, c_{-1} = 1 \text{ und } c_k = q_k c_{k-1} + c_{k-2}, k = 0, 1, 2, \dots, n$$

$$d_{-2} = 1, d_{-1} = 0 \text{ und } d_k = q_k d_{k-1} + d_{k-2}, k = 0, 1, 2, \dots, n$$

die **Bézout-Elemente** als  $x = (-1)^{n-1}, y = (-1)^n c_{n-1}$  berechnet werden.

Wir behaupten also:

(1) Es ist  $\text{ggT}(a, b) = r_n$ .

(2)  $\text{ggT}(a, b) = \underbrace{(-1)^{n-1} d_{n-1}}_x a + \underbrace{(-1)^n c_{n-1}}_y b$

**Beweis**

**zu (1)** : Da  $r_n \mid r_{n-1}, r_n \mid r_{n-2}, \dots, r_n \mid r_0 = b, r_n \mid r_{-1} = a$ , ist  $r_n$  ein Teiler von  $a$  und  $b$  (Teilen mit Rest von unten nach oben). Ist  $d$  irgendein Teiler  $\geq 1$  von  $a$  und  $b$ , folgt  $d \mid r_1 = a - q_0 b \Rightarrow d \mid r_2 = r_0 - q_1 r_1 \Rightarrow d \mid r_3 = \dots$ , also auch  $r_n$ , sodass  $d \leq r_n$  folgt (Teilen mit Rest von oben nach unten). Somit ist  $r_n = \text{ggT}(a, b)$ .

**zu (2)** : Induktiv kann  $c_{k-1} d_k - c_k d_{k-1} = (-1)^k$  gezeigt werden. Daher genügt zu zeigen:  $c_n = \frac{a}{\text{ggT}(a, b)}, d_n = \frac{b}{\text{ggT}(a, b)}$ .

Mit den  $\frac{c_k}{d_k}$  wird die Kettenbruchentwicklung von  $\frac{a}{b}$  berechnet und diese bricht bei  $\frac{c_n}{d_n} = \frac{a}{b}$  ab. Da bei der Kettenbruchentwicklung alle Brüche  $\frac{c_k}{d_k}$  gekürzt sind wegen  $c_{k-1} d_k - c_k d_{k-1} = (-1)^k$ , folgt dies.

Details siehe  
EZT-Skript Lorenz

Der Satz vom Euklidischen Algorithmus sichert uns konstruktiv also die Existenz ganzer Zahlen  $x, y \in \mathbb{Z}$  mit  $\text{ggT}(a, b) = xa + yb$ . Die Zahlen  $x$  und  $y$  heißen auch **Bézout-Elemente** von  $a$  und  $b$ . Deren Existenz ist auch in der Theorie immer wieder wichtig, z.B. hierfür:

**Lemma 2.21**

Seien  $a, b, c \in \mathbb{Z}$  und  $b, c \neq 0$ . Gilt  $c \mid ab$  und  $\text{ggT}(b, c) = 1$ , dann ist  $c \mid a$ .

**Beweis**

Aus den Voraussetzungen und  $c \mid ac$  folgt, dass  $c \mid \text{ggT}(ab, ac) = |a| \cdot \text{ggT}(b, c) = |a|$ , also  $c \mid a$ . Zur ersten Gleichheit: Nach Satz 2.20 existieren  $x, y \in \mathbb{Z}$  mit  $\text{ggT}(b, c) = xb + yc$ .

$|a| \cdot \text{ggT}(b, c)$  teilt  $|a| \cdot b$  und  $|a| \cdot c$ , also auch  $ba$  und  $ca$ , d.h. die rechte Seite ist ein gemeinsamer Teiler von  $ba$  und  $ca$ . Ist  $t$  irgendein solcher, so teilt  $t$  auch  $\text{sgn}(a) \cdot (xba + yca) = xb \cdot |a| + yc \cdot |a| = |a| \cdot (xb + yc) = |a| \text{ggT}(b, c)$ .  
□

### 1.1.2 Kongruenzenrechnen und die modulare Brille

[3] Wir behandeln nun, wie man mit Teilmengen von  $\mathbb{Z}$  und neuen Definitionen von „+“ und „·“ zu neuen algebraischen Strukturen (Gruppe, Ringe, Körper) kommt. Dazu ist das Kongruenzenrechnen modulo  $m$  wesentlich.

#### Definition 3.1 (Kongruenz, Modul)

Sei  $m \in \mathbb{N}$ . Dann heißen  $a \in \mathbb{Z}$  und  $b \in \mathbb{Z}$  **kongruent modulo**  $m$ , wenn  $m \mid (b - a)$ . Wir schreiben dann  $a \equiv b \pmod{m}$  oder  $a \equiv b (m)$ . Die Zahl  $m$  heißt der **Modul** der Kongruenz.

#### Folgerung 3.2

- (1)  $a \equiv b \pmod{m}$  bedeutet, dass  $a$  und  $b$  bei Division durch  $m$  denselben kleinsten nichtnegativen (absolut kleinsten) Rest lassen.
- (2)  $a \equiv b \pmod{m}, b \equiv c \pmod{m} \Rightarrow a \equiv c \pmod{m}$
- (3)  $a_1 \equiv b_1 \pmod{m}, a_2 \equiv b_2 \pmod{m} \Rightarrow a_1 + a_2 \equiv b_1 + b_2 \pmod{m}, a_1 \cdot a_2 \equiv b_1 \cdot b_2 \pmod{m}$ .
- (4)  $ca \equiv cb \pmod{m} \Rightarrow a \equiv b \pmod{\left(\frac{m}{\text{ggT}(c, m)}\right)}$ , insbesondere  $a \equiv b \pmod{m}$ , falls  $\text{ggT}(c, m) = 1$ .
- (5)  $a \equiv b \pmod{m_i}$  für  $i = 1, \dots, k \Rightarrow a \equiv b \pmod{\text{kgV}(m_1, \dots, m_k)}$

Dies zeigt, dass  $\equiv$  für festes  $m$  eine Äquivalenzrelation ist und  $\mathbb{Z}$  in  $m$  paarweise disjunkte Äquivalenzklassen zerlegt.

#### Definition 3.3 (Restklasse)

Die Äquivalenzklassen von  $\equiv$  modulo  $m$  heißen **Restklassen** modulo  $m$ . (auch: Kongruenzklassen modulo  $m$ ).

#### Folgerung 3.4

Die Restklassen modulo  $m$  sind Teilmengen von  $\mathbb{Z}$  der Gestalt  $x + m\mathbb{Z} := \{x + ma : a \in \mathbb{Z}\}$ . Die Restklasse  $x + m\mathbb{Z}$  heißt auch die Restklasse von  $x$  modulo  $m$ . Davon gibt es  $m$  Stück; wird in jeder Restklasse ein Element  $x_i, i = 1, \dots, m$  ausgewählt, können die  $m$  Restklassen mit  $x_1 + m\mathbb{Z}, x_2 + m\mathbb{Z}, \dots, x_m + m\mathbb{Z}$  angegeben werden; die Menge  $\{x_1, \dots, x_m\}$  heißt dann **vollständiges Restsystem** modulo  $m$ . Sind  $y_1, \dots, y_m \in \mathbb{Z}$  so, dass  $y_i \not\equiv y_j \pmod{m}$  für alle  $i \neq j, 1 \leq i, j \leq m$ , gilt (d.h. die  $y_i$  sind paarweise inkongruent modulo  $m$ ), dann ist  $\{y_1, \dots, y_m\}$  ein vollständiges Restsystem modulo  $m$ . Die Zahl  $x$  heißt **Repräsentant** der Restklasse  $x + m\mathbb{Z}$ , und  $x + m\mathbb{Z} = z + m\mathbb{Z} \Leftrightarrow x \equiv z \pmod{m}$ , weil laut Definition in der Restklasse von  $x \pmod{m}$  genau alle zu  $x$  kongruenten Zahlen liegen.

#### Beispiel 3.5

$\{0, 1, 2\}$  ist vollständiges Restsystem modulo 3, und vollständige Restsysteme modulo 8 sind etwa  $\{1, \dots, 8\}$  und  $\{3, 6, 9, 12, 15, 18, 21, 24\} = \{3a : 1 \leq a \leq 8\}$ , da  $12 \equiv 4 \pmod{8}, 15 \equiv 7 \pmod{8}, 18 \equiv 2 \pmod{8}, 21 \equiv 5 \pmod{8}, 24 \equiv 0 \pmod{8}$ . Die Menge  $\{2a : 1 \leq a \leq 8\}$  ist kein vollständiges Restsystem modulo 8. Die Reste  $0, 1, \dots, m - 1$  könnte man auch als „Standardrepräsentanten“ modulo  $m$  bezeichnen, da sie immer ein vollständiges Restsystem modulo  $m$  bilden.

#### Folgerung 3.6

Ist  $\{x_1, \dots, x_m\}$  ein vollständiges Restsystem modulo  $m$  und  $a \in \mathbb{Z}, c \in \mathbb{Z}$  mit  $\text{ggT}(c, m) = 1$ , so sind auch  $\{x_1 + a, x_m + a\}$  und  $\{x_1 \cdot c, \dots, x_m \cdot c\}$  vollständige Restsysteme modulo  $m$  (vgl. (4) aus Folgerung 3.2).

Das nützliche an den Restklassen modulo  $m$  ist, dass wir nun durch folgende naheliegende Definitionen von  $\oplus$  und  $\odot$  mit ihnen neue algebraische Strukturen gewinnen können:

**Definition 3.7 (Addition und Multiplikation auf  $\mathbb{Z}_m$ )**

Ist der Modul  $m$  klar, schreiben wir auch  $\underline{x} := x + m\mathbb{Z}$  für die Restklasse von  $x$  modulo  $m$ . Wir definieren für  $x, y \in \mathbb{Z}$  dann

$$\underline{x} \oplus \underline{y} := \underline{x + y}$$

$$\underline{x} \odot \underline{y} := \underline{x \cdot y}$$

Weiter sei  $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z} = \mathbb{Z}/m := \{x + m\mathbb{Z} : x \in \mathbb{Z}\}$  die Menge der  $m$  vielen Restklassen modulo  $m$ .

**Folgerung 3.8**

Wir addieren bzw. multiplizieren zwei Restklassen, indem wir Repräsentanten  $x, y$  auswählen und diese addieren bzw. multiplizieren. Das ist nur sinnvoll, wenn bei unterschiedlicher Repräsentantenwahl dieselbe Restklasse als Ergebnis herauskommt. Man sagt, die Definition von  $\oplus$  und  $\odot$  ist wohldefiniert, da repräsentantenunabhängig. Dies ist klar:  $\underline{x_1} = \underline{x_2}$  und  $\underline{y_1} = \underline{y_2} \Rightarrow x_1 \equiv x_2 \pmod{m}$  und  $y_1 \equiv y_2 \pmod{m} \Rightarrow x_1 + y_1 \equiv x_2 + y_2 \pmod{m} \Rightarrow \underline{x_1 + y_1} = \underline{x_2 + y_2}$ , also erhalten wir so dieselbe Restklasse für  $\underline{x_1} \oplus \underline{y_1}$  und  $\underline{x_2} \oplus \underline{y_2}$ , wenn  $\underline{x_1} = \underline{x_2}$  und  $\underline{y_1} = \underline{y_2}$  (analog für die Multiplikation). Damit kann  $(\mathbb{Z}_m, \oplus)$  oder  $(\mathbb{Z}/m, \odot)$  auf algebraische Strukturen hin untersucht werden. Wir schreiben ab jetzt auch  $+$  für  $\oplus$  und  $\cdot$  für  $\odot$ .

**Folgerung 3.9**

$(\mathbb{Z}_m, +)$  ist eine abelsche Gruppe mit neutralem Element  $\underline{0} = 0 + m\mathbb{Z}$ , denn Kommutativität und Assoziativität gelten wie in  $\mathbb{Z}$ , und  $\underline{0} + \underline{x} = \underline{0 + x} = \underline{x}$  gilt für alle  $x \in \mathbb{Z}$ , sowie  $\underline{x} + \underline{-x} = \underline{x - x} = \underline{0}$ , sodass  $\underline{-x} = \underline{-x} = \underline{m - x}$  für alle  $x \in \mathbb{Z}$  gilt. Ebenso gilt, dass  $(\mathbb{Z}_m, +, \cdot)$  ein kommutativer Ring mit 1 ist.

Das Beispiel  $\underline{2} \cdot \underline{0} = \underline{0}$ ,  $\underline{2} \cdot \underline{1} = \underline{2}$ ,  $\underline{2} \cdot \underline{2} = \underline{0}$  modulo 4 zeigt, dass es Restklassen ohne Inversen bezüglich  $\cdot$  geben kann. Der folgende Satz gibt an, welche Restklassen invertierbar sind, d.h. im Ring  $\mathbb{Z}_m$  eine Einheit sind:

**Satz 3.10 (Einheiten in  $\mathbb{Z}_m$ )**

Zu  $\underline{x} \in \mathbb{Z}_m$  existiert genau dann ein multiplikatives Inverses, d.h. ein  $\underline{y} \in \mathbb{Z}_m$  mit  $\underline{x} \cdot \underline{y} = \underline{1} \Leftrightarrow x \cdot y \equiv 1 \pmod{m}$ , falls  $\text{ggT}(x, m) = 1$ . Wir schreiben dann  $\underline{x}^{-1}$  oder  $\underline{x}^*$  für  $\underline{y}$ , die Bezeichnungen  $\frac{1}{\underline{x}}$  oder  $1/\underline{x}$  sind didaktisch ungeschickt.

**Beweis**

" $\Rightarrow$ ": Sei  $\underline{y} \in \mathbb{Z}_m$  mit  $\underline{x} \cdot \underline{y} = \underline{1}$ , d.h.  $xy \equiv 1 \pmod{m}$ , also existiert  $k \in \mathbb{Z}$  mit  $1 - xy = km \Rightarrow xy + km = 1$ . Wäre  $d = \text{ggT}(x, m) > 1$ , so folgt  $d \mid xy + km = 1 \nmid$ .

" $\Leftarrow$ ": Sei  $\text{ggT}(x, m) = 1$ . Nach Satz 2.20 existiert  $y, k \in \mathbb{Z}$  mit  $1 = yx + km$ , also folgt  $\underline{x} \cdot \underline{y} = \underline{1}$ . □

Fazit: Mit dem euklidischen Algorithmus können wir also Inverse schnell explizit berechnen.

**Definition 3.11 (Prime Reste, Eulersche  $\varphi$ -Funktion)**

$\underline{x} = x + m\mathbb{Z}$  heißt **prime** oder **reduzierte Restklasse** modulo  $m$ , falls  $\text{ggT}(x, m) = 1$  gilt. Diese sind genau die Einheiten in  $(\mathbb{Z}_m, +, \cdot)$ , d.h.

$$\mathbb{Z}_m^* = \{\underline{x} \in \mathbb{Z}_m : \text{ggT}(x, m) = 1\}$$

Die Anzahl der Einheiten sei  $\varphi(m) := \#\mathbb{Z}_m^* = \#\{a \in \mathbb{N} : a \leq m, \text{ggT}(a, m) = 1\}$ , die so erklärte Funktion  $\varphi: \mathbb{N} \rightarrow \mathbb{N}$  heißt **Eulersche  $\varphi$ -Funktion**. Jedes Repräsentantensystem  $\{x_1, \dots, x_{\varphi(m)}\}$  von  $\mathbb{Z}_m^*$  heißt **reduziertes** oder **primes Restsystem** modulo  $m$ .

**Satz 3.12 (Multiplikativität von  $\varphi$ )**

Es ist  $\varphi(p^k) = p^k - p^{k-1}$  für alle  $p$  prim, alle  $k \in \mathbb{N}$ , und  $\varphi(mn) = \varphi(m) \cdot \varphi(n)$ , falls  $\text{ggT}(m, n) = 1$ .

**Beweis**

Unter den Zahlen  $1, 2, \dots, p^k$  sind genau die Vielfachen von  $p$  zu  $p^k$  nicht teilerfremd, d.h.  $p, 2p, \dots, p^{k-1} \cdot p$ , was  $p^{k-1}$ -viele Zahlen sind. Zur Multiplikativität siehe Zusatz 3.18.

Ist  $n = \prod_{p|n} p^{e(p)}$  die Primfaktorzerlegung von  $n$ , folgt aus Satz 3.12:

$$\varphi(n) = \prod_{p|n} (p^{e(p)} - p^{e(p)-1}) = \prod_{p|n} p^{e(p)} \cdot \left(1 - \frac{1}{p}\right) = n \cdot \prod_{p|n} \left(1 - \frac{1}{p}\right)$$

**Folgerung 3.13**

$(\mathbb{Z}_m^*, \cdot)$  ist eine Gruppe, die multiplikative Gruppe von  $\mathbb{Z}_m$ , und die Gruppe  $(\mathbb{Z}_m, +)$  heißt additive Gruppe von  $\mathbb{Z}_m$ . Im Fall  $\mathbb{Z}_m^* = \mathbb{Z}_m \setminus \{0\}$  ist  $(\mathbb{Z}_m, +, \cdot)$  ein Körper; dies ist genau dann richtig, wenn  $m = p$  Primzahl ist, weil genau dann alle  $1, 2, \dots, m-1$  zu  $m$  teilerfremd sind. Wir bezeichnen für  $p$  prim diesen Körper mit  $p$  Elementen mit  $\mathbb{F}_p$ . Der Körper  $\mathbb{F}_p$  hat die Eigenschaft, dass  $p \cdot a := \sum_{i=1}^p a = 0$  in  $\mathbb{F}_p$  für alle  $a \in \mathbb{F}_p$  gilt. Wir sagen, er hat die Charakteristik  $p$ .

Weitere endliche  
Körper später

**Definition 3.14 (Charakteristik)**

Sei  $k$  ein Körper. Er hat die **Charakteristik** 0, falls für alle  $m \in \mathbb{N}$  gilt:  $m \cdot 1 := \underbrace{1 + \dots + 1}_{m\text{-mal}} \neq 0$ .

Falls es ein  $m \in \mathbb{N}$  mit  $m \cdot 1 = 0$  gibt, so heißt das kleinste solche  $m \in \mathbb{N}$  die Charakteristik von  $k$ . Wir schreiben kurz  $\text{char}(k) = 0$  bzw.  $\text{char}(k) = m$ .

Zum Beispiel ist  $\text{char}(\mathbb{Q}) = \text{char}(\mathbb{R}) = \text{char}(\mathbb{C}) = 0$  und  $\text{char}(\mathbb{F}_p) = p$ .

**Bemerkung**

Die Charakteristik eines Körpers  $k$  ist entweder 0 oder eine Primzahl, denn sonst wäre  $0 = (m \cdot n) \cdot 1 = m \cdot (n \cdot 1) = (m \cdot 1) \cdot (n \cdot 1) \Rightarrow m \cdot 1 = 0$  oder  $n \cdot 1 = 0$ , da  $k^* = k \setminus \{0\}$ . Widerspruch zu  $m \cdot n$  minimal.

Die Struktur der Zahlringe  $(\mathbb{Z}_m, +, \cdot)$  versteht man besser, indem man sie auf "kleinere" Zahlringe zurückführt:

**Satz 3.15 (Chinesischer Restsatz für Zahlringe)**

Sei  $m > 1$  eine natürliche Zahl und  $m = m_1 \cdot m_2 \cdot \dots \cdot m_r$  eine Zerlegung von  $m$  in paarweise teilerfremde Zahlen  $m_i > 1$ . Dann ist die Abbildung

$$\begin{aligned} F: \mathbb{Z}/m\mathbb{Z} &\longrightarrow (\mathbb{Z}/m_1\mathbb{Z}) \times \dots \times (\mathbb{Z}/m_r\mathbb{Z}) \\ x + m\mathbb{Z} &\longmapsto (x + m_1\mathbb{Z}, \dots, x + m_r\mathbb{Z}) \end{aligned}$$

ein Isomorphismus von Ringen.

**Satz 3.16 (Chinesischer Restsatz für simultane Kongruenzen)**

Seien  $m_1, \dots, m_r > 1$  paarweise teilerfremde Zahlen und sei  $a_1, \dots, a_r \in \mathbb{Z}$ . Dann ist das simultane Kongruenzsystem

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_r \pmod{m_r} \end{aligned}$$

in  $x$  lösbar, die Lösungen sind alle kongruent modulo  $m_1 \cdot \dots \cdot m_r$ .

**Bemerkung**

Satz 3.16 folgt aus 3.15 wegen der Bijektivität von  $F$ , denn  $(a_1 + m_1\mathbb{Z}, \dots, a_r + m_r\mathbb{Z})$  hat dann genau ein Urbild  $x + m\mathbb{Z}$ .

**Zusatz 3.17 (zu Satz 3.16)**

Genau alle  $x \equiv x_0 \pmod{m_1 \cdots m_r}$  lösen das oben angegebene System, wobei  $x_0 = a_1 M_1^* M_1 + \dots + a_r M_r^* M_r$  mit  $M_i := \frac{m_1 \cdots m_r}{m_i}$  und  $M_i^* \in \mathbb{Z}$  ein multiplikatives Inverses von  $M_i \pmod{m_i}$  repräsentiert, d.h. es gilt  $M_i^* \cdot M_i \equiv 1 \pmod{m_i}$ , wobei die  $M_i^*$  mit dem euklidischen Algorithmus (schnell) berechnet werden können.

**Zusatz 3.18 (zu Satz 3.15)**

Die Gruppe  $\mathbb{Z}_m^*$  ist isomorph zu  $\mathbb{Z}_{m_1}^* \times \dots \times \mathbb{Z}_{m_r}^*$ , beide Gruppen haben dann gleich viele Elemente, es folgt

$$\varphi(m) = \varphi(m_1) \cdot \varphi(m_2) \cdots \varphi(m_r),$$

d.h. die Multiplikativität von  $\varphi$  ist ein Korollar des chinesischen Restsatzes.

**Beweis von Satz 3.16**

**Existenz:** Ist  $x \equiv x_0 \pmod{m_1 \cdots m_r}$ , wie in Zusatz 3.17 angegeben, so folgt für alle  $1 \leq i \leq r$ :

$$x \equiv x_0 = \underbrace{a_1 M_1^* M_1}_{\equiv 0 \pmod{m_i}} + \dots + \underbrace{a_i M_i^* M_i}_{\equiv a_i \cdot 1 \pmod{m_i}} + \dots + \underbrace{a_r M_r^* M_r}_{\equiv 0 \pmod{m_i}} \equiv a_i \pmod{m_i}$$

**Eindeutigkeit modulo  $m_1 \cdots m_r$ :** Ist  $y \in \mathbb{Z}$  eine weitere Lösung des Kongruenzsystems, so gilt für alle  $j \neq i$ :

$$y \equiv a_j \pmod{m_j}, \text{ also } \underbrace{M_j^* \cdot M_j}_{\equiv 1 \pmod{m_j}} \cdot y \equiv a_j \pmod{m_j} \text{ und } M_i M_i^* a_i \equiv 0 \pmod{m_j} \text{ (Division durch } m_j), \text{ und somit}$$

$$y \equiv a_j \pmod{m_j} \equiv \sum_{j=1}^k M_j M_j^* a_j \pmod{m_j} \equiv x_0 \pmod{m_j} \text{ für alle } j = 1, \dots, r.$$

Damit folgt, dass  $m_j$  Teiler von  $y - x_0$  ist. Da die  $m_1, \dots, m_r$  alle paarweise teilerfremd sind, folgt daraus  $y \equiv x_0 \pmod{m_1 \cdots m_r}$ , vgl. 3.2 (5).  $\square$

**Beispiel zum chinesischen Restsatz**

Das System

$$x \equiv 2 \pmod{7}$$

$$x \equiv 3 \pmod{8}$$

hat die Lösung  $x \equiv 2 \cdot 1 \cdot 8 + 3 \cdot (-1) \cdot 7 = -5 \equiv 51 \pmod{56}$ , denn  $1 \equiv 8^{-1} \pmod{7}$  und  $-1 \equiv 7^{-1} \pmod{8}$ .

Beispiel-Textaufgabe dazu: Gegeben seien zwei Tüten mit gleich vielen Bonbons. Beim gleichmäßigen Aufteilen der einen Tüte an sieben Kinder bleiben zwei Bonbons übrig. Beim Aufteilen der anderen auf acht Kinder bleiben drei Bonbons übrig. Wie viele Bonbons waren in einer Tüte?

Lösung: Möglich sind 51, 107, 163, ... Stück.

**Beispiele zum Rechnen mit Kongruenzen**

- Es ist  $5x \equiv 4 \pmod{12} \Leftrightarrow 5^{-1} \cdot 5x \equiv 4 \cdot 5^{-1} \pmod{12} \Leftrightarrow x \equiv 4 \cdot 5^{-1} \equiv 4 \cdot 5 = 20 \equiv 8 \pmod{12}$ .

Analog rechnet man in der Restklasse modulo 12:

$$5x \cdot 4 \Leftrightarrow x = 5^{-1} \cdot 4 = 4 \cdot 5 = 20 = 8.$$

- Es ist

$$8x^2 - 2x + 3 \equiv -1 \pmod{7}$$

$$\Leftrightarrow (x-1)^2 - 2 + 3 \equiv -1 \pmod{7}$$

$$\Leftrightarrow (x-1)^2 \equiv -2 \equiv 5 \pmod{7}$$

man spricht auch von  
quadratischen Resten  
modulo 7

Da nun wegen  $0^2 \equiv 0 \pmod{7}$ ,  $1^2 \equiv 1 \pmod{7}$ ,  $2^2 \equiv 4 \pmod{7}$ ,  $3^2 \equiv 2 \pmod{7}$  die Zahl 5 kein Quadrat modulo 7 ist, hat die Kongruenz keine Lösung.

Die Kongruenz  $(x - 1)^2 \equiv 4 \pmod{7}$  hat die beiden Lösungen  $x \equiv 3 \pmod{7}$  und  $x \equiv -1 \pmod{7}$ .

- Die Kongruenz  $(x - 3) \cdot 4 \equiv 1 \pmod{33}$  ist schreibbar als System:

$$(x - 3) \cdot 4 \equiv 1 \pmod{3}$$

$$(x - 3) \cdot 4 \equiv 1 \pmod{11}$$

Die beiden einzelnen Kongruenzen haben die Lösungen  $x \equiv 1 \pmod{3}$  sowie  $x \equiv 6 \pmod{11}$ . Mittels chinesischem Restsatz erhält man eine Lösung der Ausgangskongruenz modulo 33:

$$x \equiv 1 \cdot 2 \cdot 11 + 6 \cdot 4 \cdot 3 = 22 + 6 \cdot 12 = 94 \equiv -5 \equiv 28 \pmod{33}$$

- Bei manchen zahlentheoretischen Aufgaben, wie z.B. die Frage, ob es ganzzahlige Lösungen zu bestimmten Gleichungen geben kann, ist die "modulare Brille" ein nützliches Hilfsmittel. Hier ein Beispiel, wo wir die modulare Brille modulo 8 aufsetzen, um mehr zu sehen:

Betrachte die Gleichung  $8x + 7 = u^2 + v^2 + w^2$  in  $u, v, w, x \in \mathbb{N}_0$ . Sie ist unlösbar, denn modulo 8 erhalten wir  $7 \equiv u^2 + v^2 + w^2 \pmod{8}$ . Alle quadratischen Reste modulo 8 sind 0, 1 und 4:

$z$	0	$\pm 1$	$\pm 2$	$\pm 3$	4
$z^2$	0	1	4	1	0

Daher ist  $v^2 + w^2 \equiv 0, 1, 4, 2, 5 \pmod{8}$ , also  $u^2 + v^2 + w^2 \equiv 0, 1, 4, 2, 5, 1, 2, 5, 3, 6, 4, 5, 0, 6, 1 \pmod{8}$ , aber nie  $\equiv 7 \pmod{8}$ . Es kann keine Lösungen modulo 8 geben, also auch keine in  $\mathbb{Z}$ .

### 1.1.3 Gruppen

- [4] Die Gruppen  $(\mathbb{Z}_m, +, 0)$  und  $(\mathbb{Z}_m^*, \cdot)$  sind endliche abelsche Gruppen. Wir untersuchen ein paar ihrer allgemeinen Eigenschaften und führen dabei ein paar Grundbegriffe ein.

#### Definition 4.1 (Gruppenordnung)

Die **Ordnung** einer endlichen Gruppe  $G$  ist die Anzahl ihrer Elemente, kurz  $\text{ord}(G) := \#G$ .

#### Definition 4.2 (Untergruppe)

Eine Teilmenge  $H$  einer Gruppe  $G$  mit Verknüpfung  $*$  heißt **Untergruppe**, falls auch  $(H, *)$  eine Gruppe ist.

#### Satz 4.3 (Satz von Lagrange)

Ist  $(G, *)$  eine endliche Gruppe, so ist die Ordnung einer Untergruppe  $H$  stets ein Teiler von  $\text{ord}(G)$ .

#### Beweis

Die **Linksnebenklassen**  $a * H := \{a * h : h \in H\}$  für  $a \in G$  sind paarweise disjunkt, das heißt es gilt stets  $a * H = b * H$  oder  $a * H \cap b * H = \emptyset$ .

(Denn: ist  $c \in a * H \cap b * H$ , so ist  $c = a * g = b * h$  für  $g, h \in H$ , also  $a = b * (h * g^{-1})$ , somit  $a * H = \{a * m : m \in H\} = \{b * h * g^{-1} * m : m \in H\} = \{b * n : n \in H\} = b * H$ .)

Also ist  $G$  die disjunkte Vereinigung endlich vieler Linksnebenklassen  $a_1 * H, \dots, a_r * H$ . Da  $\#(a * H) = \#H$  für alle  $a \in G$  gilt, folgt mit  $\text{ord}(G) = r \cdot \text{ord}(H)$  die Behauptung.  $\square$

#### Definition 4.4 (Erzeugnis, zyklisch)

Sei  $(G, +)$  eine abelsche Gruppe und  $a \in G$ . Für  $k \in \mathbb{Z}$  definieren wir  $ka := \underbrace{a + \dots + a}_{k\text{-mal}}$ , falls  $k > 0$ ,  $k \cdot 0 := 0$

klar! und  $k \cdot a := -(-k) \cdot a$ , falls  $k < 0$ . Dann ist  $\langle a \rangle := \{ka : k \in \mathbb{Z}\}$  eine Untergruppe von  $G$ . Wir nennen  $\langle a \rangle$  die



von  $a$  **erzeugte Untergruppe** bzw. das **Erzeugnis** von  $a$  und  $a$  einen **Erzeuger**. Ist  $\langle a \rangle$  eine endliche Untergruppe, heißt ihre Ordnung die **Ordnung** von  $a$ , kurz  $\text{ord}(a) := \#\langle a \rangle$ . Eine Gruppe  $G$  mit Erzeuger  $a$ , das heißt  $G = \langle a \rangle$ , heißt **zyklisch**.

Schreibt man die Gruppe multiplikativ mit Verknüpfung  $\cdot$ , so setzt man  $a^k := \underbrace{a \cdot \dots \cdot a}_{k\text{-mal}}$ , falls  $k > 0$ ,  $a^0 := 1$ ,  $a^k := \left(a^{(-k)}\right)^{-1}$ , falls  $k < 0$ , und  $\langle a \rangle := \{a^k : k \in \mathbb{Z}\}$ . Ansonsten ist bis auf Schreibweise die Begrifflichkeit und Theorie zu Erzeugern und Ordnungen dieselbe.

Nach dem Satz von Lagrange gilt für jede endliche Gruppe  $G$  und  $a \in G$  stets  $\text{ord}(a) \mid \text{ord}(G)$ .

#### Beispiel 4.5

$\mathbb{Z}/m\mathbb{Z} = \{0, 1, \dots, m-1\}$  ist "die" zyklische Gruppe mit  $\text{ord}(G) = m$ . Ist  $m = p$  prim, können außer  $\{0\}$  und  $\mathbb{Z}/p\mathbb{Z}$  keine weiteren Untergruppen existieren.

#### Lemma 4.6

Sei  $(G, +)$  eine Gruppe,  $a \in G$ . Es ist  $\text{ord}(a)$  die kleinste natürliche Zahl  $m$  mit  $ma = 0$ .

Es gilt:  $ka = 0 \Leftrightarrow \text{ord}(a) \mid k$ .

(Bei multiplikativer Schreibweise:  $\text{ord}(a) = \min\{m \in \mathbb{N} : a^m = 1\}$  und  $a^k = 1 \Leftrightarrow \text{ord}(a) \mid k$ .)

#### Beweis

Der erste Teil ist klar. Zum zweiten Teil:

" $\Rightarrow$ ": Falls  $k \in \mathbb{N}$  mit  $ka = 0$  ist, nehme Division von  $k$  durch  $\text{ord}(a)$  vor:  $k = q \cdot \text{ord}(a) + r$  mit  $0 \leq r < \text{ord}(a)$ .  
Wegen  $0 = ka = q \cdot \underbrace{\text{ord}(a) \cdot a}_{=0} + ra$  folgt  $ra = 0$ , wegen der Minimalität von  $\text{ord}(a)$  also  $r = 0$ , also  $\text{ord}(a) \mid k$ .

" $\Leftarrow$ ": Für  $k = m \cdot \text{ord}(a)$  folgt  $ka = m \cdot (\text{ord}(a) \cdot a) = 0$ . □

#### Folgerung 4.7

$\text{ord}(G) \cdot a = 0$  bzw. multiplikativ:  $a^{\text{ord}(G)} = 1$ , da  $\text{ord}(a) \mid \text{ord}(G)$  nach Lemma 4.6

#### Folgerung 4.8 (Kleiner Satz von Fermat)

Da  $\text{ord}((\mathbb{Z}/m\mathbb{Z})^*) = \varphi(m)$ , ist  $a^{\varphi(m)} \equiv 1 \pmod{m}$ , falls  $\text{ggT}(a, m) = 1$ . Für  $p$  prim:  $a^{p-1} \equiv 1 \pmod{p}$  für  $p \nmid a$ .

#### Bemerkung 4.9 (Satz von Euler-Fermat)

Die Kongruenz  $a^{\varphi(m)} \equiv 1 \pmod{m}$ , falls  $\text{ggT}(a, m) = 1$ , heißt auch **Satz von Euler-Fermat**. Als Ordnung eines  $a \in \mathbb{Z}_m^*$  (Notation:  $\text{ord}_m(a)$ ) kommt also nur ein Teiler von  $\varphi(m)$  in Frage.

#### Beispiel 4.10

Wir haben  $\varphi(15) = \varphi(3) \cdot \varphi(5) = 2 \cdot 4 = 8$ . Die möglichen Ordnungen von Zahlen  $a \pmod{15}$  mit  $\text{ggT}(a, 15) = 1$  sind also 0, 1, 2, 4, 8.

Wegen  $4^2 = 16 \equiv 1 \pmod{15}$  ist zum Beispiel  $\text{ord}_{15}(4) = 2$ . Bei anderen Zahlen muss man unter Umständen Potenzen mit größeren Exponenten ausrechnen, um die Ordnung zu bestimmen.

Generell stellt sich in Anwendungen die Frage, wie man leicht und schnell (modulare) Potenzen  $a^k \pmod{m}$  mit großem  $k$  berechnen kann. Der Satz von Euler-Fermat erlaubt bereits eine Reduktion von  $k \pmod{\varphi(m)}$ : Ist

$k = q \cdot \varphi(m) + r$  mit  $0 \leq r < \varphi(m)$ , folgt

$$a^k = a^{\varphi(m) \cdot q + r} = \left(a^{\varphi(m)}\right)^q \cdot a^r \equiv 1^q \cdot a^r = a^r \pmod{m}.$$

Ist aber auch  $\varphi(m)$  bzw.  $r$  groß, hilft man sich mit folgender Methode des schnellen Potenzierens weiter:

**Lemma 4.11 (Methode des schnellen Potenzierens)**

Gegeben sei eine Gruppe  $(G, \cdot)$ , zu berechnen ist für  $r \in \mathbb{N}, a \in G$  die Potenz  $a^r$  in der Gruppe  $G$ .

**1. Schritt:** Mit höchstens  $d := \left\lfloor \frac{\log r}{\log 2} \right\rfloor$  vielen Verknüpfungen in  $G$  berechne durch sukzessives Quadrieren  $a^2$ ,  $(a^2)^2 = a^4, \dots, a^{2^d}$ .

**2. Schritt:** Schreibe  $r$  als Binärzahl:  $r = \sum_{i=0}^d c_i \cdot 2^i$  mit  $c_i \in \{0, 1\}$ .

**3. Schritt:** Berechne  $a^r = a^{c_0} \cdot a^{2c_1} \cdot a^{2^2c_2} \dots a^{2^dc_d} = (a^{c_0}) \cdot (a^2)^{c_1} \cdot (a^{2^2})^{c_2} \dots (a^{2^d})^{c_d}$  mit maximal  $d$  weiteren Verknüpfungen in  $G$ .

Somit reichen höchstens  $2d = \mathcal{O}(\log r)$  viele Anwendungen der Gruppenverknüpfung „ $\cdot$ “. Bei additiver Schreibweise einer Gruppe  $(G, +)$  geht das Verfahren zur Berechnung von  $r \cdot a$  analog. Man nennt es dann auch das **dual and add**-Verfahren.

**Beispiel 4.12**

$5^{12} = 5^{2^2+2^3} = 5^{2^2} \cdot 5^{2^3}$ . Modulo 11 rechnen wir:

$$5^2 \equiv \text{mod } 11, 5^{2^2} \equiv 3^2 \equiv -2 \pmod{11}, 5^{2^3} \equiv (-2)^2 \equiv 4 \pmod{11},$$

also  $5^{12} \equiv (-2) \cdot 4 \equiv 3 \pmod{11}$ . Das geht schneller als  $5^{12}$  von Hand auszurechnen und durch 11 zu teilen.

**Anwendung 4.13 (Lösen quadratischer Kongruenzen)**

Im Fall  $p \equiv 3 \pmod{4}$  prim können wir Lösungen quadratischer Kongruenzen modulo  $p$  bestimmen: Sei  $p = 4k + 3$  prim und  $a$  mit  $p \nmid a$  ein quadratischer Rest modulo  $p$ , d.h. es existiert ein  $b \in \mathbb{Z}$  mit  $a \equiv b^2 \pmod{p}$ , und wir möchten  $\pm b \pmod{p}$  ausrechnen können. Nach dem kleinen Fermat folgt  $b^{4k+2} = b^{p-1} \equiv 1 \pmod{p}$ . Es folgt  $(a^{k+1})^2 \equiv (b^2)^{2(k+1)} = b^{(4k+2)+2} \equiv 1 \cdot b^2 \equiv a \pmod{p}$ , d.h. die Lösungen von  $b^2 \equiv a \pmod{p}$  sind  $b = \pm a^{k+1} \pmod{p}$ . Da  $a^{k+1} \not\equiv -a^{k+1} \pmod{p} \Leftrightarrow 2a^{k+1} \not\equiv 0 \pmod{p}$ , gibt es genau zwei Lösungen modulo  $p$ , die wir etwa im Restsystem  $\{0, 1, \dots, p-1\}$  angeben können und mit  $\pm a^{k+1} \pmod{p}$  berechnen können, zum Beispiel mit dem schnellen Potenzieren.

**Anwendung 4.14**

Sei nun  $n$  eine zusammengesetzte Zahl, etwa  $n = pq$  mit  $p \equiv q \equiv 3 \pmod{4}$  prim, etwa  $p = 4k + 3, q = 4l + 3$  mit  $k, l \in \mathbb{N}_0$ , und sei  $p \neq q$ . Sei  $a \pmod{n}$  ein quadratischer Rest modulo  $n$ . Gesucht seien die Lösungen der Kongruenz  $a \equiv x^2 \pmod{n}$ . Nach dem chinesischen Restsatz gilt  $x^2 \equiv a \pmod{n} \Leftrightarrow x^2 \equiv a \pmod{p}$  und  $x^2 \equiv a \pmod{q}$ , und die jeweiligen Lösungen  $\pm a^{k+1} \pmod{p}$  und  $\pm a^{l+1} \pmod{q}$  kann man zusammensetzen zu (maximal) vier Lösungen modulo  $n$ . Es sind genau vier Lösungen, die explizit wie folgt bestimmt werden können:

Sind  $r, s \in \mathbb{Z}$  gegeben mit  $rp + sq = 1$ , d.h. die Bézout-Elemente von  $p$  und  $q$ , und ist  $\pm b$  Lösung von  $x^2 \equiv a \pmod{p}$  sowie  $\pm c$  Lösung von  $x^2 \equiv a \pmod{q}$ , so liefert die Formel des chinesischen Restsatzes

$$x = \pm b \cdot s \cdot q \pm c \cdot r \cdot p$$

genau vier Lösungen von  $x^2 \equiv a \pmod{pq}$ . Diese müssen paarweise inkongruent modulo  $pq$  sein, da wir laut chinesischem Restsatz den Ringisomorphismus  $\mathbb{Z}_{pq} \simeq \mathbb{Z}_p \times \mathbb{Z}_q$  haben und die vier verschiedenen Lösungspaare  $(b, c), (-b, c), (b, -c), (-b, -c)$  deswegen genau vier Restklassen in  $\mathbb{Z}_{pq}$  entsprechen.

**Beispiel 4.15**

Betrachte  $p = 11$ ,  $q = 19$ , d.h.  $k = 2$ ,  $l = 4$ . Wähle  $a = 47$ .

Die Lösungen von  $x^2 \equiv 47 \equiv 3 \pmod{11}$  sind  $\pm 3^3 \equiv \pm 5 \pmod{11}$ , die Lösungen von  $x^2 \equiv 47 \equiv 9 \pmod{19}$  sind  $\pm 3 \pmod{19}$ .

Bézout-Elemente bestimmen: Das Inverse von  $19 \equiv 8 \pmod{11}$  ist 7, das von  $11 \pmod{19}$  ist 7.

$\Rightarrow s = r = 7$  und  $x \equiv \mp 5 \cdot 7 \cdot 19 \pm 3 \cdot 7 \cdot 11 \pmod{(11 \cdot 19)}$  ergibt  $x \in \{\pm 16, \pm 60\}$ .

Probe:  $16^2 \equiv 47 \pmod{(11 \cdot 19)}$ ,  $60^2 \equiv 47 \pmod{(11 \cdot 19)}$ . ✓

Man beachte, dass wir hier benötigen, dass  $a$  ein quadratischer Rest modulo 11 und modulo 19 sein muss. Würde man  $a$  zufällig wählen, wäre das nicht unbedingt der Fall. Dann ist  $x^2 \equiv a \pmod{n}$  ohnehin unlösbar, falls  $a$  kein quadratischer Rest modulo  $11 \cdot 19$  ist.

**Anwendung 4.16 (Faires Münzwurfsknochen)**

Zwei Spieler, Alice (A) und Bob (B), möchten etwas ausknobeln (zum Beispiel, wer beim Fernschach beginnen soll und dann einen Vorteil hat, etc.), allerdings sprechen sie sich am Telefon oder mailen sich, und können sich daher nicht sehen. A wirft eine Münze, und B denkt vorher "Kopf" oder "Zahl", verrät das aber nicht (Würde A die Wahl von B vorher kennen, so würde B das mitgeteilte Ergebnis des Münzwurfs unter Umständen anzweifeln). A teilt B das Ergebnis mit, und B verkündet, wer gewonnen hat: A, wenn ihr Münzwurfergebnis mit der Wahl von B übereinstimmt, ansonsten gewinnt B. Sei B's geheime Wahl "Zahl".

Teilt A mit, dass sie "Zahl" geworfen hat, akzeptieren A und B den Spielausgang, weil dann A gewinnt und B ihr dies verkündet. Falls A jedoch mitteilt, dass sie "Kopf" geworfen hat, teilt B mit, dass A verloren habe, was A natürlich nicht akzeptieren würde.

Problem: Wie kann bei Ergebnis "Kopf" Spieler B seine Mitspielerin A überzeugen, dass er vor dem Münzwurf die Wahl "Zahl" getroffen hat?

Unsere Antwort: Wenn B dann eine Zahl  $n = pq$  faktorisieren könnte, deren Primteiler  $p, q$  ansonsten nur A kennt.

**Erläuterung 4.17**

Das Verfahren funktioniert wie folgt:

**Schritt 1:** A wählt Primzahlen  $p, q \equiv 3 \pmod{4}$ ,  $p \neq q$ , berechnet  $n = pq$  und schickt  $n$  an B.

**Schritt 2:** B wählt  $1 \leq b \leq n - 1$  zufällig und behält  $b$  geheim, er berechnet  $a \equiv b^2 \pmod{n}$  und schickt  $a$  an A.

**Schritt 3:** A berechnet die vier Lösungen von  $x^2 \equiv a \pmod{n}$  (vgl. 4.14), die vier Lösungen seien  $\pm b, \pm c \in \mathbb{Z}$ , (mit  $b$  von B), die Lösungen  $\pm c$  sind andere, die B nicht kennt.

Soweit die Vorbereitung, dann der eigentliche Münzwurf:

**Schritt 4:** A wählt eine der vier Lösungen zufällig aus (etwa durch Münzwurf!), das heißt entweder  $\pm b$  oder  $\pm c$ , und schickt B das Ergebnis. A kann nicht wissen, dass B die Zahl  $b$  gewählt hat. Die Vereinbarung ist nun: Schickt A eine der Zahlen  $\pm b$ , gewinnt A. Schickt A eine der Zahlen  $\pm c$ , gewinnt B, und das verkündet B.

**Schritt 5:** Es erfolgt die Verifikation, dass A wirklich verloren hat im 2. Fall, dazu muss A sich davon überzeugen, dass B vorher wirklich  $\pm b$  gewählt hat. Das kann B nun beweisen, indem er ihr die Primfaktoren von  $n$  nennt: Er berechnet  $b + c \pmod{n}$  und  $d := \text{ggT}(b + c, n)$  mit dem euklidischen Algorithmus. Dann ist  $d = p$  oder  $d = q$ . (Denn aus  $b^2 \equiv a \equiv c^2 \pmod{pq}$  folgt:  $pq \mid (b - c)(b + c) = b^2 - c^2$ , und da  $b \not\equiv c \pmod{p}$ ,  $b \not\equiv c \pmod{q}$  folgt  $q \mid b + c$  oder  $p \mid b + c$ , und  $d \neq n$ , weil sonst  $b \equiv -c \pmod{n}$  wäre  $\frac{1}{2}$ .)

Also kann B, weil er  $c$  kennt, die von A gewählten Primfaktoren bestimmen und A mitteilen und auf diese Art A überzeugen. Das konnte B nur, weil er vorher auch wirklich nicht die von A genannte Lösung  $\pm b$  hatte. Damit ist das Spiel fair.

In der praktischen Umsetzung wird noch ein Verfahren zur Erzeugung großer, möglichst zufälliger Primzahlen  $p, q$  gebraucht. Man kennt in der Praxis schnelle Tests (den Miller-Rabin-Test), um zu entscheiden, ob eine große Zahl  $n$  (mit evtl. hunderten von Stellen in Dezimaldarstellung) zusammengesetzt ist oder (sehr wahrscheinlich) prim. Daher erzeugt man solange Zufallszahlen, bis der Primzahltest "anschlägt".

## 1.2 Public-Key-Kryptographie

**Public-Key-Kryptographie** bezeichnet man auch als asymmetrische Kryptographie. Bei diesem Kommunikationsverfahren hat jeder Nutzer einen **öffentlichen Schlüssel**, den jeder einsehen kann, und einen **privaten Schlüssel**, den jeder Nutzer geheim hält. Möchte Nutzer B eine Nachricht an Nutzer A senden, benutzt er zur Verschlüsselung den öffentlichen Schlüssel von A, die Entschlüsselung gelingt aber nur A mit dem privaten Schlüssel. Ein solches Szenario (auch **Protokoll** genannt) ist das RSA-Verfahren, das wir in Abschnitt 1.2.1 behandeln. Die Verfahren in 1.2.2 und 1.2.3 sind Kryptographie-Verfahren, die mit allgemeinen Gruppen machbar sind (RSA arbeitet  $(\mathbb{Z}_n^*, \cdot)$ ).

[5]

### 1.2.1 RSA-Verfahren

Das **RSA-Verfahren** ist benannt nach einer Arbeit von Rivest<sup>8</sup>, Shamir<sup>9</sup> und Adleman<sup>10</sup> aus dem Jahr 1978. Seine Sicherheit beruht auf der Schwierigkeit des Faktorisierungsproblems und wird bis heute zur sicheren Kommunikation benutzt.

Die Methode verlangt auch die Möglichkeit, große Primzahlen zu erzeugen, die möglichst zufällig gewählt sein sollen, ähnlich wie beim Münzwurfproblem.  $n = pq$  muss so groß sein, dass alle bekannten Faktorisierungsverfahren zu langsam wären.

#### Anwendung 5.1 (Durchführung des RSA-Verfahrens)

Die beiden Protagonisten heißen wieder Nutzer Alice (A) und Bob (B). Sie kommunizieren über einen unsicheren Kanal miteinander.

**Schritt 1:** Jeder Nutzer, z.B. (A), wählt zwei große Primzahlen  $p \neq q$ , etwa gleich groß mit ähnlicher Stellenanzahl, und berechnet  $n = pq$  sowie  $\varphi(n) = (p-1)(q-1)$ . Dann wählt (A) eine Zahl  $e$  mit  $1 < e < \varphi(n)$  und berechnet  $1 < d < \varphi(n)$  als Inverses von  $e \bmod \varphi(n)$ , das heißt  $de \equiv 1 \bmod \varphi(n)$ , unter Zuhilfenahme des euklidischen Algorithmus.

**Schritt 2:** Bob möchte Alice seinen Geheimtext (als eine Zahl  $x$  kodiert) schicken. Er besorgt sich die Daten  $n, e$  vom Server und verschlüsselt  $x$  zu  $x^e \bmod n$ . Dann schickt er ihr das Ergebnis  $v \equiv x^e \bmod n$  zwischen 1 und  $n$ .

**Schritt 3:** Alice entschlüsselt den geheimen Text  $v$  durch Berechnen von  $v^d \bmod n$ , sie erhält  $x$ , weil für ein  $k \in \mathbb{Z}$  gilt:  $ed = 1 + k \cdot \varphi(n)$ , also folgt mit Euler-Fermat, falls  $\text{ggT}(x, n) = 1$ :

$$v^d \equiv (x^e)^d \equiv x^{1+k \cdot \varphi(n)} \equiv x \cdot \underbrace{\left(x^{\varphi(n)}\right)^k}_{\equiv 1 \bmod n} \equiv x \bmod n$$

#### Bemerkung 5.2

Die nötigen Berechnungen sind: schnelles modulares Potenzieren modulo  $n$ , d.h. Berechnungen in der multiplikativen Gruppe  $(\mathbb{Z}_n^*, \cdot)$ , Berechnen von  $d$  mit dem euklidischen Algorithmus und Erzeugen großer Primzahlen  $p, q$ .

#### Bemerkung 5.3

Ein Unbefugter, der die Daten  $n, e, v$  dieser Kommunikation abfängt, ist nicht in der Lage,  $x$  ohne Kenntnis von  $d, p, q, \varphi(n)$  zu berechnen. Dazu müsste man  $n$  faktorisieren.

<sup>8</sup>[http://de.wikipedia.org/wiki/Ronald\\_L.\\_Rivest](http://de.wikipedia.org/wiki/Ronald_L._Rivest)

<sup>9</sup>[http://de.wikipedia.org/wiki/Adi\\_Shamir](http://de.wikipedia.org/wiki/Adi_Shamir)

<sup>10</sup>[http://de.wikipedia.org/wiki/Leonard\\_Adlleman](http://de.wikipedia.org/wiki/Leonard_Adlleman)

**Bemerkung 5.4**

Wie sicher das Verfahren ist, hängt davon ab, wie groß die verwendeten Schlüssel sind. Aktuell ist eine Verschlüsselung, bei der  $p, q$  eine Bitlänge von mindestens 512 haben sollten, besonders sicher: 2048 Bit. Empfehlung der Bundesnetzagentur bis Ende 2020: mindestens 1976 Bit. Gegen einen Angriff mit einem Quantencomputer hat man allerdings keine Chance.

**Bemerkung 5.5**

Auch in den seltenen Fällen  $p \mid x$  oder  $q \mid x$ , das heißt  $\text{ggT}(x, n) > 1$ , arbeitet das Verfahren korrekt (ohne Beweis).

**Bemerkung 5.6**

Das Verfahren kann auch ohne Schlüsselservers benutzt werden. B kann A erst mitteilen, dass er ihr eine Nachricht schicken will. Dann erst erledigt A Schritt 1 und teilt ihm die Daten  $n, e$  mit. Der Rest geht dann wie oben.

**Bemerkung 5.7 (Zur Geschichte von RSA)**

RSA wurde 1983 als Patent angemeldet, welches 2000 erlosch. Bis Ende der 90er Jahre verbot die US-Regierung Firmen, Software mit starker Verschlüsselung zu exportieren (z.B. T-Shirts mit aufgedruckter RSA-Anleitung...). Weiter sollten per Gesetzesvorlage Anbieter elektronischer Kommunikationsdienste dazu verpflichtet werden, Behörden die Möglichkeit zum Zugriff zu verschaffen; das Gesetz scheiterte am Widerstand von Industrie und Bürgerrechtlern. Es motivierte Phil Zimmermann<sup>11</sup> dazu, den Standard **PGP** (pretty good privacy) zu entwickeln, mit dem bis heute E-Mails und anderes für Jedermann sicher verschlüsselt werden können (speziell mit RSA; öffentliche Schlüsselservers dafür gibt es im Internet, z.B. auf [pgp.mit.edu](http://pgp.mit.edu)). Zimmermann stellte sein Programm 1991 kostenlos zur Verfügung. Es wurde ein Verfahren gegen ihn eröffnet, das sich über drei Jahre lang hinzog (Vorwurf: er exportierte Verschlüsselungstechnologie, die wie Waffentechnologie einzustufen sei). Der Fall wurde fallengelassen, heute ist die Benutzung und Export in den USA straffrei. Bis heute zählt PGP als sicherste und empfehlenswerteste Verschlüsselung privater Kommunikation.

**Anwendung 5.8 (Kodierung von Textnachrichten)**

Wir beschreiben hier ein Verfahren, das die Machbarkeit der Kodierung "Text  $\rightarrow$  Zahl" demonstrieren soll. Wenn man es so anwenden möchte, sind aber größere Blöcke erforderlich, damit nicht durch Häufigkeitsanalysen der Blöcke Rückschlüsse auf die Geheimnachricht möglich werden.

Die Buchstaben A, ..., Z des Alphabets werden mit 0, ..., 25 identifiziert, das Leerzeichen mit 26. Klartexte werden zu Blöcken aus je drei Zahlen zusammengefasst, also z.B.

$$\text{KLARTEXT\_} \Rightarrow 10, 11, 0 | 17, 19, 4 | 23, 19, 26$$

Jedem Block  $x_1, x_2, x_3$  ordnen wir die Zahl  $x = x_1 \cdot 27^2 + x_2 \cdot 27 + x_3$  (im 27er System) zu, also

$$\text{KLARTEXT\_} \Rightarrow 7587 | 12910 | 17306,$$

welche beim RSA-Verfahren gemäß  $x^e \equiv v \pmod{n}$  verschlüsselt wird. Jeder Wert  $v$  wird im 29er-System umgewandelt gemäß  $v = v_1 \cdot 29^2 + v_2 \cdot 29 + v_3$  zu einem Block  $v_1, v_2, v_3 \in \{0, \dots, 28\}$ , der wieder als Text geschrieben werden kann (mit zusätzlichen Zeichen für 27 und 28, z.B. "27", "28").

Ist  $n$  zwischen  $27^3$  und  $29^3$ , werden Ver- und Entschlüsselung eindeutig (ohne Beweis)  $\rightsquigarrow$  für größere  $n$  werden größere Blöcke nötig!

**1.2.2 Diffie-Hellman-Verfahren****Definition 5.9 (Das Problem des diskreten Logarithmus (DL-Problem))**

Gegeben sei eine abelsche Gruppe. Wir beschreiben das Problem multiplikativ und additiv:

---

<sup>11</sup>[http://de.wikipedia.org/wiki/Phil\\_Zimmermann](http://de.wikipedia.org/wiki/Phil_Zimmermann)

**In  $(G, \cdot, 1)$ :** Sei  $x \in G, n = \text{ord}(x), y \in \langle x \rangle = \{x^l : l \in \mathbb{Z}\}$ . Bestimme  $k \bmod n$  mit  $y = x^k$ .

**In  $(G, +, 0)$ :** Sei  $x \in G, n = \text{ord}(x), y \in \langle x \rangle = \{lx : l \in \mathbb{Z}\}$ . Bestimme  $k \bmod n$  mit  $y = kx$ .

### Bemerkung 5.10

Ist eine Gruppe  $G$  gegeben, in der das DL-Problem schwer ist, kann dies für ein Kryptoverfahren genutzt werden.

- Im Fall  $G = (\mathbb{Z}_m^*, \cdot, 1)$  ist das DL-Problem ähnlich schwer wie das Faktorisierungsproblem. Auch dafür konnte Short 1994 zeigen, dass es auf einem Quantencomputer schnell lösbar ist.
- Im Fall, dass  $G = (E(k), +, \mathcal{O})$  die Gruppe einer (kryptographisch) geeigneten elliptischen Kurve ist, ist das DL-Verfahren quasi unlösbar. Die besten bekannten Algorithmen sind langsamer als die für das DL-Problem für  $\mathbb{Z}_m^*$ . Darauf beruht die als höher angesehene Sicherheit bei der Kryptographie mit elliptischen Kurven. Algorithmen auf Quantencomputern, die das DL-Problem für elliptische Kurven schnell lösen könnten, sind derzeit unbekannt.

### Anwendung 5.11 (Diffie-Hellman-Schlüsselaustausch)

Hier vereinbaren Alice (A) und Bob (B) durch einen öffentlichen Kanal einen gemeinsamen geheimen Schlüssel, die sie dann für ein symmetrisches Kryptoverfahren nutzen können. Gegeben sei eine Gruppe  $G$  und  $x \in G$ , sowie  $n \in \mathbb{N}$ . Diese Daten seien öffentlich bekannt.

Das Verfahren in  $(G, \cdot, 1)$ :

**Schritt 1:** Alice denkt sich eine Zahl  $a \in \{1, \dots, n-1\}$  und schickt  $x^a \in G$  an Bob.

Bob denkt sich eine Zahl  $b \in \{1, \dots, n-1\}$  und schickt  $x^b \in G$  an Alice.

**Schritt 2:** Alice berechnet mit  $a$  das Gruppenelement  $(x^b)^a$ .

Bob berechnet mit  $b$  das Gruppenelement  $(x^a)^b$ .

Danach besitzen beide den gemeinsamen geheimen Schlüssel  $(x^b)^a = x^{ab} = x^{ba}$ .

Das Verfahren in  $(G, +, 0)$ :

**Schritt 1:** Alice denkt sich eine Zahl  $a \in \{1, \dots, n-1\}$  und schickt  $ax \in G$  an Bob.

Bob denkt sich eine Zahl  $b \in \{1, \dots, n-1\}$  und schickt  $bx \in G$  an Alice.

**Schritt 2:** Alice berechnet mit  $a$  das Gruppenelement  $a \cdot (bx)$ .

Bob berechnet mit  $b$  das Gruppenelement  $b \cdot (ax)$ .

Danach besitzen beide den gemeinsamen geheimen Schlüssel  $a \cdot (bx) = abx = b \cdot (ax)$ .

### Bemerkung 5.12 (Diffie-Hellman-Problem, DH-Problem)

Ein Unbefugter, der die Daten  $x^a, x^b$  bzw.  $ax, bx$  abhört, kann die geheimen Schlüssel berechnen, wenn er das DL-Problem lösen kann. Es genügt aber schon, dafür das folgende, eventuell leichtere Problem zu lösen:

Berechne zu  $x^a, x^b \in \langle x \rangle \subseteq G$  in  $(G, \cdot, 1)$  das Element  $x^{ab} \in \langle x \rangle$ .

Es ist aber davon auszugehen, dass auch DH ein schweres Problem ist.

(Bemerkung: DL lösbar  $\Rightarrow$  DH lösbar ist klar, " $\Leftarrow$ " ist unbekannt.)

### Bemerkung 5.13

Weiter ist beim Schlüsselaustausch entscheidend, dass sich Alice und Bob sicher sein können, wirklich mit dem angegebenen Teilnehmer zu kommunizieren: Ein Unbefugter könnte versuchen, sich erst als Alice auszugeben, und so mit Bob einen Schlüssel  $x^{eb}$  auszutauschen und dies Ebenso mit Alice tun ( $x^{ea}$ ). Gelingt dies, braucht der Unbefugte nur die verschlüsselten Nachrichten zwischen Alice und Bob abzufangen:

Die Nachrichten von Alice an Bob dekodiert er mit dem Alice-Schlüssel  $x^{ea}$ , schickt sie mit dem Bob-Schlüssel  $x^{eb}$  kodiert an Bob weiter, und umgekehrt. Er kann so die gesamte geheime Kommunikation abhören. Man nennt dies eine .

### 1.2.3 ElGamal-Verschlüsselung

- [6] Allen Teilnehmern bekannt sei eine abelsche Gruppe  $(G, +)$  und ein Gruppenelement  $x \in G$  von (großer) Ordnung  $n = \text{ord}(x)$ . Jeder Nutzer wählt eine Zufallszahl  $d \in \{1, \dots, n-1\}$  als privaten Schlüssel und erzeugt einen öffentlichen Schlüssel  $dx$ .

#### Anwendung 6.1 (ElGamal-Verschlüsselung)

Alice möchte eine geheime Botschaft  $m \in G$  an Bob schicken. Die **ElGamal-Verschlüsselung**<sup>12</sup> geht wie folgt:

**Schritt 1:** Alice wählt eine Zufallszahl  $\tilde{a} \in \{1, \dots, n-1\}$  und berechnet  $\tilde{a} \cdot x$ . Alice besorgt sich Bobs öffentlichen Schlüssel  $bx$  und berechnet  $R = \tilde{a} \cdot (bx) + m$ .

**Schritt 2:** Alice schickt  $\tilde{a}x$  und  $R$  an Bob.

**Schritt 3:** Bob berechnet  $b \cdot (\tilde{a}x) = \tilde{a} \cdot (bx)$  und die Nachricht durch  $R - b \cdot (\tilde{a}x) = m$ .

#### Bemerkung 6.2

Ein Unbefugter, der die Daten  $G, x, n, bx, \tilde{a}x$  kennt und  $R$  abgehört hat, kann  $m$  genau dann berechnen, wenn er ein Diffie-Hellman-Problem lösen kann (d.h. das Element  $\tilde{a}b \cdot x \in G$  berechnen kann).

#### Bemerkung 6.3

Alice könnte  $\tilde{a} = a$  wählen. Für die Sicherheit dieses Verfahrens ist es aber wichtig, dass sie bei jeder ihrer Nachrichten ein neues  $\tilde{a}$  wählt: Sonst könnte ein Unbefugter, der die Übertragungen  $\tilde{a}x, R_1 = \tilde{a}(bx) + m_1$  und  $\tilde{a}x, R_2 = \tilde{a}(bx) + m_2$  abhört und schon die Nachricht  $m_1$  kennt, über  $R_2 - R_1 + m_1 = (m_2 - m_1) + m_1 = m_2$  auch  $m_2$  berechnen.

---

<sup>12</sup>[http://de.wikipedia.org/wiki/Taher\\_Elgamal](http://de.wikipedia.org/wiki/Taher_Elgamal)



### 1.3 Digitale Unterschriften

#### 1.3.1 DSA-Signatur

Gegeben sei wieder eine abelsche Gruppe  $(G, +)$ ,  $x \in G$  mit  $n = \text{ord}(x)$  groß. Alice will eine Nachricht  $m$  an Bob digital unterschreiben. Wieder hat sie einen geheimen Schlüssel  $a \in \{1, \dots, n-1\}$  und einen öffentlichen Schlüssel  $ax \in G$ .

#### Definition 6.4 (Hashfunktion)

Sei  $\mathcal{M}$  die Menge aller möglichen Nachrichten (etwa beliebig lange Folgen von 0 und 1), und gegeben sei eine Funktion  $h: \mathcal{M} \rightarrow \{0, 1, \dots, n-1\}$ , deren Werte  $h(m)$  für  $m \in \mathcal{M}$  leicht zu berechnen sind und die die folgenden beiden Eigenschaften hat:

- (i) Es ist praktisch unmöglich, Urbilder unter  $h$  zu berechnen, d.h. zu  $d \in \{0, 1, \dots, n-1\}$  ein  $m \in \mathcal{M}$  zu finden mit  $h(m) = d$ .
- (ii)  $h$  ist **kollisionsresistent**, das bedeutet, dass es praktisch unmöglich ist, zwei verschiedene Elemente  $m, m' \in \mathcal{M}$  mit  $h(m) = h(m')$  zu finden. Eine solche Funktion heißt **Hashfunktion**.

#### Beispiel 6.5

Sei  $p$  prim mit  $2^{1023} < p \leq 2^{1024} - 1$  und  $g$  ein Erzeuger der multiplikativen Gruppe  $\mathbb{Z}_p^*$ , d.h.  $\langle g \rangle = \mathbb{Z}_p^*$ . Dann ist nach heutigem Wissen die Funktion

$$\begin{aligned} h: \mathbb{Z}_p^* &\longrightarrow \mathbb{Z}_p^* \\ z &\longmapsto g^z \bmod p \end{aligned}$$

eine Hashfunktion. Das ab ?? beschriebene Verfahren kann dann mit  $G = \mathbb{Z}_p^*$ ,  $x = g$  durchgeführt werden (in der Praxis nimmt man für  $p$  eine **Sophie-Germain-Primzahl**<sup>13</sup>, d.h.  $p$  prim mit  $\frac{p-1}{2}$  auch prim, denn dann ist etwa jedes zweite Element ein Erzeuger. Daher ist leicht ein Erzeuger findbar.

#### Bemerkung 6.6

Öffentlich zugänglich seien die Daten  $(G, +)$ ,  $x \in G$ ,  $n = \text{ord}(x)$ ,  $h$  und  $ax \in G$  sowie eine Bijektion  $\Psi: \langle x \rangle \rightarrow \{0, 1, \dots, n-1\}$ , deren Werte effektiv berechenbar seien (in der Praxis reicht eine Funktion, deren Urbildmenge  $\Psi^{-1}(k)$  von jedem  $k \in \{0, \dots, n-1\}$  klein ist).

#### Anwendung 6.7 (DSA-Verfahren)

Nun das **DSA-Verfahren** zur Signatur, wie Alice ihre Nachricht  $m$  unterschreiben kann:

**Schritt 1:** Alice wählt eine Zufallszahl  $\tilde{a} \in \{1, \dots, n-1\}$  mit  $\text{ggT}(\tilde{a}, n) = 1$  und berechnet das Gruppenelement  $\tilde{a}x \in G$ .

**Schritt 2:** Alice berechnet das Inverse  $\tilde{a}^{-1}$  in  $\mathbb{Z}_n$  (euklidischer Algorithmus) sowie  $s := \tilde{a}^{-1}(h(m) - \Psi(\tilde{a}x) \cdot a)$  in  $\mathbb{Z}_n$ .

**Schritt 3:** Alice schickt die Nachricht  $m$  und ihre Unterschrift  $\tilde{a}x, s$  an Bob.

**Schritt 4:** Bob berechnet  $\Psi(\tilde{a}x) \cdot ax + s\tilde{a}x$  sowie den Hashwert  $h(m)$ . Bob akzeptiert die Unterschrift als echt, wenn  $\Psi(\tilde{a}x)ax + s\tilde{a}x = h(m) \cdot x$  in  $G$  ist, was nur stimmt, wenn  $\Psi(\tilde{a}x)a + s\tilde{a} \equiv h(m) \bmod n$  gewählt ist, da ja  $n = \text{ord}(x)$  in  $G$  gilt.

<sup>13</sup>[http://de.wikipedia.org/wiki/Sophie\\_Germain](http://de.wikipedia.org/wiki/Sophie_Germain)

**Bemerkung 6.8**

Kann hier ein Unbefugter die Unterschrift von Alice fälschen? Dazu müsste er  $s, k, x$  finden mit  $\Psi(kx)ax + skx = h(m)x$  für ein beliebiges  $k$  anstelle  $\tilde{a}$ . Er würde  $kx$  berechnen und  $s$  passend wählen, wofür ein DL-Problem in  $\langle x \rangle \subseteq G$  zu lösen wäre, denn  $a$  kennt er nicht.

**Bemerkung 6.9**

Auch hier ist für die Sicherheit des Verfahrens nötig, dass Alice für jede Unterschrift ein neues  $\tilde{a}$  wählt: erzeugt Alice zwei Unterschriften  $(\tilde{a}x, s_1)$  für  $m_1$  und  $(\tilde{a}x, s_2)$  für  $m_2$ , ist  $s_2 - s_1 \equiv \tilde{a}^{-1}(h(m_2) - h(m_1)) \pmod{n}$ . Wenn  $h(m_2) - h(m_1)$  invertierbar in  $\mathbb{Z}_n$  ist, kann der Unbefugte  $\tilde{a} \pmod{n}$  berechnen. Wegen  $\Psi(\tilde{a}x)a \equiv h(m_1) - s_1\tilde{a} \pmod{n}$  ist dann auch  $a$  berechenbar, falls  $\Psi(r_1)$  invertierbar in  $\mathbb{Z}_n$  ist.

**Bemerkung 6.10**

Wozu eine Hashfunktion  $h$ ?

- Könnte man leicht Urbilder unter  $h$  berechnen, ist das Unterschriftenfälschen einfach: Der Unbefugte wählt  $j \in \mathbb{Z}$  beliebig und berechnet  $r = jx - ax, s = \Psi(r)$  und bestimmt  $m$  (nicht von Alice!) mit  $h(m) \equiv \Psi(r)j \pmod{n}$ . Dann ist  $r, s$  eine für Bob verifizierbare Unterschrift der falschen Nachricht  $m$ , denn es gilt

$$\Psi(r)ax + \underbrace{\Psi(r)}_s \underbrace{(jx - ax)}_r = \Psi(r)jx = h(m)x$$

- Wäre  $h$  nicht kollisionsresistent und ein Auffinden von  $m' \in \mathcal{M}$  mit  $h(m) = h(m')$  leicht, kann man Alice' Unterschrift unter  $m'$  fälschen, wenn man eine gültige Unterschrift  $\tilde{a}x, s$  für  $m$  hat, da

$$\Psi(\tilde{a}x)ax + s\tilde{a}x = h(m) \cdot x = h(m')x$$

**Bemerkung 6.11**

Bob muss sicher sein, dass Alice' öffentlicher Schlüssel  $ax$  auch wirklich von Alice stammt und nicht von einem Unbefugten gefälscht wurde. Man löst das Problem, indem sich jeder Nutzer bei einer Certification Authority (CA) registrieren lässt. Bob würde von dieser eine "beglaubigte Kopie" von Alice' öffentlichen Schlüssel erhalten; Einzelheiten vgl. Fachliteratur.

**Motivation 6.12**

Eine auf Koblitz und Miller zurückgehende Idee ist nun, dass für die ElGamal-Verfahren eine beliebige zyklische Gruppe  $\langle x \rangle$  verwendbar ist, wie etwa die, die von Punkten auf elliptischen Kurven erzeugt werden. Da für (geeignete) elliptische Kurven das DL-Problem bzw. DH-Problem schwieriger für  $\mathbb{Z}_m^*$  ist, gilt diese Art von Verschlüsselungstechnik heute als besonders sicher und wird vielfältig industriell angewendet. Wir werden die Mathematik elliptischer Kurven im folgenden Abschnitt der Vorlesung näher kennenlernen.

## Index

- absolut kleinster Rest, 11
- Bézout-Elemente, 11
- Caesar-Code, 4
- Charakteristik, 14
- Chinesischer Restsatz, 14
- Diffie-Hellman, 23
- diskreter Logarithmus, 22
- Division mit Rest, 11
- DSA-Verfahren, 25
- dual and add, 18
- ECC-Verfahren, 5
- Einheit, 7
- Einwegfunktion, 5
- ElGamal-Verschlüsselung, 24
- Endziffer, 8
- Erzeuger, 17
- Erzeugnis, 17
- Euklidischer Algorithmus, 11
- Eulersche  $\varphi$ -Funktion, 13
- Exponent, 9
- $g$ -adische Darstellung, 8
- Gruppe, 7
  - abelsch, 7
- größter gemeinsamer Teiler, 10
- Halbgruppe, 7
- Hashfunktion, 25
- Kleiner Satz von Fermat, 17
- kleinster nichtnegativer Rest, 11
- kollisionsresistent, 25
- Kongruenz, 12
- Körper, 7
- Leitziffer, 8
- Linksnebenklassen, 16
- man-in-the-middle-Attacke, 23
- Modul, 12
- $n$ -Bit-Zahl, 8
- Ordnung, 16, 17
- PGP, 22
- Primfaktorzerlegung, 9
- Primzahl, 9
- privater Schlüssel, 21
- Protokoll, 21
- Public-Key-Kryptographie, 21
- Repräsentant, 12
- Restklasse, 12
  - reduziert, prim, 13
- Restsystem
  - reduziert, prim, 13
  - vollständig, 12
- Ring, 7
- RSA-Verfahren, 5, 21
- Satz von Euler-Fermat, 17
- Satz von Lagrange, 16
- schnelles Potenzieren, 18
- Sophie-Germain-Primzahl, 25
- Stellenzahl, 8
- Teiler, 9
- teilerfremd, 10
- Untergruppe, 16
- Zahlkörpersieb, 10
- zusammengesetzt, 9
- zyklisch, 17
- öffentlicher Schlüssel, 21

## Liste der Sätze und Definitionen

Definition 2.1	Halbgruppe . . . . .	7
Definition 2.2	Gruppe . . . . .	7
Definition 2.3	abelsche Gruppe . . . . .	7
Definition 2.4	Ring . . . . .	7
Definition 2.6	Einheit, Einheitengruppe . . . . .	7
Definition 2.7	Körper . . . . .	7
Satz 2.8	. . . . .	8
Definition 2.9	$g$ -adische Darstellung . . . . .	8
Definition 2.12	Teilbarkeit . . . . .	9
Definition 2.14	Primzahl . . . . .	9
Satz 2.15	Satz von der eindeutigen Primfaktorzerlegung, Hauptsatz der Arithmetik . . . . .	9
Definition 2.17	Faktorisierungsproblem . . . . .	10
Definition 2.18	. . . . .	10
Satz 2.19	Teilen mit Rest . . . . .	10
Satz 2.20	Euklidischer Algorithmus . . . . .	11
Lemma 2.21	. . . . .	11
Definition 3.1	Kongruenz, Modul . . . . .	12
Definition 3.3	Restklasse . . . . .	12
Definition 3.7	Addition und Multiplikation auf $\mathbb{Z}_m$ . . . . .	13
Satz 3.10	Einheiten in $\mathbb{Z}_m$ . . . . .	13
Definition 3.11	Prime Reste, Eulersche $\varphi$ -Funktion . . . . .	13
Satz 3.12	Multiplikativität von $\varphi$ . . . . .	14
Definition 3.14	Charakteristik . . . . .	14
Satz 3.15	Chinesischer Restsatz für Zahlringe . . . . .	14
Satz 3.16	Chinesischer Restsatz für simultane Kongruenzen . . . . .	14
Definition 4.1	Gruppenordnung . . . . .	16
Definition 4.2	Untergruppe . . . . .	16
Satz 4.3	Satz von Lagrange . . . . .	16
Definition 4.4	Erzeugnis, zyklisch . . . . .	16
Lemma 4.6	. . . . .	17
Lemma 4.11	Methode des schnellen Potenzierens . . . . .	18
Definition 5.9	Das Problem des diskreten Logarithmus (DL-Problem) . . . . .	22
Definition 6.4	Hashfunktion . . . . .	25