in $\mathbb{R}$ kein Quadrat ist. Ein solches Element gibt es in $\mathbb{F}_2$ nicht: Jedes Element ist sein eigenes Quadrat. Daher muß entweder $\alpha^2 = \alpha$ oder $\alpha^2 = 1 + \alpha$ sein.

Wäre $\alpha^2 = \alpha$, so wäre $\alpha(\alpha - 1) = 0$, d.h. $\alpha = 0$ oder $\alpha = 1$, was wir natürlich nicht wollen. Also müssen wir

$$\alpha^2 = \alpha + 1$$

setzen. Damit ist dann alles klar, und wir erhalten die folgende Additions- und Multiplikationstafel, die uns insbesondere auch zeigen, daß wir tatsächlich einen Körper konstruiert haben:

+	0	1	α	$1+\alpha$
0	0	1	α	$1+\alpha$
1	1	0	$1+\alpha$	α
α	α	$1+\alpha$	0	1
$1+\alpha$	$1+\alpha$	α	1	0

und

·	0	1	α	$1+\alpha$
0	0	0	0	0
1	0	1	α	$1+\alpha$
α	0	α	$1+\alpha$	1
$1+\alpha$	0	$1+\alpha$	1	α

Dieser Körper wird üblicherweise mit $\mathbb{F}_4$ bezeichnet: Das $\mathbb{F}$ steht für *finite*, und vier ist ist die Anzahl der Elemente.

Allgemein bezeichnet man einen endlichen Körper mit q Elementen, so es einen gibt, als $\mathbb{F}_q$; in einigen Büchern auch als GF(q), wobei GF für GALOIS *field* steht nach dem französischen Mathematiker EVARISTE GALOIS (1811–1832) und dem englischen Wort *field* für *Körper*.

Man kann zeigen, daß es genau dann einen solchen Körper gibt, wenn q eine Primzahlpotenz ist, und daß dieser Körper dann bis auf Isomorphie eindeutig bestimmt ist.

d) Körper von Zweierpotenzordnung

Uns interessiert vor allem der Fall, daß $q = 2^n$ eine Zweierpotenz ist. Die Addition von $\mathbb{F}_q$ ist dann die Vektoraddition in $\mathbb{F}_2^n$, und genau wie oben geht es darum, eine Multiplikation zu definieren.

Der einfachste Weg dorthin führt über Polynome: Wir identifizieren den ersten Vektor der Standardbasis mit der Eins von $\mathbb{F}_{2^n} = \mathbb{F}_2^n$, bezeichnen den zweiten als α und definieren die α-Potenzen bis zur $(n-1)$-ten als die weiteren Basisvektoren:

$$1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \alpha = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \alpha^2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \quad \ldots, \quad \alpha^{n-1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}.$$

Damit läßt sich jedes Element von $\mathbb{F}_{2^n}$ als Polynom

$$c_0 + c_1\alpha + c_2\alpha^2 + \cdots + c_{n-1}\alpha^{n-1}$$

schreiben mit $c_i \in \mathbb{F}_2$, und wir können Produkte via Poly-nommultiplikation definieren, sobald wir wissen, was die höheren Potenzen von α sind.

Tatsächlich reicht es bereits, wenn wir nur die Potenz α^n kennen: Diese muß in der Form

$$\alpha^n = p_0 + p_1\alpha + p_2\alpha^2 + \cdots + p_{n-1}\alpha^{n-1}$$

mit $p_i \in \mathbb{F}_2$ darstellbar sein, und sobald wir die Koeffizienten p_i kennen, können wir rekursiv auch alle weiteren α-Potenzen ausrechnen: Beispielsweise ist

$$\begin{aligned}
\alpha^{n+1} = \alpha \cdot \alpha^n &= p_0\alpha + p_1\alpha^2 + p_2\alpha^3 + \cdots + p_{n-2}\alpha^{n-2} + p_{n-1}\alpha^n \\
&= p_0\alpha + p_1\alpha^2 + p_2\alpha^3 + \cdots + p_{n-2}\alpha^{n-1} \\
&\qquad + p_{n-1}\left(p_0 + p_1\alpha + p_2\alpha^2 + \cdots + p_{n-1}\alpha^{n-1}\right) \\
&= p_{n-1}p_0 + (p_{n-1}p_1 + p_2)\alpha + (p_{n-1}p_2 + p_3)\alpha^2 + \cdots \\
&\qquad + (p_{n-1}p_{n-2} + p_{n-1})\alpha^{n-2} + (p_{n-1}^2 + p_n)\alpha^{n-1},
\end{aligned}$$

und entsprechend geht es weiter für die höheren Potenzen.

Wie wir schon beim Körper mit vier Elementen gesehen haben, können wir die Koeffizienten p_i nicht beliebig aus $\mathbb{F}_2$ wählen; nur in einem Fall ergab sich dort wirklich ein Körper.

Um zu sehen, welche Bedingungen wir an die p_i stellen müssen, nehmen wir an, wir hätten bereits Koeffizienten gefunden, für die sich ein Körper $\mathbb{F}_{2^n}$ ergibt, und untersuchen, was wir dann über die p_i aussagen können.

Wir können die Gleichung

$$\alpha^n = p_0 + p_1\alpha + p_2\alpha^2 + \cdots + p_{n-1}\alpha^{n-1}$$

auch so auffassen, daß α eine Nullstelle des Polynoms

$$\begin{aligned} P(x) &= x^n - p_0 - p_1 x - p_2 x^2 - \cdots - p_{n-1}x^{n-1} \\ &= x^n + p_0 + p_1 x + p_2 x^2 + \cdots + p_{n-1}x^{n-1} \end{aligned}$$

im Körper $\mathbb{F}_{2^n}$ sein soll. (Das zweite Gleichheitszeichen kommt daher, daß es beim Rechnen im Körper $\mathbb{F}_2$ und in den Vektorräumen $\mathbb{F}_2^n$ keinen Unterschied gibt zwischen *plus* und *minus:* Für jeden Vektor $\vec{v} \in \mathbb{F}_2^n$ ist $\vec{v} + \vec{v} = \vec{0}$.)

Wenn wir nun ein Element von $\mathbb{F}_{2^n}$ als Polynom in α schreiben, ist diese Darstellung offensichtlich nicht eindeutig, denn beispielsweise ist

$$f(\alpha) = f(\alpha) + P(\alpha) = (f + P)(\alpha)$$

und allgemeiner gilt sogar für *jedes* Polynom g mit Koeffizienten in $\mathbb{F}_2$, daß

$$(f + g \cdot P)(\alpha) = f(\alpha) + g(\alpha) \cdot P(\alpha) = 0$$

ist. Offensichtlich ist $f(\alpha) = h(\alpha)$, wann immer das Polynom $f - h$ durch P teilbar ist.

Dies liefert einen neuen und schnelleren Zugang zur Multiplikation in $\mathbb{F}_{2^n}$: Um das Produkt zweier Elemente $f(\alpha)$ und $g(\alpha)$ auszurechnen, berechnen wir das Produktpolynom $f \cdot g$ und dividieren es mit Rest durch P, d.h.

$$(f \cdot g) : P = q \quad \text{Rest } h \qquad \text{oder} \qquad f \cdot g = q \cdot P + r\,.$$

Dann ist

$$(f \cdot g)(\alpha) = r(\alpha)\,,$$

und da r ein Polynom vom Grad höchstens $n - 1$ ist, kann $r(\alpha)$ direkt mit einem Vektor aus $\mathbb{F}_2^n$ identifiziert werden.

Rechnen wir etwa im Fall $n = 3$ mit dem Polynom

$$P = x^3 + x + 1\,,$$

so wird das Produkt der Vektoren

$$\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

folgendermaßen bestimmt: Die beiden Vektoren lassen sich als Linearkombination der Potenzen von α schreiben als

$$1 + 0 \cdot \alpha + 1 \cdot \alpha^2 = 1 + \alpha^2 \quad \text{und} \quad 0 + 1 \cdot \alpha + 1 \cdot \alpha^2 = \alpha + \alpha^2\,;$$

das Produkt der beiden zugehörigen Polynome

$$f = 1 + x^2 \quad \text{und} \quad g = x + x^2 \quad \text{ist} \quad x + x^2 + x^3 + x^4\,.$$

Division durch P ergibt

$$(x^4 + x^3 + x^2 + x) : (x^3 + x + 1) = x + 1 \quad \text{Rest } x + 1\,,$$

d.h.

$$(1 + \alpha^2)(\alpha + \alpha^2) = 1 + \alpha$$

oder

$$\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}\,.$$

Falls man das Polynom P als Produkt zweier Polynome f und g schreiben kann, die beide positiven Grad haben, haben beide insbesondere auch höchstens Grad $n - 1$, definieren also nichtverschwindende Elemente $f(\alpha)$ und $g(\alpha)$ aus $\mathbb{F}_{2^n}$. Deren Produkt ist aber $P(\alpha) = 0$, was in einem Körper natürlich nicht vorkommen darf. Damit haben wir eine erste Bedingung an P gefunden: P muß *irreduzibel* sein im Sinne der folgenden Definition:

Definition: Ein nichtkonstantes Polynom $P \in k[x]$ mit Koeffizienten aus einem Körper k heißt *reduzibel*, wenn es zwei nichtkonstante Polynome $f, g \in k[x]$ gibt, so daß $P = f \cdot g$ ist. Andernfalls heißt P *irreduzibel* (über k).

(Der Zusatz *über* k ist notwendig: Beispielsweise ist $x^2 + 1$ irreduzibel über $\mathbb{R}$, aber reduzibel über $\mathbb{C}$, denn dort ist $(x^2 + 1) = (x + i)(x - i)$. Da meist klar ist, über welchem Körper man arbeitet, wird der Zusatz aber oft weggelassen: Bei uns etwa geht es im Augenblick ausschließlich um Polynome über $\mathbb{F}_2$, so daß dieser Körper nicht ständig erwähnt werden muß.)

Wenn wir uns noch einmal die Konstruktion des Körpers mit vier Elementen anschauen, sehen wir, daß Irreduzibilität zumindest dort auch reicht: Von den vier Polynomen zweiten Grades über $\mathbb{F}_2$ ist genau eines irreduzibel, nämlich das, mit dem wir den Körper $\mathbb{F}_4$ definiert haben:

Ansatz für α^2	Polynom	Problem
$\alpha^2 = 0$	$f = x^2 = x \cdot x$	$\alpha \cdot \alpha = 0$
$\alpha^2 = 1$	$f = x^2 + 1 = (x + 1) \cdot (x + 1)$	$(\alpha + 1)(\alpha + 1) = 0$
$\alpha^2 = \alpha$	$f = x^2 + x = x(x + 1)$	$\alpha \cdot (\alpha + 1) = 0$
$\alpha^2 = \alpha + 1$	$f = x^2 + x + 1$	*keine Probleme*

Tatsächlich reicht die Irreduzibilität von P *immer* zur Definition eines Körpers; damit wir das zeigen können, müssen wir uns aber zunächst den (wohl zumindest teilweise schon vertrauten) EUKLIDischen Algorithmus etwas genauer anschauen.

e) Der Euklidische Algorithmus

Beginnen wir mit dem einfachsten Fall, für den der Algorithmus schon im zehnten Buch von EUKLIDs Elementen zu finden ist: Wir suchen den größten gemeinsamen Teiler zweier natürlicher Zahlen a und b, d.h. die größte natürliche Zahl d, die sowohl a als auch b teilt. Wir schreiben kurz

$$d = \mathrm{ggT}(a, b)\,.$$

Grundidee des EUKLIDischen Algorithmus ist die Anwendung der Division mit Rest: Für je zwei natürliche Zahlen x und y gibt es nichtnegative ganze Zahlen q und r, so daß

$$x = qy + r \quad \text{und} \quad 0 \le r < y$$

ist. Alsdann ist

$$\mathrm{ggT}(x, y) = \mathrm{ggT}(y, r)\,,$$

denn wegen der beiden Gleichungen

$$x = qy + r \quad \text{und} \quad r = x - qy$$

teilt jeder gemeinsame Teiler von x und y auch r, und jeder gemeinsame Teiler von y und r teilt auch x.

Der EUKLIDische Algorithmus nutzt dies aus, um die Zahlen, deren ggT bestimmt werden muß, sukzessive zu verkleinern, bis der ggT zweier Zahlen berechnet werden muß, von denen die eine Teiler der anderen ist; in diesem Fall ist natürlich die kleinere der beiden Zahlen gleich dem ggT.

Formal sieht der EUKLIDische Algorithmus zur Berechnung des ggT zweier natürlicher Zahlen a und b folgendermaßen aus:

Schritt 0: Setze $r_0 = a$ und $r_1 = b$

Schritt i, $i \ge 1$**:** Falls $r_i = 0$ ist, endet der Algorithmus mit dem Ergebnis $\mathrm{ggT}(a, b) = r_{i-1}$; andernfalls dividiere man r_{i-1} mit Rest durch r_i und bezeichne den Divisionsrest mit r_{i+1}.

(Bei einer tatsächlichen Implementierung bieten sich natürlich einige offensichtliche Optimierungen an.)

Der Algorithmus muß nach endlich vielen Schritten enden, denn bei der Division mit Rest ist stets $0 \le r_{i+1} < r_i$, so daß r_i mit jedem Schritt kleiner wird, was bei natürlichen Zahlen nicht unbegrenzt möglich ist. Da außerdem in jedem Schritt

$$\mathrm{ggT}(r_{i-1}, r_i) = \mathrm{ggT}(r_i, r_{i+1})$$

ist und im letzten Schritt, wenn r_{i-1} den vorigen Wert r_{i-2} teilt,

$$\mathrm{ggT}(r_{i-1}, r_{i-2}) = r_{i-1}$$

ist, folgt induktiv

$$\text{ggT}(a,b) = r_{i-1}\,,$$

so daß der Algorithmus das richtige Ergebnis liefert.

In dieser Form reicht der EUKLIDische Algorithmus für uns noch nicht aus; wir werden im folgenden oft den ggT nicht nur berechnen, sondern zusätzlich auch noch als ganzzahlige Linearkombination der Ausgangsdaten darstellen wollen. Daß dies tatsächlich möglich ist, zeigt der erweiterte EUKLIDische Algorithmus, der diese Darstellung auch explizit liefert:

Ausgangspunkt ist auch hier wieder die Division mit Rest; die zugehörige Gleichung

$$r_{i-1} = q_i r_i + r_{i+1}$$

läßt sich umschreiben als

$$r_{i+1} = -q_i r_i + r_{i-1}\,,$$

so daß r_{i+1} eine ganzzahlige Linearkombination von r_i und r_{i-1} ist. Da entsprechend auch r_i Linearkombination von r_{i-1} und r_{i-2} ist, folgt induktiv, daß der ggT von a und b als ganzzahlige Linearkombination von $r_0 = a$ und $r_1 = b$ dargestellt werden kann.

Algorithmisch sieht dies folgendermaßen aus:

Schritt 0: Setze $r_0 = a$, $r_1 = b$, $\alpha_0 = \beta_1 = 1$ und $\alpha_1 = \beta_0 = 0$. Mit $i = 1$ ist dann

$$r_{i-1} = \alpha_{i-1}a + \beta_{i-1}b \quad \text{und} \quad r_i = \alpha_i a + \beta_i b\,.$$

Diese Relationen werden in jedem der folgenden Schritte erhalten:

Schritt i, $i \geq 1$**:** Falls $r_i = 0$ ist, endet der Algorithmus mit

$$\text{ggT}(a,b) = r_{i-1} = \alpha_{i-1}a + \beta_{i-1}b\,.$$

Andernfalls dividiere man r_{i-1} mit Rest durch r_i mit dem Ergebnis

$$r_{i-1} = q_i r_i + r_{i+1}\,.$$

Dann ist

$$\begin{aligned} r_{i+1} = -q_i r_i + r_{i-1} &= -q_i(\alpha_i a + \beta_i b) + (\alpha_{i-1}a + \beta_{i-1}b) \\ &= (\alpha_{i-1} - q_i\alpha_i)a + (\beta_{i-1} - q_i\beta_i)b\,; \end{aligned}$$

man setze also

$$\alpha_{i+1} = \alpha_{i-1} - q_i\alpha_i \quad \text{und} \quad \beta_{i+1} = \beta_{i-1} - q_i\beta_i\,.$$

Genau wie oben folgt, daß der Algorithmus für alle natürlichen Zahlen a und b endet und daß am Ende der richtige ggT berechnet wird; außerdem sind die α_i und β_i so definiert, daß in jedem Schritt $r_i = \alpha_i a + \beta_i b$ ist, insbesondere ist also im letzten Schritt der ggT als Linearkombination der Ausgangszahlen dargestellt.

Als Beispiel wollen wir den ggT von 200 und 148 als Linearkombination darstellen. Im nullten Schritt haben wir 200 und 148 als die trivialen Linearkombinationen

$$200 = 1 \cdot 200 + 0 \cdot 148 \quad \text{und} \quad 148 = 0 \cdot 200 + 1 \cdot 148\,.$$

Im ersten Schritt dividieren wir, da 148 nicht verschwindet, 200 mit Rest durch 148:

$$200 = 1 \cdot 148 + 52 \Longrightarrow 52 = 1 \cdot 200 - 1 \cdot 148$$

Da auch $52 \neq 0$, dividieren wir im zweiten Schritt 148 durch 52 mit Ergebnis $148 = 2 \cdot 52 + 44$, d.h.

$$44 = 148 - 2 \cdot (1 \cdot 200 - 1 \cdot 148) = 3 \cdot 148 - 2 \cdot 200$$

Auch $44 \neq 0$, wir dividieren also weiter: $52 = 1 \cdot 44 + 8$ und

$$\begin{aligned} 8 = 52 - 44 &= (1 \cdot 200 - 1 \cdot 148) - (3 \cdot 148 - 2 \cdot 200) \\ &= 3 \cdot 200 - 4 \cdot 148\,. \end{aligned}$$

Im nächsten Schritt erhalten wir $44 = 5 \cdot 8 + 4$ und

$$\begin{aligned} 4 = 44 - 5 \cdot 8 &= (3 \cdot 148 - 2 \cdot 200) - 5 \cdot (3 \cdot 200 - 4 \cdot 148) \\ &= 23 \cdot 148 - 17 \cdot 200\,. \end{aligned}$$

Bei der Division von acht durch vier schließlich erhalten wir Divisionsrest null; damit ist vier der ggT von 148 und 200 und kann in der angegebenen Weise linear kombiniert werden.

Als eine kleine Anwendung des erweiterten EUKLIDischen Algorithmus wollen wir zeigen, daß die ganzen Zahlen modulo einer Primzahl p einen Körper bilden:

Ist p eine Primzahl und $0 < a < p$ eine natürliche Zahl, so ist der ggT von a und p natürlich gleich eins. Also gibt es ganze Zahlen α, β, so daß

$$\alpha a + \beta p = 1 \quad \text{oder} \quad \alpha a - 1 = \beta p$$

ist. Damit ist $\alpha a - 1$ durch p teilbar oder, anders ausgedrückt

$$\alpha a \equiv 1 \bmod p\,.$$

Somit ist α also ein multiplikatives Inverse zu a.

Zur Inversion von 20 im Körper $\mathbb{F}_{1009}$ etwa berechnen wir also zunächst den ggT:

$$1009 : 20 = 50 \text{ Rest } 9 \quad \text{und} \quad 9 = 1 \cdot 1009 - 50 \cdot 20$$
$$20 : 9 = 2 \text{ Rest } 2 \quad \text{und} \quad 2 = 20 - 2 \cdot 2 = -2 \cdot 1009 + 101 \cdot 20$$
$$9 : 2 = 4 \text{ Rest } 1 \quad \text{und} \quad 1 = 9 - 4 \cdot 2 = 9 \cdot 1009 - 454 \cdot 20$$

Also ist $(-454) \cdot 20 \equiv 1 \bmod 1003$; das Inverse von 20 in $\mathbb{F}_{1009}$ ist also -454 oder, besser ausgedrückt, $1009 - 454 = 555$. In der Tat ist

$$555 \cdot 20 = 11100 = 11 * 1009 + 1 \equiv 1 \bmod 1009\,.$$

Wenn wir auf der Menge $\mathbb{F}_p \underset{\text{def}}{=} \{0, 1, \ldots, p-1\}$ eine Addition und Multiplikation einführen durch die Vorschriften

$$a \oplus b \underset{\text{def}}{=} (a + b) \bmod p \quad \text{und} \quad a \odot b \underset{\text{def}}{=} ab \bmod p$$

definieren, ist klar, daß – genau wie in den ganzen Zahlen – das Kommutativ- und das Assoziativgesetz sowohl für die $\oplus$ als auch für $\odot$ gilt, und auch das Distributivgesetz folgt sofort aus dem für $\mathbb{Z}$. Bezüglich $\oplus$ ist die Null neutrales Element und $p - a$ invers zu a; bezüglich $\odot$ ist die Eins neutrales Element, und wie wir gerade gesehen haben, gibt es zu jedem $a \not\equiv 0$ ein Inverses. Somit ist $\mathbb{F}_p$ ein Körper.

Das Rechnen in diesem Körper ist einfach: Die Addition kann auf die gewöhnliche Addition in $\mathbb{Z}$ zurückgeführt werden; da $a + b$ für $a, b \in \mathbb{F}_p$ zwischen Null und $2p - 2$ liegt, ist

$$a \oplus b = \begin{cases} a + b & \text{falls } a + b < p \\ a + b - p & \text{sonst} \end{cases}\,.$$

Bei der Multiplikation ist die Situation nicht ganz so einfach; hier braucht man eine Division mit Rest, um $a \oplus b$ zu berechnen.

Das additive Inverse ist, wie bereits erwähnt, einfach $p - a$; die Berechnung des multiplikativen Inversen dagegen erfordert einen erweiterten EUKLIDischen Algorithmus und ist damit die rechenaufwendigste Operation in $\mathbb{F}_p$. Wenn keine Verwechslungsgefahr mit ganzen Zahlen besteht, bezeichnet man die Rechenoperationen in $\mathbb{F}_p$ meist einfach mit + und · anstelle von $\oplus$ und $\odot$.

f) Der Euklidische Algorithmus für Polynome

Nun sei k ein Körper, z.B. der Körper $\mathbb{F}_2$ mit zwei Elementen; außerdem seien

$$A = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$$

und

$$B = x^m + b_{m-1}x^{m-1} + \ldots + b_1 x + b_0$$

Polynome mit Koeffizienten a_i, b_i aus k; wir bezeichnen

$$n = \deg A \quad \text{und} \quad m = \deg B$$

als die Grade von A und B.

Dann läßt sich das Polynom A mit Rest durch B dividieren, d.h. man kann Polynome Q, R bestimmen, für die

$$A = QB + R \quad \text{ist mit} \quad \deg R < \deg B\,.$$

Mit dieser Division lassen sich sowohl der gewöhnliche als auch der erweiterte EUKLIDische Algorithmus sofort verallgemeinern auf Polynome; da der Grad von R kleiner ist als der von B und Grade als nichtnegative ganze Zahlen nicht unbegrenzt kleiner werden können, folgt daß der Algorithmus auch für Polynome stets nach endlich vielen Schritten endet.

Das Ergebnis kann allerdings in manchen Fällen unerwartet ausfallen: Betrachten wir etwa über dem Körper $\mathbb{Q}$ der rationalen Zahlen die beiden Polynome

$$P = X^8 + X^6 - 3X^4 - 3X^3 + 8X^2 + 2X - 5$$

und

$$Q = 3X^6 + 5X^4 - 4X^2 - 9X + 21\,.$$

Division von P durch Q führt auf den Quotienten $X^2/3 - 2/9$ und Divisionsrest

$$R_2 = -\frac{5}{9}X^4 + \frac{1}{9}X^2 - \frac{1}{3}\,.$$

Division von Q durch R_2 ergibt

$$R_3 = -\frac{117}{25}X^2 - 9X + \frac{441}{25}\,,$$

bei der Division von R_2 durch R_3 bleibt Rest

$$R_4 = \frac{233150}{6591}X - \frac{102500}{2197}\,,$$

und bei der letzten Divison verbleibt als Rest der ggT

$$R_5 = \frac{1288744821}{543589225}\,.$$

Da beide Ausgangspolynome ganzzahlige Koeffizienten haben, erscheint ein ggT mit einem so großen Nenner seltsam. In der Tat ist jedes Polynom durch jede von Null verschiedene Konstante teilbar; ist also ein Polynom P Teiler eines Polynoms Q, so ist auch jedes von Null verschiedene skalare Vielfache von P Teiler von Q. Somit können wir hier nicht sinnvoll von *dem* größten gemeinsamen Teiler zweier Polynome reden.

Wir haben bislang noch nicht definiert, wann ein Polynom *größer* sein soll als ein anderes: Bei zwei natürlichen Zahlen ist klar, welche größer ist, aber schon bei reellen Polynomen ist alles andere als klar, ob etwa $x + 2$ größer sein soll als $2x + 1$ oder umgekehrt. Wir werden dieses Problem ignorieren und einfach sagen, P sei *ein* größter gemeinsamer Teiler von A und B, wenn P ein gemeinsamer Teiler ist und jeder andere gemeinsamen Teiler ein Teiler von P ist.

Der größte gemeinsame Teiler, den uns der EUKLIDISCHE Algorithmus für Polynome liefert, hat diese Eigenschaft, denn da dieser ggT als Linearkombination von A und B geschrieben werden kann, muß jedes Polynom, das sowohl A als auch B teilt, auch den ggT teilen.

Problematischer ist, daß es viele solche größten gemeinsamen Teiler geben kann: Zumindest jedes von Null verschiedene skalare Vielfache eines ggT ist selbst einer. Zum Glück ist das aber auch schon alles, was passieren kann: Sind nämlich P und Q zwei größte gemeinsame Teiler von A und B, so muß nach Definition P ein Teiler von Q sein und umgekehrt. Da der Grad eines Teilers stets kleiner oder gleich dem des Polynoms ist, haben die beiden also insbesondere denselben Grad, und

ihr Quotient, egal in welcher Reihenfolge, hat Grad null und ist somit eine Konstante.

Der größte gemeinsame Teiler zweier Polynome über einem Körper ist also eindeutig bis auf Multiplikation mit einer nichtverschwindenen Konstanten; diese Konstante kann nach Belieben gewählt werden und wird meist so gewählt, daß das Ergebnis in irgendeinem Sinne einfach wird.

Auf das obige Beispiel angewendet heißt das, daß mit

$$R_5 = \frac{1288744821}{543589225}$$

auch eins ein ggT von A und B ist und man daher im allgemeinen sagen würde, „der“ ggT von A und B sei eins. Es ist ein wohlbekanntes (und umgehbares) Problem der Computeralgebra, daß der EUKLIDische Algorithmus diese einfache Lösung in einer so komplizierten Form liefert; da uns vor allem Polynome über endlichen Körpern interessieren, braucht uns das nicht weiter zu kümmern.

Kehren wir zurück zum Ausgangsproblem: Wir wollen den Vektorraum $\mathbb{F}_2^n$ zu einem Körper machen. Da es in $\mathbb{F}_2$ genau ein von null verschiedenes Element gibt, spielt die obige Diskussion hier keine Rolle: Für Polynome über $\mathbb{F}_2$ existiert *der* ggT. Trotzdem war diese Diskussion nicht umsonst, denn erstens werden wir im nächsten Kapitel im Zusammenhang mit der Integration rationaler Funktionen den EUKLIDischen Algorithmus auch auf reelle Polynome anwenden, und zweitens sei zumindest kurz erwähnt, daß die folgende Konstruktion auch für eine beliebige Primzahl p Körper mit p^n Elementen liefert. Sie werden allerdings in der Informationstechnik nur selten benutzt: Dort interessieren praktisch nur die Körper $\mathbb{F}_{2^n}$ und die Körper $\mathbb{F}_p$, denn das Rechnen in $\mathbb{F}_{p^n}$ ist umständlicher als das Rechnen in einem Körper $\mathbb{F}_q$ mit einer Primzahl q der Größenordnung p^n und bietet für $p \neq 2$ keinerlei Vorteile. Lediglich für $p = 2$, wo die Vektorraumstruktur von $\mathbb{F}_2^N$ so gut an die heutige Computer-Hardware angepaßt wird, bieten Körper von Zweierpotenzordnung oft (wenn auch keinesfalls immer!) Vorteile über Körper von Primzahlordnung.

In Abschnitt *e)* hatten wir die ganzen Zahlen modulo p zu einem Körper gemacht; der einzige nichttriviale Schritt dabei war die Existenz des multiplikativen Inversen, die wir aus der linearen Kombinierbarkeit des

ggT folgerten und daraus, daß der ggT einer Zahl mit einer Primzahl gleich eins ist, falls die Zahl kein Vielfaches der Primzahl ist.

Genauso wollen wir jetzt Körper definieren, indem wir Polynome über einem festen Körper k modulo einem vorgegebenen Polynom P betrachten: Für ein beliebiges Polynom A über k ist A mod P gleich dem Rest bei der Division von A durch P.

Falls A kleineren Grad als P hat, ist natürlich einfach A mod $P = A$; zum konkreten Rechnen können wir daher ausgehen vom Vektorraum V aller Polynome vom Grad höchstens d, wobei $d+1$ der Grad von P ist. Die Addition ist die gewöhnliche Addition von Polynomen, das Nullpolynom ist Neutralelement, und $-A$ ist invers zu A.

Das Produkt AB zweier Polynome $A, B \in V$ kann größeren Grad als d haben; wir setzen daher

$$A \odot B = AB \bmod P\,;$$

dies ist ein Polynom vom Grad höchstens d, und es ist klar, daß die so definierte Multiplikation kommutativ und assoziativ ist und das Distributivgesetz erfüllt. Das konstante Polynom 1 ist Neutralelement auch bezüglich dieser Multiplikation.

Ein inverses Polynom zu A ist ein Polynom B, für das $A \odot B = 1$ ist, d.h.

$$AB = 1 + CP \quad \text{oder} \quad AB + CP = 1$$

für ein geeignetes Polynom C. Zu vorgegebenen Polynomen A und P gibt es solche Polynome B und C genau dann, wenn der ggT von A und P gleich eins ist; alsdann lassen sich B und C nach dem EUKLIDischen Algorithmus berechnen.

Wenn wir möchten, daß jedes Polynom A, dessen Grad kleiner als deg P ist, ein Inverses hat, müssen wir sicherstellen, daß A und P immer teilerfremd sind; dies ist offensichtlich genau dann der Fall, wenn P keinen nichttrivialen Teiler hat, also irreduzibel ist.

Falls es ein irreduzibles Polynom P vom Grad n mit Koeffizienten aus k gibt, läßt sich der Vektorraum k^n also zu einem Körper machen, indem

wir ein n-tupel $(a_0, \ldots, a_{n-1})$ mit dem Polynom

$$a_{n-1}X^{n-1} + a_{n-2}X^{n-2} + \cdots + a_1X + a_0$$

identifizieren und die Multiplikation als Multiplikation von Polynomen modulo P erklären.

Betrachten wir noch einmal das altbekannte Beispiel der komplexen Zahlen: Für $n = 2$ gibt es irreduzible Polynome vom Grad n über $\mathbb{R}$, beispielsweise das Polynom $P = X^2 + 1$. Da

$$\begin{aligned}(a_1X + a_0)(b_1X + b_0) &= a_1b_1X^2 + (a_0b_1 + a_1b_0)X + a_0b_0 \\ &\equiv (a_0b_1 + a_1b_0)X + (a_0b_0 - a_1b_1) \bmod X^2 + 1\end{aligned}$$

ist, folgt $(a_0, a_1) \odot (b_0, b_1) = (a_0b_0 - a_1b_1, a_0b_1 + a_1b_0)$, wir erhalten also den Körper der komplexen Zahlen. Weitere Beispiele über $\mathbb{R}$ gibt es nicht, denn ein irreduzibles reelles Polynom muß entweder Grad eins oder Grad zwei haben, und da jedes irreduzible quadratische Polynom zwei konjugiert komplexe Nullstellen hat, entstehen dabei immer die komplexen Zahlen – lediglich die Basis über $\mathbb{R}$ ändert sich.

Über endlichen Körpern ist die Situation etwas komplizierter: Hier wissen wir nicht einmal, für welche n es überhaupt ein irreduzibles Polynom vom Grad n gibt. Tatsächlich gibt es sogar ziemlich viele solche Polynome; die Tabelle zeigt deren Anzahl über $\mathbb{F}_2$ für $n \leq 16$.

Mit etwas mehr Algebra zeigt man leicht, daß es über jedem endlichen Körper irreduzible Polynome jedes beliebigen (positiven) Grads gibt und daß zwei solche Polynome *im wesentlichen* (d.h. bis auf Isomorphie) zum selben Körper führen. Wir wollen uns darauf beschränken, für den uns hauptsächlich interessierenden Fall des Körpers $\mathbb{F}_{256}$ konkrete Polynome zu betrachten: Konkretes Rechnen mit Bitfolgen setzt schließlich ohnehin immer ein konkretes Polynom voraus.

g) Der Körper mit 256 Elementen und CD-Fehlerkorrektur

Zunächst müssen wir diesen Körper definieren, d.h. ein irreduzibles Polynom vom Grad acht über $\mathbb{F}_2$ finden. Wie die obige Tabelle zeigt, haben wir dazu dreißig Möglichkeiten. Diese führen zwar, abstrak betrachtet, alle auf denselben Körper, aber das praktische Rechnen in

N	Polynome vom Grad N	*davon irreduzibel*	*in Prozent*
2	4	1	25.0%
3	8	2	25.0%
4	16	3	18.8%
5	32	6	18.8%
6	64	9	14.1%
7	128	18	14.1%
8	256	30	11.7%
9	512	56	10.9%
10	1024	99	9.7%
11	2048	186	9.1%
12	4096	335	8.2%
13	8192	630	7.7%
14	16384	1161	7.1%
15	32768	2182	6.7%
16	65536	4080	6.2%

diesem Körper hängt natürlich stark von der Wahl des Polynoms ab. Insbesondere wird die Geschwindigkeit umso höher, je weniger Terme das Polynom hat.

Dreizehn der dreißig Polynome bestehen aus sieben nichtverschwindenden Termen, die restlichen siebzehn aus fünf; Wir wählen natürlich eines der letzteren. Alle diese Polynome haben, wie jedes Polynom vom Grad acht über $\mathbb{F}_2$, den führenden Term X^8; danach folgen vier weitere Terme. Bei der Reduktion modulo einem solchen Polynom $P = X^8 + \mathit{Rest}$ benutzt man, daß dann

$$X^8 \equiv \mathit{Rest}, \qquad X^9 \equiv X \cdot \mathit{Rest}, \quad \ldots$$

ist; dies wird umso häufiger mehrfach angewandt werden müssen, je höheren Grad die Terme in *Rest* haben. Am effizientesten kann man also rechnen, wenn das Polynom *Rest* den kleinstmöglichen Grad hat, und wenn zudem auch noch die hinteren Terme von *Rest* möglichst kleinen Grad haben. Inspektion der siebzehn Polynome mit fünf Termen zeigt,

daß das Polynom

$$x^8 + x^4 + x^3 + x + 1$$

in dieser Hinsicht optimal ist; es wird beim *Advanced Encryption Standard* AES verwendet, den wir im nächsten Abschnitt kurz betrachten werden. Für die Fehlerkorrektur auf CDs verwendet man das leicht verschiedene Polynom $x^8 + x^4 + x^3 + x^2 + 1$.

Überlegen wir uns zunächst, worum es bei dieser Fehlerkorrektur geht:

Bei der Fertigung einer CD läßt sich die Fehlerwahrscheinlichkeit pro Bit auf etwa eins zu einer Million herunterdrücken; da aber bei einer Audio-CD rund vier Millionen Bit pro Sekunde verarbeitet werden, treten trotzdem jede Menge Fehler auf, die teilweise verheerende Folgen haben können: Falls beispielsweise das Wort 0000 0000 0001 0101 versehentlich als 1000 0000 0100 0101 interpretiert wird, wird aus einem Pianissimo ein ohrenbetäubender Knacklaut von ca. 90 dB.

Nun sollten allerdings nicht nur Fertigungsfehler korrigiert werden, sondern zumindest in gewissem Umfang auch Fehler durch Fingerabdrücke, Staubkörner *usw.*

Da die Spur bei einer CD etwa einen halben Mikrometer breit ist und die einzelnen Pits zwischen 0,833μ und 3,56μ lang sind, wohingegen ein Fingerabdruck bereits Linien mit einer Breite von 15μ erzeugt und eine beim Staubwischen übriggebliebene Baumwollfaser gar eine Breite von 150μ hat, ist klar, daß sich solche Fehler nicht im Bitbereich bewegen.

Dies wird auf einer CD zum einen dadurch berücksichtigt, daß man die Information nicht linear anordnet (die geraden Bytes werden gegenüber den ungeraden verzögert), zum anderen dadurch, daß man anstelle des Bits das Byte als grundlegende Einheit betrachtet, d.h. man arbeitet mit dem Körper $\mathbb{F}_{256}$.

Die Fehlerkorrektur arbeitet mit Prüfbytes, die (wie Paritätsbits) durch lineare Abbildungen definiert sind. Zu einem Vektor aus 24 Bytes werden in zwei Stufen insgesamt acht Prüfbytes berechnet, und zwar werden zunächst vier Prüfbytes angehängt derart, daß der entstehende Vektor

aus $\mathbb{F}_{256}^{28}$ im Kern der linearen Abbildung

$$\varphi\colon \mathbb{F}_{256}^{28} \to \mathbb{F}_{256}^{4};\quad \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{27} \\ x_{28} \end{pmatrix} \mapsto \begin{pmatrix} \sum_{i=1}^{28} x_i \\ \sum_{i=1}^{28} \alpha^{28-i} x_i \\ \sum_{i=1}^{28} \alpha^{2(28-i)} x_i \\ \sum_{i=1}^{28} \alpha^{3(28-i)} x_i \end{pmatrix}$$

liegt, danach werden vier weitere Bytes angehängt derart, daß der entstehende Vektor aus $\mathbb{F}_{256}^{32}$ im Kern der linearen Abbildung

$$\psi\colon \mathbb{F}_{256}^{32} \to \mathbb{F}_{256}^{4};\quad \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{31} \\ x_{32} \end{pmatrix} \mapsto \begin{pmatrix} \sum_{i=1}^{32} x_i \\ \sum_{i=1}^{32} \alpha^{32-i} x_i \\ \sum_{i=1}^{32} \alpha^{2(32-i)} x_i \\ \sum_{i=1}^{32} \alpha^{3(32-i)} x_i \end{pmatrix}$$

liegt. $\alpha \in \mathbb{F}_{256}$ bezeichnet dabei wie üblich jenes Element, für das die Eins zusammen mit α bis α^7 eine $\mathbb{F}_2$-Basis von $\mathbb{F}_{256}$ ist, und das Nullstelle des definierenden Polynoms ist.

Durch Kombination dieser Prüfbytes mit einer geschickten (nichtlinearen) Anordnung der Bytes auf der Spirale lassen sich selbst Fehler einer Länge von etwa 4 000 Bit beheben – teils durch echte Korrektur, teils durch bloße Fehlererkennung und Interpolation aus unverfälschten Daten. Versuche von Physikern der University of Maryland haben ergeben, daß eine CD eingebohrte Löcher mit einem Durchmesser von 0,8 mm problemlos verkraftet, und selbst ein Lochdurchmesser von 1,5 mm führt kaum zu Knackgeräuschen. Einzelheiten findet man unter `www.physics.umd.edu/deptinfo/facilities/lecdem/h4-67.htm` .

h) Der Körper mit 256 Elementen in der Kryptographie

Zwar lehnt es die Internationale Standardisierungsorganisation ISO ab, ein Kryptoverfahren zu standardisieren (Ein Grund dafür ist die dann befürchtete Bündelung krimineller Energie auf dieses Verfahren), aber das amerikanische Handelsministerium hat am 2. Januar 1997 die Suche nach einem Nachfolgealgorithmus für den nach heutigen Standards nicht mehr sicheren DES *(Data Encryption Standard)* international

ausgeschrieben. Federführend für die Auswahl war das *National Institute of Standards and Technology* (NIST) in Gaithersburg, Maryland, das am 2. Oktober 2000 den Algorithmus Rijndael der beiden flämischen Kryptologen JOAN DAEMEN und VINCENT RIJMEN auswählte. (Als Aussprachehilfe für Personen, die kein Niederländisch, Flämisch, Surinamer oder Afrikaans sprechen, geben diese folgende englische Approximationen des Wortes „Rijndael“: „Reign Dahl“, „Rain Doll“ und „Rhine Dahl“.) Es steht zu erwarten, daß Rijndael mittelfristig auch außerhalb der USA zu *dem* künftigen Standardverfahren in der Kryptographie wird.

Grundidee sind, wie bei allen Kryptoverfahren, die beiden SHANNONschen Forderungen der *Diffusion* und *Konfusion:* Ersteres bedeutet, daß sich schon die Änderung eines einzigen Klartextbits an vielen, möglichst weit entfernten Stellen bemerkbar machen muß, das zweite bedeutet in erster Linie eine hohe Nichtlinearität der Verschlüsselungsabbildung, so daß diese ohne Kenntnis des Schlüssels nicht invertiert werden kann.

Nichtlinearität erreicht Rijndael durch die Abbildung $\mathbb{F}_{256} \to \mathbb{F}_{256}$, die die Null auf sich selbst abbildet und jedes andere Element auf sein multiplikatives Inverses. Über mehrere Runden hinweg wird diese Abbildung auf Byte-Ebene immer wieder mit linearen Abbildungen und Vektoradditionen auf $\mathbb{F}_{256}^4$ und Shift-Operationen auf $\mathbb{F}_{256}^{16}$ oder noch größeren Vektorräumen kombiniert. Alle linearen Abbildungen sind $\mathbb{F}_{256}$-linear, was auf Bitebene noch einmal eines Konfusionseffekt hat. Die einzelnen Operationen hängen ab von einem Schlüsselvektor, der Element von $\mathbb{F}_{256}^{16}$, $\mathbb{F}_{256}^{24}$ oder $\mathbb{F}_{256}^{32}$ sein kann und somit 128, 192 oder 256 Bit lang ist.

i) Der diskrete Logarithmus

Ein Verfahren wie AES kann nur dann sicher angewendet werden, wenn Sender und Empfänger sich auf einen Schlüssel geeinigt haben; dessen Austausch durch sicheren Boten ist beispielsweise für Anwendungen der Kryptographie im Internet zu aufwendig um praktikabel zu sein.

Zum Glück gibt es seit etwa 25 Jahren auch Verfahren, mit denen ein

solcher Schlüssel über eine unsichere Leitung sicher vereinbart werden kann; Internetbrowser tun dies beispielsweise bei der sicheren Datenübertragung automatisch, ohne daß der Benutzer etwas merkt.

Das mathematische Verfahren, das meist dahinter steckt, hat zwar nichts mit Vektorräumen zu tun, dafür aber immerhin mit endlichen Körpern; es sei daher zum Abschluß dieses Paragraphen kurz erwähnt. Es beruht auf den sogenannten *diskreten Logarithmen.*

In $\mathbb{R}$ ist der Logarithmus zur Basis a die Umkehrfunktion der Funktion $x \mapsto a^x$; genauso definieren wir ihn auch für endliche Körper:

$$y = a^x \Longrightarrow x = \log_a y\,.$$

Trotz dieser formalen Übereinstimmung gibt es es allerdings große Unterschiede zwischen reellen Logarithmen und ihren Analoga in endlichen Körpern: Während reelle Logarithmen sanft ansteigende stetige Funktionen sind, die man leicht mit beliebig guter Genauigkeit annähern kann, sieht der diskrete Logarithmus typischerweise so aus, wie es in der Abbildung zu sehen ist. Auch ist im Reellen der Logarithmus zur Basis $a > 1$ für jede positve Zahl definiert; in endlichen Körpern ist es viel schwerer zu entscheiden, ob ein bestimmter Logarithmus existiert: Modulo sieben etwa sind $2, 4$ und 1 die einzigen Zweierpotenzen, so daß $3, 5$ und 6 keine Zweierlogarithmen haben. Ein Satz aus der Algebra besagt allerdings, daß es stets Elemente a gibt, für die a^x jeden Wert außer der Null annimmt, die sogenannten primitiven Wurzeln. In $\mathbb{F}_7$ wären dies etwa drei und fünf.

Die Berechnung der Potenzfunktion durch sukzessives Quadrieren und Multiplizieren ist auch in endlichen Körpern einfach, für ihre Umkehrfunktion, den diskreten Logarithmus gibt es aber derzeit nur deutlich schlechtere Verfahren. Die derzeit besten Verfahren zur Berechnung von diskreten Logarithmen in Körpern mit N Elementen erfordern etwa denselben Aufwand wie die Faktorisierung eines RSA-Moduls der Größenordnung N; diese Diskrepanz zwischen Potenfunktion und Logarithmen kann kryptologisch ausgenutzt werden.

Als Körper verwendet man entweder Körper von Zweipotenzordnung, da man in diesen gut rechnen kann, oder Körper von Primzahlordnung.

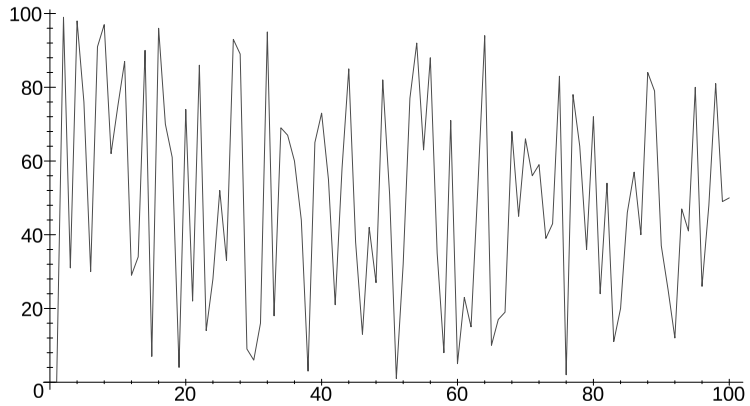

Abb. 11: Die Funktion $\log_{51} x$ in $\mathbb{F}_{101}$

Da es für viele interessante Körper von Zweipotenzordnung bereits Chips gibt, die dort diskrete Logarithmen berechnen, dürften Körper von Primzahlordnung bei ungefähr gleicher Elementanzahl wohl etwas sicherer sein: Es gibt einfach viel mehr Primzahlen als Zweierpotenzen, und jeder Fall erfordert einen neue Hardwareentwurf. Falls man die Primzahlen hinreichend häufig wechselt, dürfte sich dieser Aufwand für kaum einen Gegner lohnen.

Da Körper von Primzahlordnung auch einfacher sind als solche von Primzahlpotenzordnung, wollen wir uns hier auf die ersteren beschränken; die Übertragung des Algorithmus auf Körper von Zweipotenzordnung sollte dem Leser keine Schwierigkeiten machen.

Beim DIFFIE-HELLMANN-Verfahren, dem ältesten auf der Grundlage diskreter Logarithmen, geht es darum, wie zwei Teilnehmer, die weder über gemeinsame Schlüsselinformation noch über eine sichere Leitung verfügen, einen Schlüssel vereinbaren können.

Nach DIFFIE-HELLMANN einigen sie sich zunächst (über die unsichere Leitung) auf eine Primzahl p und eine natürliche Zahl a derart, daß die Potenzfunktion $x \mapsto a^x$ möglichst viele Werte annimmt.

Als nächstes wählt Teilnehmer A eine Zufallszahl $x < p$ und B ent-