PRIMZAHLEN UND DIE RSA-VERSCHLÜSSELUNG

FLORIAN KRANHOLD
Kurfürst-Salentin-Gymnasium Andernach

Zusammenfassung. Verschlüsselungstechniken und -mechanismen sind aus unserem alltäglichen Leben nicht mehr wegzudenken. Jeder Geldtransfer, jede E-Mail und jede Passworteingabe soll nach Möglichkeit vor dritten geheim bleiben. Dafür müssen Methoden entwickelt werden, um die relevanten Botschaften sicher zu verschlüsseln. Dies ist die Aufgabe der Kryptologie (von griech. κρύπτειν *kryptein* – verbergen).

Die Grundlage für diverse Verschlüsselungsmethoden finden sich in einem wichtigen Gebiet der Mathematik wieder, der Zahlentheorie. Hier beschränken wir uns auf die natürlichen Zahlen und versuchen, – z. B. über die Teilbarkeit – Muster in den Zahlen aufzudecken.

Die Zahlenmenge, die dabei eine große Rolle spielt, ist die der Primzahlen. Primzahlen werden nicht selten als der „Heilige Grahl der Mathematik“ bezeichnet. Sie sind so unregelmäßig auf dem Zahlenstrahl verteilt, dass es bis heute keine Formel gibt, die ihr Auftreten sinnvoll berechnen kann. Zahlentheoretisch gesehen sind die Primzahlen wie die „Atome der Artihmetik“[1] – unteilbar und gleichsam die Grundlage von allem Teilbaren. Auch bei diversen Verschlüsselungsmethoden sind sie unverzichtbar.

Da die Zahlentheorie an sich (die Primzahlen insbesondere) viele interessante Aspekte beinhaltet, bedauere ich es, dass ich sie in dieser Arbeit nur als Mittel zum Zweck verwende, denn das Ziel ist ja eine gute Verschlüsselungsmethode, die auf Zahlentheorie aufbaut. Ich werde allerdings auf diesem Wege dorthin öfters einmal anhalten und die Umgebung beschreiben, sodass es dem Leser möglich sein wird, auch den größeren zahlentheoretischen Kontext zu erkennen.

Ich gehe induktiv vor und von den Grundlagen der Zahlentheorie aus, eben von den Primzahlen. Schritt für Schritt werde ich systematisch einige Muster vorstellen, die uns helfen, neue Muster zu erkennen. Solche erkannten Muster werden als mathematischer Satz formuliert und anschließend logisch bewiesen. Über bewiesene Sätze werden wir uns neue, kompliziertere Sätze herleiten. Diese werden später benötigt, um das RSA-Kryptosystem zu erklären, ein Verschlüsselungssystem, das im Jahre 1977 von Ronald L. Rivest, Adi Shamir und Leonard Adleman entwickelt wurde und heute – aufgrund ihrer Sicherheit – von extrem großer Bedeutung ist.

Inhaltsverzeichnis

1. Primzahlen und andere Zahlen

1.1. Definitionen und Axiome.

Definition 1.1.1. Die Menge $\mathbb{N} := \{1, 2, 3, \dots\}$ ist die Menge der *natürlichen Zahlen*. Für einige Zwecke muss die 0 auch in die Menge einbezogen werden, dann schreibt man $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. Die Menge $\mathbb{Z} := \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$ ist die Menge der *ganzen Zahlen*.

Definition 1.1.2. Eine Zahl $p > 1$, die nur durch 1 und sich selbst teilbar ist, heißt *Primzahl*. Die Menge $\mathbb{P} := \{2, 3, 5, 7, \dots\} \subset \mathbb{N}$ ist die Menge aller Primzahlen.

Definition 1.1.3. Der ggT zweier Zahlen gibt den Wert des größten gemeinsamen Teilers zurück, der kgV den des kleinsten gemeinsamen Vielfachen selbiger.

Axiom 1.1.4 (Kleinstes und größtes Element)**.** *Sei $M \subseteq \mathbb{N}$, $M \neq \emptyset$.*

(i) *Es gibt $m \in M$ so, dass für alle $t \in M$ gilt $t \geq m$. Somit ist m das kleinste Element von M.*
(ii) *Falls M endlich, gibt es ein $m \in M$ so, dass $\forall t \in M$ gilt $t \leq m$. Somit ist m das größte Element von M.*

Axiom 1.1.5. *Sei $x \in \mathbb{N} \setminus \mathbb{P}$. Dann gibt es $a, b \in \mathbb{N}$ mit $1 < a, b < x$ und $x = a \cdot b$.*

Axiom 1.1.6 (Zerlegung)**.** *Sei $x, y \in \mathbb{N}$, $x \leq y$. Dann gibt es $a \in \mathbb{N}$, $r \in \mathbb{N}_0$ mit $y = a \cdot x + r$, also $y \equiv r \mod x$.*

1.2. **Primfaktorzerlegung.**

Satz 1.2.1. *Für jede Zahl $n \in \mathbb{N}$ gibt es mindestens eine Primfaktorzerlegung.*

Beweis. Ang. es gäbe Zahlen, die keine Primfaktorzerlegung (PFZ) haben. Sei $M \neq \emptyset$ die Menge aller solcher Zahlen und $m \in M$ das kleinste Element von M. Dann gibt es zwei Möglichkeiten:

(i) $m \in \mathbb{P}$
Dann wäre $m \notin M$, denn eine Primzahl hat a priori immer eine PFZ, mit einem Primfaktor, nämlich sich selbst.

(ii) $m \notin \mathbb{P}$
Dann lässt sich m faktorisieren: $m =: a \cdot b$ mit $1 < a, b < m$. a und b haben eine PFZ, da sie kleiner als m sind. Somit lässt sich aber auch m, das Produkt von a und b als Produkt von Primzahlen schreiben.

Somit ist $m \notin M$. Somit hat M kein kleinstes Element und ist somit leer. Dies ist aber ein Widerspruch zur Annahme. □

Satz 1.2.2. *Für jede Zahl $n \in \mathbb{N}$ gibt es genau eine Primfaktorzerlegung.*

Lemma 1.2.3. *Für $n, m \in \mathbb{N}$ gilt stets* $\mathrm{kgV}(n, m) \mid n \cdot m$

Beweis des ersten Lemmas. Sei $M := \{v \in \mathbb{N};\ n \mid v \text{ und } m \mid v\}$. Da $M \neq \emptyset$ und $M \subseteq \mathbb{N}$ hat M ein kleinstes Element. Dieses sei v. Per definitionem ist dann $v = \mathrm{kgV}(m, n)$. Es ist $v \leq n \cdot m$. Daher kann man – wegen Axiom 1.1.6 – $n \cdot m$ schreiben als $a \cdot v + r$ mit $a \in \mathbb{N}$, $r \in \mathbb{N}_0$, genauer $0 \leq r < v$.

Da $n, m \mid n \cdot m$, gilt $n, m \mid (a \cdot v + r)$. Und weil $n, m \mid v$, gilt $n, m \mid r$. Wir haben aber festgestellt, dass $r < v$, und v ist die kleinste Zahl, die durch n und m teilbar ist. Folglich muss $r = 0$. Dann ist $n \cdot m = a \cdot v$ und somit $\mathrm{kgV}(n, m) \mid n \cdot m$. □

Lemma 1.2.4. *Sei $p \in \mathbb{P}$ und $p \mid n \cdot m$. Dann gilt $p \mid n$ oder $p \mid m$.*

Beweis des zweiten Lemmas. Sei $v = \mathrm{kgV}(p, n) \overset{\text{L1}}{\Rightarrow} \frac{np}{v} \in \mathbb{N}$. Sei nun $d := \frac{np}{v}$, dann gilt

$$\frac{dv}{n} = p = d \cdot \frac{v}{n}$$

Nun ist $\frac{v}{n} \in \mathbb{N}$ und. $d \in \mathbb{N}$. Das Produkt dieser beiden natürlichen Faktoren soll eine Primzahl ergeben. Hierzu gibt es nur zwei Möglichkeiten:

(i) $d = 1$ und $\frac{v}{n} = p$
Dann ist $v = n \cdot p$. Somit sind n und p teilerfremd. Per definitionem gilt $p \mid n \cdot m$ und weil p nicht in n steckt, kann man das Produkt $n \cdot m$ schreiben als $n \cdot p \cdot k$, $k \in \mathbb{N}$. Das ist äquivalent zu $m = k \cdot p \Rightarrow p \mid m$.

(ii) $d = p$ und $\frac{v}{n} = 1$
Dann ist $v = n$. Da aber wegen der Definition von v gilt $p \mid v$, gilt hier $p \mid n$.

□

Beweis des Satzes. Sei $M \subset \mathbb{N}$ die Menge aller Zahlen, die keine eindeutige PFZ haben. Ang. $M \neq \emptyset$, dann gibt es ein kleinstes Element $m \in M$. Ferner ist m nicht prim (Primzahlen haben eine eindeutige Zerlegung), d. h. $m = a \cdot b$ mit $a, b \in \mathbb{N}$, $1 < a, b < m$.

Nun schauen wir uns an, was es bedeutet, dass m keine eindeutige PFZ hat. Dann müsste m ja auf mindestens zwei verschiedene Weisen als Primprodukt darzustellen sein. Sind zwei Primfaktordarstellungen unterschiedlich, so müssen sie sich mindestens in der Häufigkeit eines Primfaktors p_i unterscheiden. p_i muss also in einer der beiden Darstellungen häufiger vorkommen. Wegen Lemma 2 müsste er dann aber auch in einer der beiden Faktoren von m, also a oder b mit unterschiedlicher Häufigkeit vorkommen. Somit hätte entweder a oder b auch keine eindeutige PFZ und m kann nicht das kleinste Element aus M sein. Das ist ein Widerspruch zur Annahme, somit ist M leer. □

1.3. **Die Unendlichkeit der Primzahlen.**

Satz 1.3.1. *Es gibt unendlich viele Primzahlen.*

Beweis. Ang. $\mathbb{P} = \{p_1, p_2, \ldots\}$ ist endlich. Wegen Axiom 1.1.4 gibt es eine größte Primzahl p_n. Man betrachte nun die Zahl

$$x := 1 + \prod_{i=1}^{n} p_i.$$

x ist um genau 1 größer als eine Zahl, die durch alle Primzahlen teilbar ist. Somit ist sie durch keine Primzahl teilbar. Alle anderen Zahlen $< x$, die mögliche Teiler von x sein könnten, haben als Faktor wenigstens eine Primzahl p_i, $i < n$, somit muss x selbst eine Primzahl sein. x ist aber größer als die größte Primzahl p_n. Somit ist p_n nicht die größte Primzahl. Das ist ein Widerspruch, also ist $\mathbb{P}$ nicht endlich. □

2. Der Satz von Euler

2.1. **Die Eulersche Φ-Funktion.**

Definition 2.1.1. Für $n \in \mathbb{N}$ sei $\Phi(n)$ die Anzahl aller Zahlen $a < n$, die zu n teilerfremd sind, also

$$\Phi(n) = \#\left\{1 \leq a < n;\, \mathrm{ggT}(a, n) = 1\right\}.$$

Satz 2.1.2. *Sei $p_1, p_2 \in \mathbb{P}$. Dann gilt $\Phi(p_1 \cdot p_2) = (p_1 - 1) \cdot (p_2 - 1)$*

Beweis. Alle Zahlen $n < p_1 \cdot p_2$, die durch p_1 teilbar sind, müssen die Form $a \cdot p_1$, $1 \leq a < p_2$ haben. Dies sind genau $p_2 - 1$ Möglichkeiten für n. Alle Zahlen $m < p_1 \cdot p_2$, die durch p_2 teilbar sind, müssen die Form $b \cdot p_2$, $1 \leq b < p_1$ haben. Dies sind genau $p_1 - 1$ Möglichkeiten für m. Da $p_1 \cdot p_2$ die kleinste Zahl ist, die sowohl durch p_1 als auch durch p_2 teilbar ist, gibt es für die Möglichkeiten für m und n keine Überschneidungen.

Aufgrund der eindeutigen PFZ gilt für alle Zahlen t, die weder durch p_1 noch durch p_2 teilbar sind, dass $\mathrm{ggT}(t, p_1 \cdot p_2) = 1$. Dies sind alle Zahlen von 1 bis $p_1 \cdot p_2 - 1$ abzgl. der Möglichkeiten für m und n, daher ist $\Phi(p_1 \cdot p_2) = p_1 \cdot p_2 - 1 - (p_1 - 1) - (p_2 - 1) = (p_1 - 1) \cdot (p_2 - 1)$. □

2.2. **Der Satz von Euler.**

Satz 2.2.1. *Sei $n, a \in \mathbb{N}$, $a < n$ und $\mathrm{ggT}(a, n) = 1$, dann ist $a^{\Phi(n)} \equiv 1 \mod n$.*

Lemma 2.2.2. *Sei $M := \{1 \leq m < n;\, \mathrm{ggT}(r; n) = 1\}$. Dann gilt für alle $m \in M$: $(a \cdot m \mod n) \in M$.*

Beweis des Lemmas. Aufgrund der eindeutigen PFZ gilt $\mathrm{ggT}(a \cdot m, n) = 1$ für alle $m \in \mathbb{N}$. Es gibt nun zwei Möglichkeiten:

(i) $a \cdot m < n$
Dann gilt $a \cdot m = (a \cdot m \mod n) \in M$

(ii) $a \cdot m > n$

So lässt sich $a \cdot m \overset{A4}{=} k \cdot n + r$, $k \in \mathbb{N}$, $r \in \mathbb{N}$ (ohne 0 wegen der Teilerfremdheit) und $r = (a \cdot m \mod n)$. Ang. $r \notin M$. Dann ist aber $\mathrm{ggT}(r, n) > 1$. Sei also $\mathrm{ggT}(r; n) = t$ mit $t > 1$, so gilt:

$$\frac{a \cdot m}{t} = k \cdot \frac{n}{t} + \frac{r}{t}$$

Aufgrund der angenommenen gemeinsamen Teilbarkeit durch t entspricht die rechte Seite der Gleichung dem Wert einer natürliche Zahl. Das bedeutet aber, dass $t \mid a \cdot m$, und somit wäre $\mathrm{ggT}(a \cdot m, n) > 1$. Das ist aber ein Widerspruch. Somit ist $(a \cdot m \mod n) < n$ und $\mathrm{ggT}((a \cdot m \mod n), n) = 1$, also gilt $(a \cdot m \mod n) \in M$.

□

Beweis des Satzes. Per definitionem ist a teilerfremd zu n. Da $a \cdot p_i \equiv a \cdot p_j \mod n$, also $p_i \equiv p_j \mod n$, muss auf M die Zuordnung $M \ni m \mapsto (am \mod n) \in M$ permutativ sein. Das heißt aber

$$\prod_{i=1}^{\Phi(n)} (a \cdot m_i) \equiv \prod_{i=1}^{\Phi(n)} m_i \mod n.$$

Da $\mathrm{ggT}\left(\prod_{i=1}^{\Phi(n)} m_i, n\right) = 1$ (PFZ), kann die Gleichung durch $\prod_{i=1}^{\Phi(n)} m_i$ geteilt werden zu

$$\prod_{i=1}^{\Phi(n)} a \equiv 1 \mod n \Leftrightarrow a^{\Phi(n)} \equiv 1 \mod n.$$

□

3. Die RSA-Verschlüsselung

3.1. **Einführung in die Methoden der Kryptologie.** Die Wissenschaft der Kryptologie beschäftigt sich damit, Informationen sicher und für dritte unlesbar zu transportieren. Der Klartext muss auf irgendeine Weise verschlüsselt werden, sodass er wieder entschlüsselt werden muss, damit er lesbar ist.

Hier gibt es im Wesentlichen zwei grundlegend verschiedene Möglichkeiten: Substitution und Transposition. Während bei der Subsitution die einzelnen Zeichen durch andere ersetzt werden, ändert man bei der Transposition die Reihenfolge. Das RSA-Kyptosystem baut auf der Substitution auf.

Die Schwierigkeit dabei ist, dass der Empfänger der Nachricht sie wieder entschlüsseln können muss, während es für jeden anderen unmöglich sein soll. Wenn aber bekannt ist, wie eine Nachricht verschlüsselt wurde, kann man bei einfachen Kryptosystemen (= Verschlüsselungsmethoden) eine inverse Methode zur Entschlüsselung finden und die Nachricht wäre nicht mehr geheim. Für jede Korrespondenz ein eigenes Kryptosystem zu entwickeln wäre denkbar unpraktisch.

Man versucht also nun, ein Kryptosystem zu entwickeln, wo die reine Kenntnis über die Methode nicht ausreicht, um sinnvoll zu entschlüsseln. Nur der Empfänger, der einen bestimmten „Schlüssel“ hat, soll dies können.

Hierbei ist die Mathematik gefragt, insbesondere die Zahlentheorie. Man transferriert sprachliche Informationen in Form von Buchstaben, Zahlen und Sonderzeichen in arithmetische Informationen, natürliche Zahlen. So eine Zahl, die durch eine einheitliche Buchstabenkodierung (z.B. ASCII) als **Klartext** gelesen werden kann, wird nun durch mathematische Zuordnungen (Chiffrierung) in eine andere Zahl transferriert, die zwar immer noch die gleichen Informationen enthält, allerdings nicht durch eine Buchstabenkodierung wieder in Klartext übersetzt werden kann. Diese verschlüsselte Form heißt **Code**. Erst durch eine geeignete Invers-Zuordnung kann die ursprüngliche Zahl wieder in „Klartext“ übersetzt werden.

Eines dieser Ver- und Entschlüsselungsverfahren ist eben das RSA-Verfahren. RSA wurde von Ronald L. Rivest, Adi Shamir und Leonard Adleman im Jahre 1977 am Massachusetts Institute of Technology entwickelt.

3.2. **Das RSA-Kryptosystem im Überblick.** Alice möchte Bob eine Nachricht schicken, die für dritte nicht lesbar ist. Man geht nun wie folgt vor:

(i) Alice sucht zwei möglichst große Primzahlen p_1 und p_2. Aus ihnen wird das Produkt gebildet. Dies ist das RSA-Modul $N = p_1 p_2$ und ist für jeden (also auch für dritte) zugänglich (= „public key“, öffentlicher Schlüssel)

(ii) Sie berechnet $\Phi(N) = (p_1 - 1) \cdot (p_2 - 1)$

(iii) Sie sucht ein v, das teilerfremd zu $(p_1 - 1) \cdot (p_2 - 1)$ ist.

(iv) Es wird e als multiplikatives Inverses von v auf $\mathbb{N}/\Phi(p_1 \cdot p_2)\mathbb{N}$ ermittelt. Für e muss also gelten:

$$v \cdot e \equiv 1 \mod \Phi(p_1 \cdot p_2)$$

Danach werden die beiden Primzahlen p_1 und p_2 sowie $\Phi(N)$ sicher gelöscht. Es verbleiben die folgenden Zahlen:

(i) Das RSA-Modul $N = p_1 \cdot p_2$, *public key*
(ii) Der Verschlüsselungsexponent v, *public key*
(iii) Der Entschlüsselungsexponent e, *private key* (erhält nur der Empfänger)

Der öffentliche Schlüssel ist also das Zahlenpaar (N, v), welches Alice zum Verschlüsseln verwendet, der private Schlüssel das Zahlenpaar (N, e), das Bob hat. Für eine beidseitige Kommunikation benötigt jede Partei einen privaten Schlüssel zum Entschlüsseln.

Dieses Verfahren gehört zu den *asymmetrischen Kryptosystemen*, da der private Schlüssel aus einer öffentlichen Zahl (N) und einer privaten Zahl (e) besteht. Im Gegensatz zu *symmetrischen Kryptosystemen* ist der private Schlüssel nicht für beide – Sender und Empfänger – gleich.

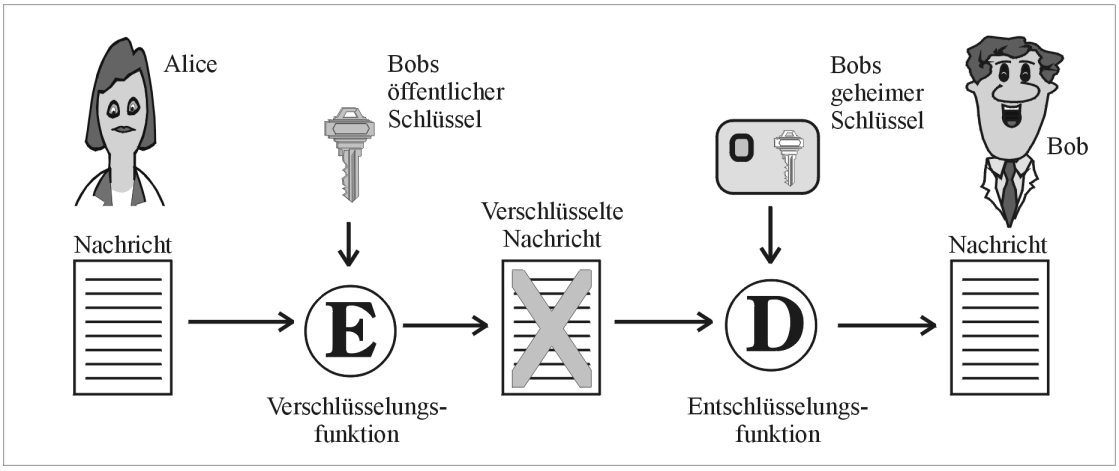

Abb. 1. Prinzip asymmetrischer Kyptosysteme[2]

Wichtig ist, dass zur Bestimmung von v und e das RSA-Modul nicht ausreicht. Hierfür muss man $\Phi(N)$ kennen und dazu die beiden Primfaktoren von N. Ein großes Modul aber wieder zu faktorisieren ist eine algorithmisch äußerst aufwendige Aufgabe, sodass es sehr lange dauern würde, nur mit Kenntnis von v und N den Entschlüsselungsexponenten e zu finden. Diese mathematische Eigenschaft macht dieses Kryptosystem so sicher.

3.3. **Die Ver- und Entschlüsselungsfunktion.** Mit den ermittelten Zahlen ist gemäß der folgenden Sätze die Bildung einer Ver- und einer Entschlüsselungsfunktion möglich:

Satz 3.3.1. *Sei* $p_1, p_2 \in \mathbb{P}$, $v \in \mathbb{N}$, $\mathrm{ggT}\,(v; \Phi(p_1 \cdot p_2)) = 1$. *Und sei* V *wie folgt definiert:*

$$V : \mathbb{N}/(p_1 \cdot p_2)\mathbb{N} \ni x \mapsto x^v \in \mathbb{N}/(p_1 \cdot p_2)\mathbb{N}$$

[2]informatik.tu-darmstadt.de/BS/Lehre/Sem98_99/T11/abb1.gif (13. Januar 2011)

Sei weiter $e \in \mathbb{N}/\Phi(p_1 \cdot p_2)\mathbb{N}$ multiplikatives Inverses von v auf $\mathbb{N}/\Phi(p_1 \cdot p_2)\mathbb{N}$, d. h. $v \cdot e \equiv 1 \mod \Phi(p_1 \cdot p_2)$, und E wie folgt definiert:

$$E : \mathbb{N}/(p_1 \cdot p_2)\mathbb{N} \ni x \mapsto x^e \in \mathbb{N}/(p_1 \cdot p_2)\mathbb{N}$$

Dann ist E die Umkehrfunktion von V, es gilt $E(V(x)) \equiv x \mod p_1, p_2$.

Beweis. Durch die Wahl von e ist $v \cdot e \equiv 1 \mod \Phi(p_1 \cdot p_2) \overset{A4}{\Rightarrow} \exists k \in \mathbb{N} : v \cdot e = k \cdot \Phi(p_1 \cdot p_2) + 1$. Aus der Definition von V und E folgt $E(V(x)) = x^{e \cdot v}$. Somit gilt

$$E(V(x)) = x^{1+k \cdot \Phi(p_1 \cdot p_2)} = x \cdot \left(x^{\Phi(p_1 \cdot p_2)}\right)^k.$$

Nach dem Satz von Euler gilt hier

$$x \cdot \left(x^{\Phi(p_1 \cdot p_2)}\right)^k \equiv x \cdot 1^k \mod p_1 \cdot p_2 \equiv x \mod p_1 \cdot p_2.$$

□

3.4. **Algorithmen zur schnellen Findung geeigneter Zahlen.**

Bemerkung 3.4.1. Möchte man das obige Verfahren zur Verschlüsselung für lange Nachrichten anwenden, ist es wichtig, effiziente Algorithmen zu entwerfen, mittels derer geeignete Werte für v und e ermittelt werden können

Algorithmus 3.4.2 (Euklid)**.** *Seien $a, b \in \mathbb{N}$. Wähle nun $q_1, r_1 \in \mathbb{N}$ so, dass $a = q_1 \cdot b + r_1$ und $r_1 < b$. Nun geht es weiter, b wird dargestellt als $b = q_2 \cdot r_1 + r_2$ mit $r_2 < r1$. Ab jetzt wird die Iteration*

$$r_{k-2} = q_k \cdot r_{k-1} + r_k$$

so lange weiter geführt, bis $r_k = 0$. Dann gilt $r_{k-1} = \mathrm{ggT}(a, b)$.

Lemma 3.4.3. *Bei einer Division mit Rest, wie oben $r_{k-2} = q_k \cdot r_{k-1} + r_k$, ist immer $\mathrm{ggT}(r_{k-2}; r_{k-1}) = \mathrm{ggT}(r_{k-1}; r_k)$.*

Beweis des Lemmas. Sei $\mathrm{ggT}(r_{k-1}, r_k) =: d$. Dann sei $r_{k-1} := d \cdot x_{k-1}$ und $r_k := d \cdot x_k$ mit $x_{k-1}, x_k \in \mathbb{N}$. So lässt sich die Gleichung schreiben als $r_{k-2} = q_k \cdot dx_{k-1} + dx_k = d(q_k x_{k-1} + x_k)$. Dann muss aber auch $d \mid r_{k-2}$ und somit ist $\mathrm{ggT}(r_{k-2}, r_{k-1}) \geq \mathrm{ggT}(r_{k-1}; r_k)$.

Sei weiter $\mathrm{ggT}(r_{k-2}, r_{k-1}) =: e$ und $r_{k-2} := e \cdot y_{k-2}$ und $r_{k-1} := e \cdot y_{k-1}$ mit $y_{k-2}, x_{k-1} \in \mathbb{N}$. So lässt sich die Gleichung schreiben als $ey_{k-2} = q_k \cdot ey_{k-1} + r_k \Leftrightarrow r_k = e(y_{k-2} - q_k y_{k-1})$. Dann muss aber auch $e \mid r_k$, somit ist $\mathrm{ggT}(r_{k-1}; r_k) \geq \mathrm{ggT}(r_{k-2}, r_{k-1})$. □

Beweis des Algorithmus. Es gilt am Ende $\mathrm{ggT}(r_{k-1}, r_k) = \mathrm{ggT}(r_{k-1}, 0) = r_{k-1}$. Wegen Lemma 3.4.3 ist $\mathrm{ggT}(r_{k-1}, r_k) = \mathrm{ggT}(r_{k-2}, r_{k-1}) = \cdots = \mathrm{ggT}(b, r_1) = \mathrm{ggT}(a, b)$. Somit ist $\mathrm{ggT}(a, b) = r_{k-1}$. □

Satz 3.4.4 (Lemma von Bézout)**.** *Wenn $\mathrm{ggT}(a, b) = d$, dann gibt es $x, y \in \mathbb{Z}$, für die gilt $ax + by = d$.*

Beweis. Für $r \in \mathbb{N}$ sei $r \cdot \mathbb{Z} = \{r \cdot s;\ s \in \mathbb{Z}\}$ und weiter

$$a\mathbb{Z} + b\mathbb{Z} := \{ax + by;\ x, y \in \mathbb{Z}\}$$

Es ist zu zeigen: Ist $\mathrm{ggT}(a; b) = d$, dann gilt $d\mathbb{Z} = a\mathbb{Z} + b\mathbb{Z}$. Dies teilen wir in zwei Teile auf:

(i) Zu Zeigen: $a\mathbb{Z} + b\mathbb{Z} \subseteq d\mathbb{Z}$
Da $\mathrm{ggT}(a, b) = d$, ist $a =: d \cdot s$ und $b =: d \cdot t$ mit $s, t \in \mathbb{N}$. Somit ist $ax + by = dsx + dty = d(sx + ty) \in d\mathbb{Z}$.

(ii) Zu Zeigen: $d\mathbb{Z} \subseteq a\mathbb{Z} + b\mathbb{Z}$
Wir zeigen zunächst, dass beim euklidschen Algorithmus für jeden Rest gilt:

$$r_n \in a\mathbb{Z} + b\mathbb{Z}$$

Dies machen wir per Induktion nach n

(a) *Induktionsanfang* „$n = 1$“
Für $n = 1$ gilt $r_1 = a + (-q_1) \cdot b \in a\mathbb{Z} + b\mathbb{Z}$

(b) *Induktionsschritt* „$1, \ldots, n-1 \to n$“
Gelte die Aussage bis einschließlich $n-1$, dann ist auch $r_{n-1} \in a\mathbb{Z} + b\mathbb{Z}$ und $r_{n-2} \in a\mathbb{Z} + b\mathbb{Z}$. Sei nun $r_{n-1} = ax_1 + by_1$ und $r_{n-2} = ax_2 + by_2$. Es ist

$$\begin{aligned} r_n &= r_{n-2} - q_n r_{n-1} \\ &= ax_2 + by_2 - q_n(ax_1 + by_1) \\ &= a(x_2 - q_n x_1) + b(y_2 - q_n y_1) \in a\mathbb{Z} + b\mathbb{Z} \end{aligned}$$

Somit gilt auch für $r_{k-1} = \mathrm{ggT}(a; b)$, dass $r_{k-1} \in a\mathbb{Z} + b\mathbb{Z}$. Da $r_{k-1} = d$, gilt $d\mathbb{Z} \subseteq a\mathbb{Z} + b\mathbb{Z}$
Da nun $a\mathbb{Z} + b\mathbb{Z} \subseteq d\mathbb{Z}$ und $d\mathbb{Z} \subseteq a\mathbb{Z} + b\mathbb{Z}$, ist $d\mathbb{Z} = a\mathbb{Z} + b\mathbb{Z}$. □

Bemerkung 3.4.5 (Erweiterter euklidischer Algorithmus)**.** Mit dem erweiterten euklidschen Algorithmus ist es nun nicht nur möglich, den ggT großer Zahlen zu bestimmen sondern auch die ganzen Zahlen x und y aus dem Lemma von Bézout zu ermitteln, für die dann gilt

$$ax + by = \mathrm{ggT}(a, b)$$

Sagen wir, wir haben den einfachen euklidschen Algorithmus durchgeführt und sind bei $r_k = 0$, also $r_{k-1} = \mathrm{ggT}(a, b)$. Aufgrund der Iterationsdefinition gilt $r_{k-1} = r_{k-3} - q_{k-1} r_{k-2}$, also kann man r_{k-1} darstellen als $x r_{k-3} + y r_{k-2}$ mit $x, y \in \mathbb{Z}$. nun kann man r_{k-2} weiter zerteilen, sodass sich insgesammt ergibt

$$\begin{aligned} \mathrm{ggT}(a; b) = r_{k-1} &= r_{k-3} - q_{k-1} r_{k-2} \\ &= r_{k-3} - q_{k-1}(r_{k-4} - q_{k-2} r_{k-3}) \\ &= (1 + q_{k-1} q_{k-2}) r_{k-3} - q_{k-1} r_{k-4}. \end{aligned}$$

So hat man r_{k-1}, also den ggT dargestellt als $x r_{k-4} + y r_{k-3}$ mit $x, y \in \mathbb{Z}$ und ebenso kann man immer weiter verfahren und gelangt irgendwann zu a und b, sodass die ganzen Zahlen aus dem Lemma von Bézout durch diesen Algorithmus sehr schnell ermittelt werden können.

Bemerkung 3.4.6 (Binäre Exponentiation)**.** Für die Entwicklung der Ver- und Entschlüsselungsfunktion, die oben vorgestellt wurde, ist es nötig, den Rest von sehr großen Potenzen n einer Zahl x beim Teilen durch eine andere Zahl m zu ermitteln, also $x^n \mod m$. Dies geht relativ einfach über das Verfahren der „binären Exponentiation“, und zwar wird n zerlegt in eine Summe aus Zweierpotenzen. $n = 2^{e_1} + 2^{e_2} + \ldots$, wobei $e_i \in \mathbb{N}_0$. Dann kann die Zahl x^n dargestellt werden als

$$\begin{aligned} x^n &= x^{2^{e_1} + 2^{e_2} + \ldots} \\ &= \left(\left(\left(x^2\right)^2\right)^2\right)^{\cdots} \cdot \left(\left(x^2\right)^2\right)^{\cdots} \cdot \ldots \end{aligned}$$

Möchte man nun $x^n \mod m$ berechnen, kann man dies durch die Rechenregeln der Modulorechnung umschreiben zu

$$\begin{aligned} & x^n \mod m \\ &= \left(\left(\left(x^2\right)^2\right)^2\right)^{\cdots} \cdot \left(\left(x^2\right)^2\right)^{\cdots} \cdot \ldots \mod m \\ &= \left[\left[\left(\left(\left(x^2\right)^2\right)^2\right)^{\cdots} \mod m\right] \cdot \left[\left(\left(x^2\right)^2\right)^{\cdots} \mod m\right] \cdot \ldots\right] \mod m. \end{aligned}$$

Beispiel 3.4.7. Sei $x = 7$, $n = 33$, $m = 35$. Den Rest von 7^{33} beim Teilen durch 35 zu ermitteln ist wohl relativ zeitaufwendig (alleine schon, 7^{33} zu berechnen). n kann aber zerlegt werden in

$n = 2^5 + 2^0$, So kann $7^{33} \mod 35$ geschrieben werden als

$$\begin{aligned}
7^{33} \mod 35 &= \left(\left(\left(\left(7^2\right)^2\right)^2\right)^2\right) \cdot 7 \mod 35 \\
&= \left(\left(\left(\left(7^2 \mod 35\right)^2\right)^2\right)^2\right) \cdot 7 \mod 35 \\
&= \left(\left(\left(14^2 \mod 35\right)^2\right)^2\right) \cdot 7 \mod 35 \\
&= \left(\left(29^2 \mod 35\right)^2\right)^2 \cdot 7 \mod 35 \\
&= \left(1^2 \mod 35\right)^2 \cdot 7 \mod 35. \\
&= 7 \mod 35
\end{aligned}$$

3.5. **Anwendungsbeispiel.** Wir wollen die Nachricht „Hi.“ mithilfe des RSA-Verfahrens verschlüsseln. Dazu müssen die verbalen Informationen in artihmetische Informationen transferriert werden. Hierfür entwerfen wir eine Zuordnungstabelle:

A	...	Z	a	...	z	0	...	9	.		:	,	;
01	...	26	27	...	52	53	...	62	63	64	65	66	67

Ein zu verschlüsselnder Buchstabe beansprucht immer zwei Ziffern. Unser Satz wird also laut unserer obigen Tabelle zu folgender Zahl:

H	i	.
08	35	80

$x = 83580$. Für die beiden Primzahlen p_1, p_2 muss $p_1 p_2 > x$, da $x \in \mathbb{N}/(p_1 p_2)\mathbb{N}$. Um die Primzahlen möglichst klein zu halten, sollten sie möglichst nah beieinander liegen, also etwa um $\sqrt{x} \approx 289,1$. Primzahlen in dieser Umgebung wären z.B. $p_1 = 293$ und $p_2 = 307$. Nach folgendem Verfahren wird nun verschlüsselt:

(i) Berechne das RSA-Modul $N = p_1 p_2 = 293 \cdot 307 = 89951$.

(ii) Berechne $\Phi(N) = (293 - 1)(307 - 1) = 89352$.

(iii) Finde eine zu 89352 teilerfremde Zahl, wir wählen $v = 23$.

(iv) Ermittle e als multiplikatives Inverses von v auf $\mathbb{N}/\Phi(p_1 p_2)\mathbb{N}$:
Da e multiplikatives Inverses von v sein soll, muss für ein geeignetes $k \in \mathbb{Z}$ gelten: $e \cdot v + k \cdot \Phi(N) = 1$ Somit sind e und k die Koeffizienten aus dem euklidschen Algorithmus zur Ermittlung von $\mathrm{ggt}(v; \Phi(N))$.

i	r_{i-2}	r_{i-1}	r_i	q_i
1	89352	23	20	3884
2	23	20	3	1
3	20	3	2	6
4	3	2	1	1
5	2	$\underline{1}$ (= ggT)	0	2

t Um nun die Koffizienten zu ermitteln, lösen wir rückwärts auf:

$$\begin{aligned}
1 &= 3 - 1 \cdot 2 \\
&= 23 - 1 \cdot 20 - 1 \cdot (20 - 6 \cdot 3) \\
&= 23 - 1 \cdot (89352 - 3884 \cdot 23) - 1 \cdot (89352 - 3884 \cdot 23 - 6 \cdot (23 - 20)) \\
&= 23 \cdot 31079 - 8 \cdot 89352
\end{aligned}$$

Somit ist $e = 31079$.

Nun lauten die Abbildungen wie folgt:

$$\begin{aligned} \text{Verschlüsselungsfunktion} &: \mathbb{N}/89951\mathbb{N} \ni x \mapsto x^{23} \in \mathbb{N}/89951\mathbb{N} \\ \text{Entschlüsselungsfunktion} &: \mathbb{N}/89951\mathbb{N} \ni x \mapsto x^{31079} \in \mathbb{N}/89951\mathbb{N} \end{aligned}$$

Folgende Schlüssel werden vergeben, die übrigen gelöscht:

$N = 89951$	public key	[öffentlich]
$v = 23$	private key	Sender
$e = 31079$	private key	Empfänger

Mit der Verschlüsselungsfunktion wird nun die Nachricht $x = 83580$ verschlüsselt, dies geht mit binärer Exponentiation am schnellsten.

$$\begin{aligned} (83580^{23} \mod 89951) &= \left(\left(\left(83580^2\right)^2 \cdot 83580\right)^2 \cdot 83580\right)^2 \cdot 83580 \mod 89951 \\ &= \left(\left((21740)^2 \cdot 83580\right)^2 \cdot 83580\right)^2 \cdot 83580 \mod 89951 \\ &= \left((5008)^2 \cdot 83580\right)^2 \cdot 83580 \mod 89951 \\ &= (763)^2 \cdot 83580 \mod 89951 \\ &= 40835 \end{aligned}$$

Die verschlüsselte Nachricht ist nun $y = 40835$. Diese kann nur vom Empfänger mit dem richtigen Entschlüsselungsexponenten e entschlüsselt werden:

$$(40835^{31079} \mod 89951) = 83580 = x$$

(Auch hier wurde wieder binäre Exponentiation angewandt, aufgrund der Länge der Binärdarstellung von e wurde aber auf genauere Ausführungen verzichtet.)

Wie hier nochmal verdeutlicht wurde, lassen sich also mithilfe des RSA-Kryptosystems allerlei Informationen schnell und sicher verschlüsseln. Sie findet in der Praxis häufig Anwendung, da der Algorithmus als Einwegfunktion fungiert. Wie wir gesehen haben, ist die Verschlüsselungsfunktion – komplextheoretisch betrachtet – zwar seht leicht anzuwenden, aber nur sehr schwer umzukehren.