

DHBW Mannheim
Studiengang Informatik

Kryptologie

Manuskript zur Vorlesung

Autor:
Reinhold Hübl, Fakultät für Technik, DHBW Mannheim
e-mail:reinhold.huebl@dhw-mannheim.de

Inhaltsverzeichnis

1. Überblick und historische Verfahren	7
1.1. Caesar–Verschlüsselung	8
1.2. Monoalphabetische Substitution	12
1.3. Vigenère–Verschlüsselung	16
2. Grundlagen der Kryptologie	26
2.1. Symmetrische Verschlüsselungsverfahren	26
2.2. Blockchiffrierung	27
2.2.1. ECB–Mode (Electronic Code Book–Mode)	29
2.2.2. CBC–Mode (Cipher Block Chaining–Mode)	31
2.2.3. CTR–Mode (CounTeR–Mode)	33
2.3. Stromchiffrierung	37
2.4. Asymmetrische Verschlüsselung	40
2.5. Kryptoanalyse	42
3. Data Encryption Standard DES	49
3.1. Feistel–Chiffrierung	49
3.2. Die <i>S</i> –Boxen	50
3.3. Der Schlüsselfahrplan und Vorbereitung der Runden	54
3.4. Rundenverarbeitung und Abschluss	61
3.5. Dechiffrierung von DES	67
3.6. Die Sicherheit von DES	71
4. Advanced Encryption Standard AES	74
4.1. Der Körper mit 256 Elementen	74
4.2. <i>S</i> –Boxen und Byte–Substitution in AES	78
4.3. Der AES–Schlüsselfahrplan	80
4.4. Die AES–Diffusionsoperationen	83
4.4.1. ShiftRow–Operations	83
4.4.2. MixColumns–Operations	84
4.5. Ablauf der AES–Verschlüsselung	85
4.6. Entschlüsselung von AES	86
5. Grundlagen der asymmetrischen Verschlüsselung	88

6. Asymmetrische Verfahren basierend auf dem Faktorisierungsproblem	93
6.1. Das RSA–Verfahren	93
6.2. Das Rabin–Verschlüsselungsverfahren	102
7. Asymmetrische Vefahren basierend auf dem diskreten Logarithmusproblem	106
7.1. Diffie–Hellman–Schlüsselaustausch DHKE	106
7.2. Das ElGamal–Verschlüsselungsverfahren	110
8. Digitale Signaturen	114
8.1. RSA–Signatur	115
8.2. ElGamal–Signatur	118
9. Hash–Funktionen	124
9.1. Probleme bei digitalen Signaturen	124
9.2. Hash–Funktionen und ihre Eigenschaften	125
9.3. Die Merkle–Damgård–Metakonstruktion	129
9.4. Secure Hash Algorithm 1 (SHA-1)	132
9.5. Die Schwamm–Konstruktion	135
9.6. Der digitale Signaturalgorithmus DSA	143
10. Message Authentication Codes	148
10.1. MAC–Systeme, die auf Hash–Funktionen basieren	150
10.2. Hash–Based Message Authentication Code HMAC	151
10.3. MAC mit Blockchiffren: CBC–MAC	153
10.4. Galois/Counter–Mode Message Authentication Code GMAC	154
11. Elliptische Kurven	157
11.1. Definition elliptischer Kurven	157
11.2. Elliptische Kurven und Geraden	165
12. Arithmetik elliptischer Kurven	175
12.1. Projektive elliptische Kruven	175
12.2. Addition auf elliptischen Kurven in der Charakteristik $p > 3$	176
13. Elliptische Kurven in der Charakteristik 2	184
13.1. Beschreibung elliptischer Kurven in der Charakteristik 2	184
13.2. Addition auf elliptischen Kurven in der Charakteristik 2	191

14. Elliptische Kryptosysteme	199
14.1. Vielfachenbildung durch iteriertes Verdoppeln	199
14.2. ECDH Diffie–Hellman–Schlüsselaustausch mit elliptischen Kurven	206
14.3. ElGamal–Verschlüsselung mit elliptischen Kurven	213
14.4. ECDSA - Digitale Signatur mit elliptischen Kurven	221
14.5. Unsichere elliptische Kurven	228
14.6. Empfohlene elliptische Kurven	229
15. Gitter–basierte Kryptosysteme	230
15.1. Gitter	230
15.2. Gitterprobleme	236
15.3. Gitterbasierte Verschlüsselung - Erster Versuch	240
15.4. Learning With Errors LWE	249
A. Rechnen mit Restklassen ganzer Zahlen	253
B. Endliche Körper – Primkörper	266
C. Endliche Körper – Erweiterungskörper	275
D. Quadratische Gleichungen in der Charakteristik 2	288
E. Primzahltest und die Suche nach großen Primzahlen	301
E.1. Der Sieb des Eratosthenes	301
E.2. Probiedivision	304
E.3. Der Fermat–Test	306
E.4. Der Miller–Rabin–Test	308
F. Faktorisierung großer Zahlen	314
F.1. Probiedivision	314
F.2. Fermat–Faktorisierung	315
F.3. Die $p - 1$ –Methode von Pollard	318
F.4. Das Quadratische Sieb	321
G. Das diskrete Logarithmus–Problem	329
G.1. Endliche zyklische Gruppen	330
G.2. Babystep–Giantstep–Algorithmus	334
G.3. Der Pollard– ρ –Algorithmus	336
G.4. Der Pohlig–Hellman–Algorithmus	339

G.5. Index–Calculus	344
H. Polynome und Kurven	348
H.1. Polynome in zwei Variablen	348
H.2. Ebene Kurven	350

Einführung

Die Kryptologie ist die Wissenschaft, die sich mit dem Ver- und Entschlüsseln von Nachrichten und Informationen und mit der Informationssicherheit beschäftigt, also damit, Geheimnisse auch geheim zu halten.

Geheimschriften zum Verschleiern von Informationen sind wohl so alt wie die Schriften selbst. Vermutlich wurden sie bereits im alten Ägypten benutzt. Gesichert ist die Verwendung im Griechenland der Antike (Skytale in Sparte, ca. 500 v. Chr.) und in Rom zur Zeit Caesars (Caesar-Chiffre).

Die Kryptologie besteht im wesentlichen aus den beiden Teilgebieten Kryptographie und Kryptoanalyse (oder Kryptanalyse).

Themen der modernen Kryptographie sind

- i) *Geheimhaltung*: eine Nachricht, die während der Übertragung von einer dritten Partei abgefangen wird, darf sich dieser inhaltlich nicht erschließen.
- ii) *Datenintegrität*: Der Empfänger einer Nachricht muss in der Lage sein, zu überprüfen, ob die Nachricht während der Übertragung (absichtlich oder unabsichtlich) verändert wurde.
- iii) *Authentifizierung*: Dem Empfänger der Nachricht muss es möglich sein, den Sender zu identifizieren. Niemand anders als der angegebene Sender darf in der Lage sein, die Nachricht in genau dieser Form zu versenden.
- iv) *Nachweisbarkeit*: Der Sender darf nicht in der Lage sein, später zu leugnen, dass er die Nachricht tatsächlich geschrieben bzw. gesendet hat.

Ein kryptographisches Protokoll ist ein Prozess bzw. eine Vorgehensweise, die mindestens eines dieser Sicherheitsziele adressiert. Es definiert eindeutig eine festgelegte Abfolge von Handlungen für den Sender und den Empfänger, durch die Form und Ablauf der Kommunikation bestimmt sind. Beispiele für Protokolle sind etwa TLS, SSL, SSH und viele mehr.

Die Kryptoanalyse (oder auch Kryptanalyse) hat das Ziel, Angriffe auf ein kryptographisches Protokoll zu finden und entweder zu beweisen, dass das Protokoll einem gewissen Sicherheitsstandard genügt, oder dies zu widerlegen. Angriffe auf Verschlüsselungsverfahren sind nämlich so alt wie die Verschlüsselungsverfahren selbst, denn immer, wenn zwei Parteien geheim miteinander kommunizieren wollen, gibt es auch eine dritte Partei, vor der diese Kommunikation geheim gehalten werden soll, die sich aber für diese

Kommunikation interessiert, und die alles unternehmen wird, um an die gesendete Information zu kommen. Kryptoanalyse ist dabei sowohl für den Angreifer interessant, um Wege zu finden, eine Verschlüsselung zu knacken, als auch für die Kommunikationspartner, um sich sicher zu sein, dass dies (mit großer Wahrscheinlichkeit) nicht möglich ist.

Begleitliteratur für diese Vorlesung:

- BEUTELSPACHER, A.: *Kryptologie: Eine Einführung in die Wissenschaft vom Verschlüsseln, Verbergen und Verheimlichen*. Springer Spektrum, Berlin Heidelberg 2015. ISBN: 978-3-658-05975-0.
- BUCHMANN, J.: *Einführung in die Kryptographie*. Springer Spektrum, Berlin Heidelberg 2016. ISBN: 978-3-642-39774-5.
- DELFS, H. UND H. KNEBL: *Introduction to Cryptography: Principles and Applications*. Springer, Berlin Heidelberg, 2015. ISBN: 978-3-662-47973-5.
- PAAR, C. UND J PELZL: *Kryptografie verständlich : Ein Lehrbuch für Studierende und Anwender*. Springer Vieweg, Berlin Heidelberg 2016. ISBN: 978-3-662-49297-0.
- SCHNEIER, B.: *Applied Cryptography: Protocols, Algorithms and Source Code in C*, Wiley, Indianapolis 2015. ISBN: 978-1-119-09672-6.

Bei der Behandlung elliptischer Kurven werden wir oft Bezug nehmen auf

- SILVERMAN, J.: *The Arithmetic of Elliptic Curves*. Springer Spektrum, Berlin Heidelberg 2016. ISBN 978-0-387-09493-9.

1. Überblick und historische Verfahren

Kryptologie beschäftigt sich mit Fragen zur Sicherheit der Datenübertragung in folgender Grundsituation:

Ein Sender A (*Alice*) möchte einem Empfänger B (*Bob*) eine Nachricht zukommen lassen und benutzt dazu einen angreifbaren Datenübertragungskanal (Brief, email, Telearbeit, Telefon). Ein Angreifer C (*Catherine*) will die Unsicherheit des Übertragungskanals ausnutzen und die Daten abfangen, um

- a) in Erfahrung zu bringen, was Alice an Bob gesendet hat (*passiver Angriff*).
- b) zu verändern, was Alice an Bob gesendet hat (*aktiver Angriff*).

Da Alice und Bob mit der Unsicherheit des Übertragungskanals leben müssen, können sie das Abfangen bzw. Manipulieren des Datenstroms per se nicht unterbinden. Daher muss es ihr Ziel sein, zu verhindern, dass Catherine die Daten korrekt interpretieren kann, bzw. sicherzustellen, dass Manipulationen am Datenstrom erkannt werden können. Beide Ziele können durch geeignete Verschlüsselungstechniken erreicht werden.

Ein Verschlüsselungsverfahren besteht im wesentlichen aus zwei Algorithmen, einem Verschlüsselungsalgorithmus e (der in der Regel von einem Schlüssel k abhängt, also $e = e_k$) und einem Entschlüsselungsalgorithmus d (der ebenfalls von k abhängig ist, also $d = d_k$), sodass gilt

$$d_k(e_k(m)) = m$$

für jeden Klartext (also jede zu übertragende Nachricht in „verständlicher“ Sprache) m .

Im Prinzip ist die Idee der Verschlüsselung und der Anwendung von Geheimschriften wohl (fast) so alt wie die Schrift selbst. Vermutlich hatten schon die Ägypter neben den offiziellen Hieroglyphen auch Geheimhieroglyphen. Gesichert ist auf jeden Fall die Verwendung von **Skytalen** in Sparta um 500 v. Chr.



Abbildung 1: Skytale, Quelle: www.wikipedia.de, CC BY-SA 3.0

Hierbei wird ein langer, schmaler Pergament- oder Lederstreifen wendelförmig auf einen Stab gewickelt. Die Nachricht wird längs des Stabes auf das Band geschrieben, und zwar so, dass auf eine Bandbreite genau ein Buchstabe kommt. Schließlich wird das Band wieder abgewickelt. Es entsteht ein langes Band mit einer scheinbar wahllos aufeinander folgenden Buchstabenreihe.

Der Empfänger besitzt einen Stab gleichen Durchmessers. Wickelt er das Band wieder so auf wie der Sender, so kann er also die übermittelte Nachricht korrekt lesen.

1.1. Caesar–Verschlüsselung

Die sogenannte **Caesar–Verschlüsselung** ist ein erstes mathematisch–algorithmisches Verfahren, dass von Gaius Julius Caesar (100 v. Chr. - 44 v. Chr.) in seiner militärischen Korrespondenz verwendet wurde.

Beim Caesar–Verfahren erfolgt die Verschlüsselung durch eine Verschiebung jedes Buchstabens des Alphabets um eine fest vorgegebene Anzahl von Stellen. Bei einer Verschiebung um sechs Stellen etwa wird aus dem *A* ein *G*, aus dem *B* ein *H*, . . . , aus dem *T* ein *Z*, dem *U* ein *A*, dem *V* ein *B* usw.

Beispiel 1.1. Bei einer Verschiebung um 6 Buchstaben wird aus der Anrede AVE CAESAR der verschlüsselte Text GBK IGKYGX.

Beispiel 1.2. Verschlüsseln wir den Text

HEUTE UM MITTERNACHT AN DER GEWOHNTEN STELLE
mit einem Caesar–Verfahren, dass jeden Buchstaben um 9 Stellen verschiebt, so erhalten
wir den verschlüsselten Text

QNDCN DV VRCCNAWJLQC JW MNA PNFXQWCNW BCNUUN

Der Algorithmus der Caesar–Chiffrierung lässt sich wie folgt beschreiben:

Vorbereitung:

1. Alice und Bob identifizieren das Alphabet $\mathbb{A} = \{A, B, \dots, Y, Z\}$ mit den Zahlen $\{0, 1, \dots, 24, 25\}$.
2. Alice und Bob einigen sich auf einen Schlüssel $k \in \{1, \dots, 25\}$.

Verschlüsselung:

1. Alice übersetzt den Klartext m in eine Zahlenfolge (m_1, m_2, \dots, m_t) (gemäß obiger Identifikation).
2. Alice setzt für $j = 1, \dots, t$:

$$y_j = e_k(m_j) = m_j + k \mod 26$$

3. Alice setzt $y = e_k(m) = (y_1, \dots, y_t)$ und sendet y an Bob.

Entschlüsselung:

1. Bob empfängt $y = (y_1, \dots, y_t)$.
2. Bob setzt für $j = 1, \dots, t$:

$$m_j = d_k(y_j) = y_j - k \mod 26$$

und $d_k(y) = (m_1, \dots, m_t)$.

3. Bob übersetzt die Zahlenfolge (m_1, \dots, m_t) gemäß obiger Identifikation in Buchstaben und einen Klartext m .

Beispiel 1.3. Alice und Bob einigen sich auf den Schlüssel $k = 17$.

Alice will die Nachricht „*Finster wars, der Mond schien helle*“ an Bob schicken.

Verschlüsselung:

1. Alice übersetzt den Text in die Zahlenfolge

$$\begin{aligned} m = & (5, 8, 13, 18, 19, 4, 17, , 22, 0, 17, 18, ', ', 3, 4, 17, , 12, 14, 13, 3, , \\ & 18, 2, 7, 8, 4, 13, , 7, 4, 11, 11, 4) \end{aligned}$$

Satzzeichen und Leerzeichen bleiben dabei unverändert (wobei Satzzeichen hier zur Abgrenzung durch ‘’ markiert sind).

2. Für jedes $j \in \{1, \dots, 34\}$ ersetzt Alice die Zahl m_j durch

$$y_j = m_j + 17 \mod 26$$

(Leerzeichen und Satzzeichen werden nicht modifiziert.)

3. Alice erhält

$$\begin{aligned} y = & (22, 25, 4, 5, 21, 8, , 13, 17, 8, 9, ', ', 20, 21, 8, , 3, 5, 4, 20, , \\ & 9, 19, 24, 25, 21, 4, , 24, 21, 2, 2, 21) \end{aligned}$$

bzw. den Text

$$y = \text{WZEJKVI NRIJ, UVI DFEU JTYZVE YVCCV}$$

und schickt diesen an Bob.

Entschlüsselung:

1. Bob empfängt WZEJKVI NRIJ, UVI DFEU JTYZVE YVCCV und macht daraus die Zahlenfolge

$$\begin{aligned} y = & (22, 25, 4, 5, 21, 8, , 13, 17, 8, 9, ', ', 20, 21, 8, , 3, 5, 4, 20, , \\ & 9, 19, 24, 25, 21, 4, , 24, 21, 2, 2, 21) \end{aligned}$$

2. Für jedes $j \in \{1, \dots, 34\}$ ersetzt Bob die Zahl y_j durch

$$m_j = y_j - 17 \mod 26$$

(Leerzeichen und Satzzeichen werden nicht modifiziert.)

3. Bob erhält

$$\begin{aligned} m = & (5, 8, 13, 18, 19, 4, 17, , 22, 0, 17, 18, ', ', 3, 4, 17, , 12, 14, 13, 3, , \\ & 18, 2, 7, 8, 4, 13, , 7, 4, 11, 11, 4) \end{aligned}$$

und übersetzt das in den Text

$$m = \text{FINSTER WARS, DER MOND SCHIEN HELLE}$$

Beispiel 1.4. Wir chiffrieren und dechiffrieren den Text

$$m = \text{Gallia est omnis divisa in partes tres}$$

mit dem Caesar–Verfahren mit Schlüssel $k = 7$.

Der Text als Zahlenfolge ist

$$m = (6, 0, 11, 11, 8, 0, , 4, 8, 19, , 14, 12, 13, 4, 18, , 3, 8, 21, 8, 18, 0, , 8, 13, 15, 0, 17, 19, 4, 18, , 19, 17, 4, 18)$$

Nach Verschlüsselung wird daraus die Zahlenfolge

$$y = (12, 7, 18, 18, 15, 7, , 11, 15, 0, , 21, 19, 20, 11, 25, , 10, 15, 2, 15, 25, 7, , 15, 20, 22, 7, 24, 0, 11, 25, , 0, 24, 11, 25)$$

bzw. der Text

$$y = \text{NHSSPH LZA VTUPZ KPCPZH PU WHYALZ AYLZ}$$

Beim Entschlüsseln entsteht daraus zunächst wieder die Zahlenfolge

$$y = (12, 7, 18, 18, 15, 7, , 11, 15, 0, , 21, 19, 20, 11, 25, , 10, 15, 2, 15, 25, 7, , 15, 20, 22, 7, 24, 0, 11, 25, , 0, 24, 11, 25)$$

die mit $k = 7$ zu

$$m = (6, 0, 11, 11, 8, 0, , 4, 8, 19, , 14, 12, 13, 4, 18, , 3, 8, 21, 8, 18, 0, , 8, 13, 15, 0, 17, 19, 4, 18, , 19, 17, 4, 18)$$

dechiffriert wird, also zu dem Text

$$m = \text{GALLIA EST OMNIS DIVISA IN PARTES TRES}$$

Bemerkung 1.1. Das Caesar–Verfahren produziert zunächst einen vollkommen sinnlosen Text mit dem niemand etwas anfangen kann. Dennoch hat sich sehr schnell herausgestellt, dass dieses Verfahren relativ leicht anzugreifen ist.

Da es nur 25 mögliche Schlüssel k gibt, kann Catherine relativ schnell alle möglichen Schlüssel durchprobieren, dh. Catherine fängt eine Nachricht

$$y = (y_1, y_2, \dots, y_t)$$

ab und startet den folgenden Angriff

Für $l = 1, \dots, 25$:

$$\text{Für } j = 1, \dots, t: \quad b_{l,j} = y_j - l \mod 26.$$

Im Regelfall wird nur für ein l ein sinnvoller Text $b_l = (b_{l,1}, \dots, b_{l,t})$ herauskommen, und damit ist die Nachricht entschlüsselt. Meist wird dazu nicht einmal der gesamte Text benötigt. In der Regel ist schon nach wenigen Buchstaben zu erkennen, ob sich ein sinnvoller Text ergeben kann oder nicht.

Beispiel 1.5. Catherine fängt die Nachricht

$$y = \text{BJSS IJW UTXYGTYJ EBJNRFQ PQNSLJQY}$$

ab. Sie probiert den Schlüssel $l = 1$ aus und erhält aus den ersten beiden Wörtern

$$a_1 = \text{AIRR HIV}$$

Damit ist schon klar, dass das keinen sinnvollen Text ergibt. Ebenso ist es bei $l = 2$,

$$a_2 = \text{ZHQQ GHU}$$

bei $l = 3$,

$$a_3 = \text{YGPP FGT}$$

und bei $l = 4$,

$$a_4 = \text{XFOO EFS}$$

Erst bei $l = 5$ erhalten wir einen vernünftigen Anfang

$$a_5 = \text{WENN DER}$$

und auch das ganze Chiffrat ergibt einen (mehr oder minder) sinnvollen Text,

$$m = \text{WENN DER POSTBOTE ZWEIMAL KLINGELT}$$

Zur Sicherheit kann Catherine auch noch die verbleibenden Schlüssel durchspielen, für den extrem unwahrscheinlichen Fall, dass noch ein anderer einen sinnvollen Text ergibt. Das ist aber hier nicht der Fall, und $k = 5$ ist der gesuchte Schlüssel.

Bemerkung 1.2. Ein Angriff auf ein Verschlüsselungsverfahren, bei dem alle infrage kommenden Schlüssel durchprobiert werden, heißt **brute force**-Angriff. Diese Möglichkeit steht Catherine immer zur Verfügung, setzt aber voraus, dass die Anzahl der Schlüssel klein genug ist, alle durch zu probieren.

1.2. Monoalphabetische Substitution

Um die Probleme des brute force-Angriffs, der das Caesar-Verfahren kompromittiert hat, zu adressieren, ist eine Verschlüsselungstechnik gesucht, bei der es hinreichend viele Schlüssel gibt, um einen brute force-Angriff auszuschließen. Ein erster Ansatz in diese Richtung war die monoalphabetische Substitution.

Dazu schreiben Alice und Bob das Alphabet in einer beliebigen neuen Reihenfolge auf, etwa

HWXOBZVJYDRIPFUMTGNKCEALSQ

Diese Buchstabenreihenfolge ist der gemeinsame Schlüssel k von Alice und Bob. Die Verschlüsselung erfolgt nun dadurch, dass der i -te Buchstabe des Alphabets durch den i -ten Buchstaben des Alphabets in der Reihung des Schlüssels k ersetzt wird, also A durch H , B durch W , C durch X usw.

Mit diesem Schlüssel wird etwa

$m = \text{Wenn bei Capri die rote Sonne im Meer versinkt}$

das Chiffrat

$y = \text{ABFF WBY XHMGY OYB GUKB NUFFB YP PBBG EBGNYFRK}$

Die Verschlüsselung entspricht einer Permutation der Buchstaben A, \dots, Z bzw. einer Permutation der Menge $\{1, \dots, 26\}$ (wenn wir die Buchstaben mit den entsprechenden Ziffern identifizieren, wobei wir allerdings her, anders als beim Caesar–Code mit der 1 für A beginnen). Da es davon $26!$ viele gibt (das ist etwa $4 \cdot 10^{26}$) ist eine vollständige Schlüsselsuche (selbst mit Computereinsatz) zumindest sehr aufwendig. Beachtet man sogar noch Groß– und Kleinschreibung (sodass man 52 Buchstaben permutieren kann), und eventuell sogar noch Sonderzeichen, so ist definitiv jeder Rechner überfordert und ein brute force–angriff ausgeschlossen.

Der Algorithmus der monoalphabetischen Verschlüsselung lässt sich wie folgt beschreiben:

Vorbereitung:

1. Alice und Bob identifizieren das Alphabet $\mathbb{A} = \{A, B, \dots, Y, Z\}$ mit den Zahlen $\{1, 2, \dots, 25, 26\}$.
2. Alice und Bob einigen sich auf einen Schlüssel $k = \sigma \in S_{26}$, also auf eine Permutation der Zahlen $1, \dots, 26$. Bob ermittelt die Umkehrung σ^{-1} von σ .

Verschlüsselung:

1. Alice übersetzt den Klartext m , der aus t Buchstaben und Zeichen besteht, in eine Zahlenfolge (m_1, m_2, \dots, m_t) (gemäß obiger Identifikation).
2. Alice setzt für $j = 1, \dots, t$:

$$y_j = e_k(m_j) = \sigma(m_j)$$

3. Alice setzt $y = e_k(m) = (y_1, \dots, y_t)$ und sendet y an Bob.

Entschlüsselung:

1. Bob empfängt $y = (y_1, \dots, y_t)$.

2. Bob setzt für $j = 1, \dots, t$:

$$m_j = d_k(y_j) = \sigma^{-1}(y_j)$$

und $d_k(y) = (m_1, \dots, m_t)$.

3. Bob übersetzt die Zahlenfolge (m_1, \dots, m_t) gemäß obiger Identifikation in Buchstaben und einen Klartext m .

Allerdings hat sich auch dieses Verfahren nicht bewährt. Der Zugang zur Dechiffrierung monoalphabetischer Verschlüsselungen liegt in den Häufigkeitsverteilungen von Buchstaben. In der deutschen Sprache sind nicht nur nicht alle Buchstaben gleich häufig, einige kommen sogar sehr viel öfter vor als alle anderen Buchstaben.

Aus Untersuchungen ist bekannt, dass der (mit großem Abstand) häufigste Buchstabe in deutschen Texten das E ist, gefolgt vom N ; die Häufigkeiten dieser beiden Buchstaben sind

Platz	Buchstabe	Häufigkeit
1.	E	17.40%
2.	N	9.78%

Bei jedem längeren Text, der monoalphabetisch verschlüsselt ist, kann man also davon ausgehen, dass der am häufigsten auftretende Buchstabe ein E ist, und es ist auch sehr wahrscheinlich, dass der zweithäufigste Buchstabe ein N ist.

Aus den Häufigkeitstabellen erhalten wir ferner

Platz	Buchstabe	Häufigkeit
3.	I	7.55%
4.	S	7.27%
5.	R	7.00%
6.	A	6.51%
7.	T	6.15%

Bei diesen Buchstaben liegen die Häufigkeiten eng beieinander, sodass hier die Zuordnung nicht so klar ist. Oft hilft dabei die Analyse von Buchstabenpaaren (*Bigrammen*), die ebenfalls in unterschiedlichen Häufigkeiten auftreten:

Bigramm	Häufigkeit	Bigramm	Häufigkeit
<i>ER</i>	4.09%	<i>GE</i>	1.45 %
<i>EN</i>	4.00%	<i>ES</i>	1.40 %
<i>CH</i>	2.42%	<i>NE</i>	1.22 %
<i>DE</i>	2.27%	<i>UN</i>	1.19 %
<i>EI</i>	1.93%	<i>ST</i>	1.16 %
<i>ND</i>	1.87%	<i>RE</i>	1.12 %
<i>TE</i>	1.85%	<i>HE</i>	1.02 %
<i>IN</i>	1.68%	<i>AN</i>	1.02 %
<i>IE</i>	1.63%	<i>BE</i>	1.01 %

Es ist also *er* mit einer Häufigkeit von ca. 4.1 % das häufigste Bigramm der deutschen Sprache, wohingegen *re* relativ selten ist. Dadurch lässt sich das *R* in der Regel bestimmen. Ebenfalls relativ Häufige (und zwar mit ca. 1.9 %, 1.7 % und 1.6 % etwa gleich häufig) sind die Bigramme *ei*, *in* und *ie*, woraus sich im allgemeinen das *I* ermitteln lässt. Das *T* tritt als Bigramm nur in der Form *te* gehäuft auf (1.9 %) und das *A* nur als *an* (*et* und *na* sind dagegen selten). Das hilft bei der Bestimmung von *T* und *A*. Etwas seltener, aber immer noch gehäuft (mit 1.4 % bzw. 0.9 %) treten *es* und *se* auf, wohingegen *sn* kaum vorkommt (unter 0.2 %). Das hilft bei der Bestimmung von *S* (und der Unterscheidung von *S* und *I*). Darüberhinaus decken diese sieben Buchstaben (im Mittel) bereits über 60 % eines Textes ab, bei entsprechenden Substitutionen ergeben sich also bereits sinnvolle (bzw. sinnlose) Wortteile oder gar ganze Wörter, die zusätzlich helfen, die Zuordnung der Buchstaben zu erleichtern.

In der Häufigkeitsverteilung als nächstes kommen

Platz	Buchstabe	Häufigkeit
8.	<i>D</i>	5.08%
9.	<i>H</i>	4.76%
10.	<i>U</i>	4.35%

mit etwas Abstand gefolgt von

Platz	Buchstabe	Häufigkeit
11.	<i>L</i>	3.44%
12.	<i>C</i>	3.06%
13.	<i>G</i>	3.01%
14.	<i>M</i>	2.53%
15.	<i>O</i>	2.51%

Diese Buchstaben sind schon nicht mehr so häufig und weniger klar voneinander zu unterscheiden. Oft helfen auch hier Bigramme. So sind etwa *de* und *nd* relativ häufig (da 2%), und ebenso *un* mit 1.2% sowie *au* mit 0.8%. Ebenso hilft, dass *ch* mit 2.4% oft auftritt, und dass *ll* der häufigste Doppelbuchstabe ist.

Mit diesem systematischen Angriff lässt sich praktisch jeder monoalphabetisch verschlüsselte Text einer gewissen Länge vergleichsweise zügig entschlüsseln.

1.3. Vigenère–Verschlüsselung

Das Problem der Häufigkeitsanalyse adressiert eine aus dem 16. Jahrhundert stammende Verschlüsselungstechnik, die Vigenère–Chiffrierung. Die Grundidee dabei ist es, zwar die sehr einfache Caesar–Chiffrierung zu benutzen, allerdings mit unterschiedlichen Verschiebungen an verschiedenen Stellen. Um diese unterschiedlichen Verschiebungen algorithmisch in den Griff zu bekommen, wiederholen sie sich in einem bestimmten Rhythmus. Dieser Rhythmus wird in der Regel festgelegt durch ein Schlüsselwort. Nehmen wir als Beispiel den Schlüssel KRYPTO, so wird dieses Wort zunächst in Zahlenwerte übersetzt (wobei *A* der Zahl 0 entspricht, *B* der 1 usw. bis zum *Z*, das der 25 entspricht). Bei KRYPTO erhalten wir

$$\text{KRYPTO} = (10, 17, 24, 15, 19, 14) = (k_1, k_2, k_3, k_4, k_5, k_0)$$

Durch die Länge 6 des Schlüsselwortes ist schon festgelegt, dass sich die Verschiebungen nach jeweils sechs Stellen wiederholen. Der gesamte Text (ohne Leer- und Satzzeichen) wird dabei in Blöcke der Länge 6 zerlegt, und in jedem dieser Blöcke wird der *i*-te Buchstabe um k_i Stellen verschoben, hier also der erste um 10, der zweite um 17 usw. Das kann man sich dadurch veranschaulichen, dass man unter den Klartext immer wieder den Schlüssel aufführt und etwa bei der Verschlüsselung des Textes *Diese Nachricht ist geheim* wie folgt vorgeht:

Klartext	D	I	E	S	E	N	A	C	H	R	I	C	H	T
Schlüssel	K	R	Y	P	T	O	K	R	Y	P	T	O	K	R
Chiffrat	N	Z	C	H	X	B	K	T	F	G	B	Q	R	K
Klartext	I	S	T		G	E	H	E	I	M				
Schlüssel	Y	P	T		O	K	R	Y	P	T				
Chiffrat	G	H	M		U	O	Y	C	X	F				

Das Chiffrat ergibt sich dann als „Addition“ der Klartextzeile mit der Schlüsselzeile.

Für die Entschlüsselung wird der Vorgang dann einfach rückgängig gemacht.

Der zugrundeliegende Algorithmus lässt sich wie folgt beschreiben:

Vorbereitung:

1. Alice und Bob identifizieren das Alphabet $\mathbb{A} = \{A, B, \dots, Y, Z\}$ mit den Zahlen $\{0, 1, \dots, 24, 25\}$.
2. Alice und Bob einigen sich auf ein Schlüsselwort k mit l Buchstaben und schreiben dieses Schlüsselwort als $k = (k_1, \dots, k_{l-1}, k_0)$ (mit $k_i \in \{0, \dots, 25\}$).

Verschlüsselung:

1. Alice übersetzt den Klartext m in eine Zahlenfolge (m_1, m_2, \dots, m_t) (gemäß obiger Identifikation).
2. Alice berechnet für $j = 1, \dots, t$ den Wert $i = j \bmod l$ und setzt:

$$y_j = e_k(m_j) = m_j + k_i \quad \bmod 26$$

3. Alice setzt $y = e_k(m) = (y_1, \dots, y_t)$ und sendet y an Bob.

Entschlüsselung:

1. Bob empfängt $y = (y_1, \dots, y_t)$.
2. Bob berechnet für $j = 1, \dots, t$ den Wert $i = j \bmod l$ und setzt:

$$m_j = d_k(y_j) = y_j - k_i \quad \bmod 26$$

und $d_k(y) = (m_1, \dots, m_t)$.

3. Bob übersetzt die Zahlenfolge (m_1, \dots, m_t) gemäß obiger Identifikation in Buchstaben und einen Klartext m .

Beispiel 1.6. Alice und Bob einigen sich auf den Schlüssel

$$k = \text{GEHEIMSCHRIFT}$$

also (in Zahlen)

$$k = (6, 4, 7, 4, 8, 12, 18, 2, 7, 17, 8, 5, 19)$$

Alice will Bob den folgenden Text schicken (aus: Th. Fontane, *Frau Jenny Treibel*): :
An einem der letzten Maitage, das Wetter war schon sommerlich, bog ein zurueckgeschlagener Landauer vom Spittelmarkt her in die Kur- und dann in die Adlerstrasse ein und hielt gleich danach vor einem, trotz seiner Front von nur fuenf Fenstern, ziemlich

ansehnlichen, im uebrigen aber altmodischen Hause, dem ein neuer gelbbrauner Oelfarbenanstrich wohl etwas mehr Sauberkeit, aber keine Spur von gesteigerter Schoenheit gegeben hatte, beinahe das Gegenteil.

Zunächst übersetzt sie diesen Text (ohne Leer- und Sonderzeichen) in Zahlen:

$$m = (0, 13, 4, 8, 13, 4, 12, 3, 4, 17, 11, 4, 19, 25, 19, 4, 13, 12, 0, 8, 19, 0, 6, 4, \\ 3, 0, 18, 22, 4, 19, 19, 4, 17, 22, 0, 17, 18, 2, 7, 14, 13, 18, 14, 12, 12, 4, \\ 17, 11, 8, 2, 7, 1, 14, 6, 4, 8, 13, 25, 20, 17, 20, 4, 2, 10, 6, 4, 18, 2, \\ 7, 11, 0, 6, 4, 13, 4, 17, 11, 0, 13, 3, 0, 20, 4, 17, 21, 14, 12, 18, 15, 8, \\ 19, 19, 4, 11, 12, 0, 17, 10, 19, 7, 4, 17, 8, 13, 3, 8, 4, 10, 20, 17, 20, 13, \\ 3, 3, 0, 13, 13, 8, 13, 3, 8, 4, 0, 3, 11, 4, 17, 18, 19, 17, 0, 18, 18, 4, 4, 8, \\ 13, 20, 13, 3, 7, 8, 4, 11, 19, 6, 11, 4, 8, 2, 7, 3, 0, 13, 0, 2, 7, 21, 14, 17, \\ 4, 8, 13, 4, 12, 19, 17, 14, 19, 25, 18, 4, 8, 13, 4, 17, 5, 17, 14, 13, 19, \\ 21, 14, 13, 13, 20, 17, 5, 20, 4, 13, 5, 5, 4, 13, 18, 19, 4, 17, 13, 25, 8, \\ 4, 12, 11, 8, 2, 7, 0, 13, 18, 4, 7, 13, 11, 8, 2, 7, 4, 13, 8, 12, 20, 4, 1, 17, 8, \\ 6, 4, 13, 0, 1, 4, 17, 0, 11, 19, 12, 14, 3, 8, 18, 2, 7, 4, 13, 7, 0, 20, 18, 4, \\ 3, 4, 12, 4, 8, 13, 13, 4, 20, 4, 17, 6, 4, 11, 1, 1, 17, 0, 20, 13, 4, 17, 14, \\ 4, 11, 5, 0, 17, 1, 4, 13, 0, 13, 18, 19, 17, 8, 2, 7, 22, 14, 7, 11, 4, 19, 22, \\ 0, 18, 12, 4, 7, 17, 18, 0, 20, 1, 4, 17, 10, 4, 8, 19, 0, 1, 4, 17, 10, 4, 8, 13, \\ 4, 18, 15, 20, 17, 21, 14, 13, 6, 4, 18, 19, 4, 8, 6, 4, 17, 19, 4, 17, 18, 2, 7, \\ 14, 4, 13, 7, 4, 8, 19, 6, 4, 6, 4, 1, 4, 13, 7, 0, 19, 19, 4, 1, 4, 8, 13, 0, 7, \\ 4, 3, 0, 18, 6, 4, 6, 4, 13, 19, 4, 8, 11)$$

Insgesamt hat sie damit 382 Zeichen. Sie verlängert den Schlüssel durch Wiederholen zu einem Schlüssel k_{ext} der Länge 382 (dh. sie wiederholt k 29 mal, dann kommen noch die ersten fünf Buchstaben 6, 4, 7, 4, 8 dazu)

$$k_{\text{ext}} = (6, 4, 7, 4, 8, 12, 18, 2, 7, 17, 8, 5, 19, 6, 4, 7, 4, 8, 12, \dots, 8, 5, 19, 6, 4, 7, 4, 8)$$

und sie berechnet

$$y = m + k_{\text{ext}} \quad \text{mod } 26$$

und erhält

$$y = (6, 17, 11, 12, 21, 16, 4, 5, 11, 8, 19, 9, 12, 5, 23, 11, 17, 20, 12, 0, 21, 7, 23, 12, \\ 8, 19, 24, 0, 11, 23, 1, 16, 9, 24, 7, 8, 0, 7, 0, 20, 17, 25, 18, 20, 24, 22, 19, 18, \\ 25, 10, 12, 20, 20, 10, 11, 12, 21, 11, 12, 19, 1, 21, 10, 15, 25, 10, 22, 9, 11, \\ 19, 12, 24, 6, 20, 21, 25, 16, 19, 19, 7, 7, 24, 12, 3, 13, 16, 19, 9, 23, 13, 12, 25, \\ 8, 18, 16, 8, 3, 2, 21, 14, 21, 25, 13, 6, 9, 12, 11, 14, 2, 3, 12, 15, 10, 20, 8, 18, \\ 6, 14, 17, 10, 12, 12, 12, 21, 13, 11, 8, 0, 24, 10, 6, 22, 25, 8, 12, 20, 5, 22, 20, \\ 20, 15, 13, 23, 17, 23, 13, 15, 12, 20, 20, 9, 10, 17, 21, 5, 21, 13, 25, 21, 21, \\ 12, 20, 5, 6, 19, 10, 25, 19, 12, 5, 22, 11, 12, 21, 16, 9, 7, 24, 5, 21, 24, 14, 20, \\ 17, 20, 24, 25, 17, 12, 6, 20, 22, 13, 9, 6, 24, 23, 11, 21, 21, 11, 0, 6, 19, 2, 16, \\ 7, 0, 6, 17, 25, 8, 15, 25, 3, 10, 9, 24, 12, 18, 1, 18, 24, 11, 5, 25, 20, 24, 6, 20, \\ 17, 9, 9, 10, 6, 15, 0, 16, 22, 15, 0, 20, 9, 24, 12, 18, 0, 6, 24, 25, 8, 11, 16, 4, 6, \\ 15, 4, 21, 9, 13, 10, 21, 13, 8, 19, 13, 19, 19, 7, 11, 21, 9, 10, 20, 8, 18, 9, 8, 3, \\ 19, 6, 20, 17, 21, 23, 12, 23, 12, 9, 11, 4, 0, 25, 13, 11, 10, 4, 5, 11, 18, 8, 14, \\ 21, 0, 12, 12, 3, 11, 8, 18, 9, 1, 25, 4, 8, 8, 25, 22, 22, 10, 20, 21, 0, 20, 13, 23, \\ 25, 21, 17, 14, 16, 10, 21, 11, 25, 14, 9, 10, 25, 8, 24, 22, 10, 19, 6, 6, 20, 24, \\ 12, 13, 12, 12, 8, 13, 8, 9, 16, 5, 9, 7, 10, 1, 9, 20, 10, 12, 20, 4, 15, 16, 21, 2, \\ 25, 23, 12, 11, 23, 19, 23, 11, 12, 19)$$

bzw. als Text (wieder mit Leer- und Sonderzeichen):

Gr lmvqe fli tjmfplr Umavhxm, ity Alxbqj yhi ahaur zsuywtszkm, uuk lmv lmtbvkpzkujlt-myguvz Qtthhymd nqt Jxnmzisqidcv ovz ng jml Ocd- mpk uisg or kmm Mvnliaykgwzi muf wuu pnxrx npmuuj krvfn zvv mufgt, kztmf wlmvqj Hyfvy our uyz rmguw Njgyxlvv, lagtc-qha grzipzdkjyms, bs ylfzuygu rjjk gpaqwpauijyms Agyzi, lqe gpe vjnkv nitntthlvjk Uisjidl-gurvxmxmjl eazn lkefl siov Ammdlisjbz, eizz wwkuv Aunx zvr oqkvlojkziy Wktgguynmnm minijqf jhkbj, ukmuepq vcz Xmlxtxlmt.

Bemerkung 1.3. Das Vigenère–Verfahren enthält zwei entscheidende Sicherheitsmechanismen:

1. Bei einem Schlüsselwort k der Länge l gibt es 25^l viele verschiedene Verschlüsselungen. Da Catherine weder den Schlüssel k noch seine Länge l kennt, ist ein brute-force–Angriff aussichtslos.
2. Da jeder Buchstabe (in einem Block) um eine unterschiedliche Anzahl von Stellen verschoben wird, ist auch eine Häufigkeitsanalyse aussichtslos.

Aus diesen Gründen galt das Vigenère–Verfahren lange Zeit als nicht brechbar. Erst 1863 veröffentlichte F. Kasiski einen ersten erfolgreichen Angriff auf dieses Verfahren, den sogenannten **Kasiski–Test**. Die Grundidee dahinter ist:

1. Ermittle zuerst die Schlüssellänge l .
2. Ermittle dann für jedes $i \in \{0, \dots, l - 1\}$ die Verschiebung k_i durch eine Häufigkeitsanalyse.

Im Detail geht man dazu bei einem Chiffrat $y = (y_1, y_2, \dots, y_t)$ (das wir hier ohne Satz- und Leerzeichen schreiben) wie folgt:

1. Untersuche den Text auf Buchstabenfolgen, die wiederholt vorkommen (Bigramme, Trigramme, allgemein N –Gramme). Die Grundannahme des Angriffs ist nun, dass diese Wiederholungen oft von denselben Buchstabenfolgen aus dem Originaltext stammen, eine Annahme, die durch die allgemeine Sprachstruktur gerechtfertigt ist. Die Abstände dieser N –Gramme sind dann Vielfache der Schlüssellänge l . Die Schlüssellänge l ist daher eine Zahl, die möglichst viele dieser Abstände teilt.
2. Für alle Kandidaten l , die aus (1) als Schlüssellängen in Frage kommen, und jedes $i \in \{0, \dots, l - 1\}$ bestimme die Menge

$$M_i = \{y_j : j \bmod l = i\}$$

also die Menge aller Buchstaben, deren Position sich nur um ein Vielfaches von l unterscheiden, bestimme den häufigsten Buchstaben dieser Menge, identifiziere diesen Buchstaben mit e und leite daraus die Verschiebung k_i ab.

3. Falls sich noch kein sinnvoller Gesamttext aus den so ermittelten Verschiebungen ergeben sollte, untersuche den Text auf sinnvolle Teile und ändere die Verschiebung an den Stellen, an denen Buchstaben offensichtlich nicht passen, indem etwa der zweithäufigste Buchstabe mit e identifiziert wird oder der häufigste mit n .

In der Regel kann damit ein erfolgreicher Angriff auf das Vigenère–Verfahren durchgeführt werden, wenn

- der Text „hinreichend lang“ und
- der Schlüssel „hinreichend kurz“

ist.

Beispiel 1.7. Catherine fängt den folgenden Vigenère-verschlüsselten Text ab:

Rusd xoe, zmb! Ggsffryjyho, Dlqsmkdbyz txx Ddncqhx, oec vyzcol rtmb Kgoicnqcv celtgkoj rdouholk, lsn ydsmjdw Vvleyym. Nu jsob zbr hlm swy zbgvq Dii txx shx mf jvox zvm nho tluyl; ydsmjd Wuxhcenvq, ryzrcy Ununfq qui, txx qhobv rmbfm kh uho tvgoh Azrl, ydbule, ryizl oec aovq ehu jbndl, wyzmo Mtgeycdb ue col Ezcy ydbod txx jdry, uzcm nhb hzbrnj vsmwdx ef dxhvm! Nuj vsfc lsl jbrcvq nuj Golq uolsqohedx. Tnzb uzm swy fomtgockdb ucr kfcd ncv Kkzwdx, Xfjdiidx Grfsmkdb, Mtgbzyaol lmn Jwzpzm; wctg zfrfoh bdshv Rulloof enmb Qvocwdv, zldbwyso gzbr qvcol mnb Bfdvfv mywy Soowdv, xreyyi hcn dhb ulbr ucko Zidex vmdlzrcye, asfud wci mswys oce vkm Idmbkr jo nhcmvm. Lccco gzq xctgd yzm, swy jyyemdy nzc fvgye csy Ddxmtgoh qt lyjrole txx qt lybdrlvm. Kotg rus hmb ndnyi Fen enmb Xdvx, enmb Vgb oec Ryiqvctguyzs nyi Vofk; dc gfdmbkd uyzm Roec ci czohxdb fvaoh! Uqeg yzl ctg wctg nyi Lkazd olxdlye, nl gzq noibr Avhcnvr Ulred oec Woec xctgd grmmb Xdryzlxjc veyico elmn; xrrc ctg xctgd gvgb, gzs culdbg Jbrqvhcm, qt cuwdx vizewyd guj hmb ehmbk vocjr; nujr swydbevmxy nzc xxz GyCs sg Zmxyirdye yemrlwyegkycs, cwyze ucko Qzquyerulred oec Cuddx, oec do ehmbk lobi hx Qfqdye jbuadx.

Zunächst untersucht sie den Text auf Brigramme und Trigramme, die sich wiederholen und auf deren Positionen. Dazu empfiehlt es sich, aus dem Text alle Leer- und Sonderzeichen zu entfernen und alles einheitlich klein oder groß zu schreiben:

rusdxoezbggssffryjyholqsmkdbyztxddncqhxoecvyzcolrtmbkgoiicnqcvceltgkojrdouholklsnyd smjdwwvleymnusobzbrhlmswyzbgvqdiitxxshxmjvoxzvmnhotluyllydsmjdwxuhcnvqryzrcyun unfqquitxxqhabvrbfmkhuhotvgohazrllydbuleryizloecaovqehujbodlwyzmamtgeycdbuecolezc ydbodtxxjdryuzcmnhbhzbrnjvsmjdxefdxhvmnujvsfcsljbrcvqnujgolquolsqohedxtnzbvzmswyfo mtgockdbucrkfcndcvkkzwxxfjdiidxgrfsmkdbmtgbazaollmnjwzpzvwmwctgzfrfohbdshvruullofe nmbqvocwdvzldbwysogzbrqvcolumnbbfdvfmywysoowdvxreeyihcndhbulbruckozidexvmdlzrcye asfudwcimswyosocevkmidmbkrjonhcmvmlccogzxctgdyzmswyjyyemdynzcfvgyecsyddxmtgo hqtljyroletxxqtlbybdrlvkmotgrushmbndnyifenenmbxdvxnmbvgboecryiqvctguyzsnyivofkdcgf mbkduyzmroecciczeichfdbfvaohuqegyzlctgwctgnyilkazdolxdlyenlgzqnoibravhcnvrulredoecwoec xctgdgrmmbxdryzlxjcveyicoelmnxrrcctgxctgdvgbzsculdbqjbrqvhcmqtcuxdvvizewydgujhmbe hmbkvocjrnujrswydbevmxynzczdgycssgzmxyirdyeeyemrlwyegkykcscwyzeuckoqzquyerulredoe ccuddxoecdoehmbklobihxqfqdyejbuadx

Damit kann die Analyse recht einfach mit entsprechender Software durchgeführt werden, und Catherine erhält:

Am häufigsten findet sie die folgenden Trigramme:

Trigramm	Anzahl	Stellen	sukkzessive Differenzen
oec	8	42, 210, 626, 662, 730, 734, 868, 906	168, 416, 36, 68, 4, 164, 8 (ggT = 4)
ctg	8	390, 534, 634, 686, 690, 738 770, 774	144, 100, 52, 4, 48, 32, 4 (ggT = 4)
txx	5	32, 120, 172, 252, 580	88, 52, 80, 328 (ggT = 4)
mbk	5	53, 513, 653, 821, 913	460, 140, 168, 928 (ggT = 4)
swy	5	109, 325, 501, 541, 833	216, 176, 40, 292 (ggT = 4)

Alle anderen Trigramme treten höchstens viermal auf. Die Abstände der Trigramme deuten darauf hin, dass die Länge des Schlüssels 4 oder ein Vielfaches davon ist (beachten Sie: auch 8 teilt recht viele Differenzen).

Die am häufigsten auftretenden Bigramme sind:

Bigramm	Anzahl	Stellen	sukkzessive Differenzen
tg	14	67, 231, 331, 371, 391, 535, 567, 595, 635, 687, 691, 739, 771, 775	164, 100, 40, 20, 144, 32, 28, 40, 52, 4, 48, 32, 4 (ggT = 4)
mb	13	9, 53, 181, 413, 513, 601, 613, 621, 653, 745, 817, 821, 913	44, 128, 232, 100, 88, 12, 8, 32, 92, 72, 4, 92 (ggT = 4)
yz	12	330, 46, 111, 158, 226, 374, 538, 638, 658, 683, 750, 879	16, 105, 47, 68, 148, 164, 100, 20, 25, 67, 129 (ggT = 1)
oe	11	6, 42, 210, 626, 662, 730, 734, 761, 898, 906, 910	36, 168, 416, 36, 68, 4, 27, 137, 8, 4 (ggT = 1)
wy	11	10, 225, 326, 426, 450, 502, 542, 810, 834, 869, 878	115, 101, 100, 24, 52, 40, 268, 24, 35, 9 (ggT = 1)

Bei *yz* ist der größte gemeinsame Teiler der sukzessiven Differenzen gleich 1, nehmen wir aber 111, 683 und 879 aus, so ergibt sich auch hier ein größter gemeinsamer Teiler von 4.

Ähnlich ist es bei *oe* und *wy*: Auch hier ist der größte gemeinsame Teiler der sukzessiven Differenzen gleich 1, allerdings teilt 4 sehr viele der Differenzen.

Wir haben also Grund zur Annahme, dass der Schlüssel die Länge 4 hat (eventuell 8 oder ein anderes Vielfaches von 4).

Buchstaben an den Stellen mit folgendem Rest mod 4:

Rest 0: String 0 = *dzgrhqdtdhccgnegrhlddlmsbmzqthjzhuddhqrnqthrmhgdezcqjlmgdczdtdzbvddmvlbqguqdzmfgdrkdjdfgamzmgfdronvddsbcnmdmsdehhbkdmradmsvdrhmcqgmj*

*mzgcdgtrttdmghdfndngcqgsvdddmczdaqzglddnqbhreccgndlvcmruggsdbhtdzdhvrrdm
zdsmyrlgszkqrecdchlhqjd*

Rest 1: String 1 = *rxmsyosbxnxvomoqekdosswenorsbdxxvvoyswercuqxomkoorbrlaebwoebocbx
rcbrsxxnssrnoooxbsoobknkdxsbbonpwzosuomovborbvyovecbroedcswsokmjcloxdsyd
cbsxoloxlrkrmnemvmbrvunocmurcoboelwnkollnrcudwdxmrxeoncxdbcbrccxegmmonsbr
xcgsxdewkceouudcxmoxdbx*

Rest 2: String 2 = *uobfjdmyxcoylbicloolnmvyubhwgixmomtlmunyynuxbbhthluyoohoymyulyox
ymhnmehuflcullhtvumcufczxigmmyljzcfhhlfbczwgqlbfwoxnuuzzlyfcwcmbomcgcywy
fyymhylxyloubynbxboyccyfgbyoihfhgccyalgoanloocgbocyexccggugqmuvwubbcuweyxyg
yymyywuqylouoobbqyu*

Rest 3: String 3 =
*segfylkzdqezrkcvtjukyjvyjzlyvisfxnlyjxzufiqvfuwaylievudzceeydjunzjjfvjcjvjqsenzytkc
cvwfirktslwvtrbuleqwllyzvmfywridlcivzeuiyeiknvctzyenvedtqjeqbvtsniexeveitzikfkzec
xvuuyttizxezivvreertrxzjilrttvzljvqxiyjekjjyvnzczierecyczeredeekifed*

Häufigkeitsanalyse von String 0:

Der Buchstabe *d* tritt 45 mal auf, *g* 22 mal und *m* 20 mal. Also wurde vermutlich *e* auf *d* verschoben, also um 25 Stellen; im Schlüssel ist daher wohl ein *z* an Stelle 0 bzw. 4.

Häufigkeitsanalyse von String 1:

Der Buchstabe *o* tritt 36 mal auf, „*x*“ 24 mal und „*b*“ 21 mal. Also wurde vermutlich *e* auf *o* verschoben, also um 10 Stellen; im Schlüssel ist daher wohl ein *k* an Stelle 1.

Häufigkeitsanalyse von String 2:

Der Buchstabe *y* tritt 37 mal auf, *o* 20 mal und *u* und *c* je 19 mal. Also wurde vermutlich *e* auf *y* verschoben, also um 20 Stellen; im Schlüssel ist daher wohl ein *u* an Stelle 2.

Häufigkeitsanalyse von String 3:

Der Buchstabe *e* tritt 27 mal auf, *v* 25 mal und *z* 21 mal. Hier ist keine eindeutige Aussage möglich, sehr wahrscheinlich sind Verschiebungen von *e* auf *e* (also um 0 Stellen) oder von *e* auf *v* (also um 17 Stellen), nicht ausgeschlossen ist aber auch eine Verschiebung von *e* auf *z*, also um 21 Stellen; im Schlüssel ist daher wohl ein *a* an Stelle 3, oder ein *r* oder ein *v*.

Als Schlüsselwörter kommen somit in Frage: KUAZ, KURZ und KUVZ‘.

Dechiffrierung mit dem Schlüsselwort KUAZ ergibt:

*Hase nue, ach! Ghilfsopyie, Jlriskerez und Dediqin, ued lezder ruch Kheocogiv durthauj
stuuierk, mit yeisjem Bvmueyn. Da jteh zch nln icy armvr Toi und sin sf klux als nie
zvor; yeisje Maxistvr, hezsse Uoktfr gai, und qiehv schfn an uie zvhen Aahr, yeralf,*

heiab ued quvr unu krudm, mezne Sthuecer ae der Ease yerud und jehe, uass nir nzchtj wisjen kfennvn! Daj wilc mir jchivr daj Herq versreneen. Znar bzn icy gestheiker acs alce div Lafwen, Dfktoien Mrgisker, Sthrezber lnd Pwaffvn; mith plrgen beinv Skrlpel eoch Qweiwel, flercyte mzch wder mor Hfellv nocy Teuwel, drfuei ist dir alch acle Fieud vntrzssee, bilue mii nicyt eie was Iechks zu nissvn. Bicde mjr nitht ezn, icy koeente nas lhree die Densthen qu bejsere und qu bebehrvn. Auth has ich nedei Gut eoch Xeld, eoch Vhr ued Heirlithkezt dei Welk; es mfechke kezn Hued so caenxer lvben! Urum yab ith mith dei Magze erxebee, ob mjr duich Gvistvs Krrft ued Mued nitht mrnch Xehezmij wueide klnd; drss ith nitht mvhr, mzt salerm Jchwiss, qu saxen biaucye waj ich eichk weijs; dajs icy erkvnne nas dze Wect im Znneistee zusrmmeahaect, scyau acle Wzrkeeskrrft ued Saden, ued tu eichk mehi in Wfrtee kraden.

Der Text ist insgesamt nicht lesbar (auch wenn einige Wörter und Wortteile, wie **ach**, **und**, **an**, **nie** oder **mit**, schon sinnvoll klingen). Das spricht dafür, dass wir zumindest mit drei Buchstaben (vermutlich K, U und Z) schon richtig liegen. Insgesamt war dieser Schlüssel vermutlich aber falsch.

Dechiffrierung mit dem Schlüsselwort KURZ ergibt:

Habe nun, ach! Philosophie, Juristerei und Medizin, und leider auch Theologie durchaus studiert, mit heissem Bemuehn. Da steh ich nun ich armer Tor und bin so klug als wie zuvor; heisse Magister, heisse Doktor gar, und ziehe schon an die zehen Jahr, herauf, herab und quer und krumm, meine Schueler an der Nase herum und sehe, dass wir nichts wissen koennen! Das will mir schier das Herz verbrennen. Zwar bin ich gescheiter als alle die Laffen, Doktoren Magister, Schreiber und Pfaffen; mich plagen keine Skrupel noch Zweifel, fuerchte mich weder vor Hoelle noch Teufel, dafuer ist mir auch alle Freud entrissen, bilde mir nicht ein was Rechts zu wissen. Bilde mir nicht ein, ich koennte was lehren die Menschen zu bessern und zu bekehren. Auch hab ich weder Gut noch Geld, noch Ehr und Herrlichkeit der Welt; es moechte kein Hund so laenger leben! Drum hab ich mich der Magie ergeben, ob mir durch Geistes Kraft und Mund nicht manch Geheimnis wuerde kund; dass ich nicht mehr, mit sauerm Schweiss, zu sagen brauche was ich nicht weiss; dass ich erkenne was die Welt im Innersten zusammenhaelt, schau alle Wirkenskraft und Samen, und tu nicht mehr in Worten kramen.

(aus J.W. von Goethe, *Faust. Der Tragödie erster Teil*)

Wir erhalten einen vernünftigen Text, was für diesen Schlüssel spricht.

Zur Kontrolle auch noch die Dechiffrierung mit dem Schlüsselwort KUVZ:

Haxe nuj, ach! Lhilksopdie, Jqrisperee und Iedivin, ujd leeder wuch Pheohogia duryhauo stuzierp, mit deisoem Bamuedn. Da oteh ech nqn icd armar Ton und xin sk kluc als sie zqvor; deisoe Macistar, heesse Zoktkr gan, und vieha schkn an zie zahen Fahr, deraqf,

henab ujd quar unz kruim, meene Syhueher aj der Jase derui und oehe, zass sir nechto wisoen kkennen! Dao wilh mir ochiar dao Herv verxrenjen. Zsar ben icd gesyheiper ahs alhe dia Lafben, Dkktonen Mwgisper, Syhreeber qnd Pbaffan; miyh plwgen geina Skrqpel joch Vweibel, fqercdte mech wader ror Hkella nocd Teubel, dwfuen ist iir aqch ahle Fneud antressej, bilze min nicdt eij was Nechps zu sissan. Bihde mer niyht een, icd koejnte sas lahrej die Iensyhen vu beoserj und vu begehran. Auyh hax ich seden Gut joch Celd, joch Ahr ujd Henrliyhkeet den Welp; es mkechpe keen Hujd so haencer laben! Zrum dab iyh miyh den Magee ercebej, ob mer dunch Gaistas Kruft ujd Mujd niyht munsch Ceheemnio wuende kqnd; dwss iyh niyht mahr, met saqerm Ochwaiss, vu sacen bnaucde wao ich jichp weios; daos icd erkanne sas dee Weht im Ennenstej zuswmmejhaeht, scdau ahle Werkejskrwft ujd Saien, ujd tu jichp mehn in Wkrtej kraien.

Auch hier zwar einige sinnvolle Wortteile, insgesamt aber kein lesbarer Text. Das Schlüsselwort war demnach KURZ, der Klartext der, den wir im zweiten Versuch erhalten haben.

2. Grundlagen der Kryptologie

Kryptologie zerfällt in die beiden Teilgebiete

- Kryptographie
- Kryptoanalyse

Dabei beschäftigt sich die Kryptographie mit dem Absichern und Verschlüsseln von Daten und hat die folgenden drei wesentlichen Untergebiete

- symmetrische Chiffrierung
- asymmetrische Chiffrierung
- kryptographische Protokolle

Die ersten beiden Untergebiete beschäftigen sich mit diversen Algorithmen zum Absichern von Daten. Mit den kryptographischen Protokollen werden Gesamtprozesse, basierend auf den kryptographischen Algorithmen, konstruiert, die eine sichere Kommunikation zwischen Alice und Bob ermöglichen sollen, etwa das TLS–Protokoll (**T**ransport **L**ayer **S**ecurity) oder das SSL–Protokoll (**S**ecurity **S**ocket **L**ayer), das in Webbrowsersn zum Einsatz kommt.

Die Kryptoanalyse (auch Kryptanalyse) beschäftigt sich mit dem Brechen von Kryptosystemen, also mit Angriffen auf kryptographische Algorithmen und Protokolle.

Jeder kryptographische Algorithmus hat im wesentlichen drei Komponenten:

1. Einen Schlüssel k aus einem Schlüsselraum K .
2. Einen Verschlüsselungsalgorithmus (Chiffrierungsverfahren) $e = e_k$ (der vom Schlüssel k abhängt), der aus einem Klartext m ein Chiffrat $y = e(m)$ erzeugt.
3. Einen Entschlüsselungsalgorithmus (Dechiffrierungsverfahren) $d = d_k$ (der ebenfalls von k abhängt), der aus dem Chiffrat y den Klartext $m = d(y)$ zurückgewinnt.

2.1. Symmetrische Verschlüsselungsverfahren

Symmetrische Verfahren zeichnen sich dadurch aus, dass es für einen Ver- und Entschlüsselungsvorgang einen Schlüssel k gibt, der sowohl e_k als auch d_k schon eindeutig und berechenbar bestimmt. Beispiele hierfür sind etwa die Caesar–Chiffren bzw. die monoalphabetische Substitution. Ist hier die Verschiebung (bei Caesar) bzw. die Permutation festgelegt, so sind dadurch sowohl die Ver- als auch die Entschlüsselung bestimmt.

Prinzipielle Vorgehensweise:

1. Alice und Bob tauschen einen (geheimen) Schlüssel k aus.
2. Alice benutzt diesen Schlüssel k , um den Klartext m zu verschlüsseln,

$$y = e(m) = e_k(m)$$

3. Bob benutzt diesen Schlüssel k , um das Chiffraut y zu entschlüsseln,

$$m = d(y) = d_k(y)$$

Die Verfahren $e = e_k$ und $d = d_k$ sind dabei Algorithmen, die von k abhängen, und für die gilt

$$d_k(e_k(m)) = m$$

dh. mathematisch gesprochen ist d_k eine linksinverse Abbildung zu e_k . Beachten Sie dabei, dass nicht notwendig gelten muss, dass auch $e_k(d_k(y)) = y$ gilt, obwohl das bei allen gängigen Verfahren der Fall ist.

Der Klartext besteht dabei aus einer Folge von Buchstaben aus einem Alphabet \mathbb{A} , wobei \mathbb{A} eine beliebige (in der Regel endliche) Menge sein kann. Gängige Alphabete sind

- $\mathbb{A} = \{A, B, C, \dots, X, Y, Z\}$, das klassische Alphabet.
- \mathbb{A} ist der ASCII-Zeichensatz (oder der UTF8-Zeichensatz).
- $\mathbb{A} = \{0, 1\} = \mathbb{F}_2$.
- $\mathbb{A} = \{0, 1\}^8 = \mathbb{F}_2^8 = \mathbb{F}_{256}$, die Menge der binären Acht-Tupel.
- $\mathbb{A} = \mathbb{Z}_n \quad (= \mathbb{Z}/n \cdot \mathbb{Z})$.

Das Chiffraut besteht ebenfalls aus einer Folge von Buchstaben, die aber nicht aus dem gleichen Alphabet stammen müssen wie der Klartext (z.B. kann eine Textnachricht als binäre Zeichenfolge verschlüsselt werden).

2.2. Blockchiffrierung

Ein Blockverschlüsselungsverfahren der Länge r ist ein Verschlüsselungsalgorithmus, der den Klartext m in Blöcke der Länge r aufteilt (gegebenenfalls mit Auffüllen durch geeignete Zeichen), und der aus jedem Klartextblock m_j der Länge r nach einem eindeutigen

Verfahren (das von dem gewählten Schlüssel k abhängig ist) ein Chiffrat y_j derselben Länge r (und mit demselben Alphabet) erzeugt. Für jeden Schlüssel k ist daher

$$e = e_k : \mathbb{A}^r \longrightarrow \mathbb{A}^r$$

eine injektive Abbildung und damit bijektiv, wenn das Alphabet endlich ist, mit Umkehrabbildung d_k .

Beispiel 2.1. Das Vigenère–Verfahren mit dem Schlüssel $k = \text{KRYPTO}$ ist eine Blockchiffrierung der Länge 6 mit dem Alphabet

$$\mathbb{A} = \{A, B, \dots, Y, Z\} = \{0, 1, \dots, 24, 25\}$$

wobei

$$e_k : \mathbb{A}^6 \longrightarrow \mathbb{A}^6$$

gegeben ist durch

$$e_k(m_1, m_2, m_3, m_4, m_5, m_6) = (m_1 + 10, m_2 + 17, m_3 + 24, m_4 + 15, m_5 + 19, m_6 + 14) \mod 26$$

Die Dechiffrierung

$$d_k : \mathbb{A}^6 \longrightarrow \mathbb{A}^6$$

ist gegeben durch

$$\begin{aligned} d_k(y_1, y_2, y_3, y_4, y_5, y_6) &= (y_1 - 10, y_2 - 17, y_3 - 24, y_4 - 15, y_5 - 19, y_6 - 14) \mod 26 \\ &= (y_1 + 16, y_2 + 9, y_3 + 2, y_4 + 11, y_5 + 7, y_6 + 12) \mod 26 \end{aligned}$$

Bemerkung 2.1. Eine Blockchiffrierung der Länge r ist eine Permutation der Elemente von \mathbb{A}^r (abhängig von einem Schlüssel k). Es gibt also insgesamt $(|\mathbb{A}|^n)!$ viele Blockchiffrierungen der Länge r .

Ein Operationsmodus für eine Blockchiffrierung ist eine Verfahrensvorschrift, die festlegt, wie ein längerer Text im Rahmen einer bestimmten Blockchiffierungsmethode zu verarbeiten ist.

Definition 2.1. Für zwei Tupel $x = (x_1, \dots, x_r) \in \mathbb{A}^r$ und $y = (y_1, \dots, y_s) \in A^s$ bezeichnet

$$x \| y = (x_1, \dots, x_r, y_1, \dots, y_s) \in \mathbb{A}^{r+s}$$

die **Konkatenation** von x und y .

2.2.1. ECB–Mode (Electronic Code Book–Mode)

Der ECB–Modus ist das vielleicht naheliegendste Verfahren.

Vorbereitung:

1. Bob und Alice einigen sich auf einen Schlüssel k .
2. Alice bestimmt das zugehörige Verschlüsselungsverfahren

$$e = e_k : \mathbb{A}^r \longrightarrow \mathbb{A}^r$$

3. Bob bestimmt das zugehörige Entschlüsselungsverfahren

$$d = d_k : \mathbb{A}^r \longrightarrow \mathbb{A}^r$$

Verschlüsselung:

1. Alice ergänzt den Klartext m so, dass seine Länge durch r teilbar ist.
2. Alice teilt m in Blöcke der Länge r auf,

$$m = m_1 \| m_2 \| \dots \| m_t$$

3. Für $j = 1, \dots, t$ berechnet Alice

$$y_j = e(m_j)$$

4. Alice schickt

$$y = y_1 \| y_2 \| \dots \| y_t$$

Entschlüsselung:

1. Bob teilt die eingegangene Nachricht y in Blöcke der Länge r auf,

$$y = y_1 \| y_2 \| \dots \| y_t$$

2. Für $j = 1, \dots, t$ berechnet Bob

$$m_j = d(y_j)$$

3. Bob setzt

$$m = m_1 \| m_2 \| \dots \| m_t$$

Bemerkung 2.2. Im ECB–Modus werden gleiche Texte und gleiche Textpassagen (an passenden Stellen) auch immer in gleiche Chiffre umgewandelt (solange sich der Schlüssel nicht ändert). Kennt Catherine den Klartext zu gewissen Textpassagen (z.B. Begrüßungsformeln, die immer an der gleichen Stelle der Nachricht stehen), so kann sie daraus eventuell Information über den verwendeten Schlüssel ableiten (*plaintext-ciphertext-attack*).

Daher sollte der Schlüssel häufig geändert werden, um Catherine die Analyse zu erschweren.

Beispiel 2.2. Das Vigenère–Verfahren, wie wir es bis jetzt benutzt haben, arbeitet (bis auf das Padding) im ECB–Modus.

Betrachten wir als Beispiel den Schlüssel

$$k = \text{KRYPTO} = (10, 17, 24, 15, 19, 14)$$

und den Klartext

$$m = \text{Verschlüsselungsverfahrensweg}$$

also in Zahlen

$$m = (21, 4, 17, 18, 2, 7, 11, 20, 4, 18, 18, 4, 11, 20, 13, 6, 18, 21, 4, 17, 5, 0, 7, 17, 4, 13, 18, 22, 4, 6)$$

Die Nachricht m zerfällt schon in fünf Blöcke der Länge 6

$$\begin{aligned} m_1 &= (21, 4, 17, 18, 2, 7) \\ m_2 &= (11, 20, 4, 18, 18, 4) \\ m_3 &= (11, 20, 13, 6, 18, 21) \\ m_4 &= (4, 17, 5, 0, 7, 17) \\ m_5 &= (4, 13, 18, 22, 4, 6) \end{aligned}$$

Die Chiffrierung kann damit formal wie folgt durchgeführt werden (alle Rechnungen modulo 26):

$$\begin{aligned} y_1 &= (21, 4, 17, 18, 2, 7) + (10, 17, 24, 15, 19, 14) &= (5, 21, 15, 7, 21, 21) \\ y_2 &= (11, 20, 4, 18, 18, 4) + (10, 17, 24, 15, 19, 14) &= (21, 11, 2, 7, 11, 18) \\ y_3 &= (11, 20, 13, 6, 18, 21) + (10, 17, 24, 15, 19, 14) &= (21, 11, 11, 21, 11, 9) \\ y_4 &= (4, 17, 5, 0, 7, 17) + (10, 17, 24, 15, 19, 14) &= (14, 8, 3, 15, 0, 5) \\ y_5 &= (4, 13, 18, 22, 4, 6) + (10, 17, 24, 15, 19, 14) &= (14, 4, 16, 11, 23, 20) \end{aligned}$$

Damit erhalten wir das Chiffrat

$$y = (5, 21, 15, 7, 21, 21, 11, 2, 7, 11, 18, 21, 11, 11, 21, 11, 9, 14, 8, 3, 15, 0, 5, 14, 4, 16, 11, 23, 20)$$

also

$$y = \text{fvphvvvlchlsvlljoidpafoeqbthbzxtg}$$

2.2.2. CBC–Mode (Cipher Block Chaining–Mode)

Im ECB–Modus kann Caterine sehr leicht dadurch Verwirrung stiften, dass sie die Blöcke des Chiffrats vertauscht und in eine neue Reihenfolge bringt. Ist die Nachricht nicht zusätzlich abgesichert und authentiziert, so erhält Bob eine korrekt verschlüsselte Nachricht und kann die Manipulation schwer erkennen. Das adressiert der CBC–Modus, bei dem die Verschlüsselungsschritte aufeinander aufbauen.

Vorbereitung:

1. Bob und Alice einigen sich auf einen Schlüssel k und einen Initialisierungsvektor IV .
2. Alice bestimmt das zugehörige Verschlüsselungsverfahren

$$e = e_k : \mathbb{A}^r \longrightarrow \mathbb{A}^r$$

3. Bob bestimmt das zugehörige Entschlüsselungsverfahren

$$d = d_k : \mathbb{A}^r \longrightarrow \mathbb{A}^r$$

Verschlüsselung:

1. Alice ergänzt den Klartext m so, dass seine Länge durch r teilbar ist.
2. Alice teilt m in Blöcke der Länge r auf,

$$m = m_1 \| m_2 \| \dots \| m_t$$

3. Alice setzt $y_0 = IV$.
4. Für $j = 1, \dots, t$ berechnet Alice

$$y_j = e_k(m_j + y_{j-1})$$

5. Alice schickt

$$y = y_1 \| y_2 \| \dots \| y_t$$

Entschlüsselung:

1. Bob teilt die eingegangene Nachricht y in Blöcke der Länge r auf,

$$y = y_1 \| y_2 \| \dots \| y_t$$

2. Bob setzt $y_0 = IV$.
3. Für $j = 1, \dots, t$ berechnet Bob

$$m_j = d_k(y_j) - y_{j-1}$$

4. Bob setzt

$$m = m_1 \| m_2 \| \dots \| m_t$$

Bemerkung 2.3. Der CBC–Modus hat einige Vorteile gegenüber dem ECB–Modus:

1. Klartextmuster sind am Chiffrat schwerer zu erkennen.
2. Identische Klartextblöcke ergeben unterschiedliche Geheimtextblöcke.
3. Catherine kann die Verschlüsselung schwerer analysieren, auch wenn sie Abschnitte des Klartextes kennt (*plaintext-ciphertext-attack*).

Damit verbundenen Nachteile sind

1. Das Verfahren ist aufwendiger.
2. Übertragungsfehler können sich fortpflanzen.
3. Ver– und Entschlüsselung sind nicht parallelisierbar.

Beispiel 2.3. Wir modifizieren das Vigenère–Verfahren zu einem CBC–Verfahren. Dazu betrachten wir als Beispiel (wieder) den Schlüssel

$$k = \text{KRYPTO} = (10, 17, 24, 15, 19, 14)$$

den Initialisierungsvektor

$$IV = \text{GEHEIM} = (6, 4, 7, 4, 8, 12)$$

und den Klartext

$$m = \text{Verschluesselungsverfahrensweg}$$

also in Zahlen

$$m = (21, 4, 17, 18, 2, 7, 11, 20, 4, 18, 18, 4, 11, 20, 13, 6, 18, 21, 4, 17, 5, 0, 7, 17, 4, 13, 18, 22, 4, 6)$$

Wie wir schon gesehen haben, zerfällt die Nachricht bereits in Blöcke der Länge 6,

$$\begin{aligned}m_1 &= (21, 4, 17, 18, 2, 7) \\m_2 &= (11, 20, 4, 18, 18, 4) \\m_3 &= (11, 20, 13, 6, 18, 21) \\m_4 &= (4, 17, 5, 0, 7, 17) \\m_5 &= (4, 13, 18, 22, 4, 6)\end{aligned}$$

Die Initialisierung setzt

$$y_0 = (6, 4, 7, 4, 8, 12)$$

Um die Chiffrierung übersichtlicher zu gestalten, führen wir in Schritt j den Zwischen-schritt

$$x_i = m_i + y_{i-1}$$

ein (dann ist $y_i = e(x_i) = x_i + k$) (alle Rechnungen wieder modulo 26). Damit wird die Chiffrierung zu

$$\begin{aligned}x_1 &= (21, 4, 17, 18, 2, 7) + (6, 4, 7, 4, 8, 12) &= (1, 8, 24, 22, 10, 19) \\y_1 &= (1, 8, 24, 22, 10, 19) + (10, 17, 24, 15, 19, 14) &= (11, 25, 22, 11, 3, 7) \\x_2 &= (11, 20, 4, 18, 18, 4) + (11, 25, 22, 11, 3, 7) &= (22, 19, 0, 3, 21, 11) \\y_2 &= (22, 19, 0, 3, 21, 11) + (10, 17, 24, 15, 19, 14) &= (6, 10, 24, 18, 14, 25) \\x_3 &= (11, 20, 13, 6, 18, 21) + (6, 10, 24, 18, 14, 25) &= (17, 4, 11, 24, 6, 20) \\y_3 &= (17, 4, 11, 24, 6, 20) + (10, 17, 24, 15, 19, 14) &= (1, 21, 9, 13, 25, 8) \\x_4 &= (4, 17, 5, 0, 7, 17) + (1, 21, 9, 13, 25, 8) &= (5, 12, 14, 13, 6, 25) \\y_4 &= (5, 12, 14, 13, 6, 25) + (10, 17, 24, 15, 19, 14) &= (15, 3, 12, 2, 25, 13) \\x_5 &= (4, 13, 18, 22, 4, 6) + (15, 3, 12, 2, 25, 13) &= (19, 16, 4, 24, 3, 19) \\y_5 &= (19, 16, 4, 24, 3, 19) + (10, 17, 24, 15, 19, 14) &= (3, 7, 2, 13, 22, 7)\end{aligned}$$

Damit erhalten wir hier das Chiffrat

$$y = (11, 25, 22, 11, 3, 7, 6, 10, 24, 18, 14, 25, 1, 21, 9, 13, 25, 8, 15, 3, 12, 2, 25, 13, 3, 7, 2, 13, 22, 7)$$

also

$$y = \text{lzwldhgkysoabvjnaipdmcanhcwh}$$

2.2.3. CTR–Mode (CounTeR–Mode)

Der Counter–Modus operiert ganz anders als die bisher behandelten Verfahren. Seine Sicherheit beruht auf einem für jedes Chiffrat neu zu wählendem Initialisierungsvektor IV . Dieser wird in jedem Verarbeitungsschritt mit einem Zähler verknüpft (z.B. durch

Addition oder Konkatenation, also anhängen) und verschlüsselt. Der Klartext selbst wird bei diesem Verfahren nicht unmittelbar verschlüsselt sondern mit einem verschlüsselten Vektor verknüpft.

Vorbereitung:

1. Bob und Alice einigen sich auf einen Schlüssel k und ein Inkrement inc (mit dem Defaultwert $\text{inc} = 1$), sowie auf ein Verknüpfungsverfahren v zwischen Initialisierungsvektor IV und Zähler ctr_i .
2. Bob und Alice einigen sich auf einen Initialisierungsvektor IV (der bei diesem Verfahren auch Nonce genannt wird), der nur für einen Übertragsvorgang gilt.
3. Alice bestimmt das zugehörige Verschlüsselungsverfahren

$$e = e_k : \mathbb{A}^r \longrightarrow \mathbb{A}^r$$

4. Bob bestimmt das zugehörige Entschlüsselungsverfahren

$$d = d_k : \mathbb{A}^r \longrightarrow \mathbb{A}^r$$

Verschlüsselung:

1. Alice ergänzt den Klartext m so, dass seine Länge durch r teilbar ist.
2. Alice teilt m in Blöcke der Länge r auf,

$$m = m_1 \| m_2 \| \dots \| m_t$$

3. Alice initialisiert den Zähler $\text{ctr}_0 = 0$.
4. Für $j = 1, \dots, t$ berechnet Alice

$$\begin{aligned} \text{ctr}_j &= \text{ctr}_{j-1} + \text{inc} \\ v_j &= v(IV, \text{ctr}_j) \\ y_j &= m_j + e_k(v_j) \end{aligned}$$

5. Alice schickt

$$y = y_1 \| y_2 \| \dots \| y_t$$

Entschlüsselung:

1. Bob teilt die eingegangene Nachricht y in Blöcke der Länge r auf,

$$y = y_1 \| y_2 \| \dots \| y_t$$

2. Bob initialisiert den Zähler $\text{ctr}_0 = 0$.

3. Für $j = 1, \dots, t$ berechnet Bob

$$\begin{aligned}\text{ctr}_j &= \text{ctr}_{j-1} + \text{inc} \\ v_j &= v(\text{IV}, \text{ctr}_j) \\ m_j &= y_j - e(v_j)\end{aligned}$$

4. Bob setzt

$$m = m_1 \| m_2 \| \dots \| m_t$$

Bemerkung 2.4. Der CTR–Modus hat einige Vorteile gegenüber dem ECB–Modus und dem CBC–Modus:

1. Klartextmuster sind am Chiffrat schwerer zu erkennen.
2. Identische Klartextblöcke ergeben unterschiedliche Geheimtextblöcke.
3. Catherine kann die Verschlüsselung schwerer analysieren, auch wenn sie Abschnitte des Klartextes kennt (*plaintext-ciphertext-attack*).
4. Bitfehler in der Übertragung pflanzen sich nicht fort.
5. Ver– und Entschlüsselung können gut parallelisiert werden.

Damit verbundene Nachteile sind

1. Das Verfahren ist von der Vorbereitung und der Initialisierung her aufwendiger.
2. Für jede Übertragung wird eine neue Nonce benötigt.

Bemerkung 2.5. Benutzen Alice und Bob ein binäres Alphabet, so gilt dort $-1 = 1$, Plus und Minus stimmen also überein. Deshalb ist dann Bobs letzter Verarbeitungsschritt

$$m_j = y_j + e(v_j)$$

Damit laufen in diesem Fall Ver– und Entschlüsselung vollkommen parallel.

Beispiel 2.4. Wir modifizieren das Vigenère–Verfahren zu einem CTR–Verfahren. Dazu betrachten wir als Beispiel (wieder) den Schlüssel

$$k = \text{KRYPTO} = (10, 17, 24, 15, 19, 14)$$

und das Inkrement

$$\text{inc} = 7$$

Nonce IV und Zähler ctr_j werden wie folgt verknüpft:

Wir betrachten den Initialisierungsvektor IV als Zahl im 26-er System und rechnen IV in eine ganze Zahl um. Dann werden IV und ctr_j addiert (als Zahlen) und wieder als Zahl zur Basis 26 dargestellt.

Für die Übertragung des Klartextes

$$m = \text{Verschlüsselungsverfahrensweg}$$

wählen Alice und Bob die Nonce

$$IV = \text{GEHEIM} = (6, 4, 7, 4, 8, 12)$$

was der Zahl

$$IV = 6 \cdot 26^5 + 4 \cdot 26^4 + 7 \cdot 26^3 + 4 \cdot 26^2 + 8 \cdot 26^1 + 12 \cdot 26^0 = 48\,192\,814\,916$$

entspricht. Der Klartext in Zahlen schreibt sich wieder als

$$m = (21, 4, 17, 18, 2, 7, 11, 20, 4, 18, 18, 4, 11, 20, 13, 6, 18, 21, 4, 17, 5, 0, 7, 17, 4, 13, 18, 22, 4, 6)$$

Wie wir schon gesehen haben, zerfällt die Nachricht bereits in Blöcke der Länge 6,

$$\begin{aligned} m_1 &= (21, 4, 17, 18, 2, 7) \\ m_2 &= (11, 20, 4, 18, 18, 4) \\ m_3 &= (11, 20, 13, 6, 18, 21) \\ m_4 &= (4, 17, 5, 0, 7, 17) \\ m_5 &= (4, 13, 18, 22, 4, 6) \end{aligned}$$

Die Initialisierung setzt

$$\text{ctr}_0 = 0$$

Um die Chiffrierung übersichtlicher zu gestalten, führen wir in Schritt j die Zwischen-schritte

$$\text{ctr}_i = \text{ctr}_{i-1} + \text{inc}$$

(in \mathbb{Z}) und

$$\begin{aligned} v_j &= IV + \text{ctr}_j = [IV + \text{ctr}_j]_{26} \\ w_j &= e(v_j) = v_j + k \end{aligned}$$

ein (wobei $[z]_{26}$ für die Darstellung einer Zahl zur Basis 26 steht). Damit wird die Chif-frierung zu

ctr_1	$=$	$ctr_0 + 7$	$=$	7
v_1	$=$	48 192 814 923	$=$	(6, 4, 7, 4, 8, 19)
w_1	$=$	(6, 4, 7, 4, 8, 19) + (10, 17, 24, 15, 19, 14)	$=$	(16, 21, 5, 19, 1, 7)
y_1	$=$	(21, 4, 17, 18, 2, 7) + (16, 21, 5, 19, 1, 7)	$=$	(11, 25, 22, 11, 3, 14)
ctr_2	$=$	$ctr_1 + 7$	$=$	14
v_2	$=$	48 192 814 930	$=$	(6, 4, 7, 4, 9, 0)
w_2	$=$	(6, 4, 7, 4, 9, 0) + (10, 17, 24, 15, 19, 14)	$=$	(16, 21, 5, 19, 2, 14)
y_2	$=$	(11, 20, 4, 18, 18, 4) + (16, 21, 5, 19, 2, 14)	$=$	(1, 15, 9, 11, 20, 18)
ctr_3	$=$	$ctr_2 + 7$	$=$	21
v_3	$=$	48 192 814 937	$=$	(6, 4, 7, 4, 9, 7)
w_3	$=$	(6, 4, 7, 4, 9, 7) + (10, 17, 24, 15, 19, 14)	$=$	(16, 21, 5, 19, 2, 21)
y_3	$=$	(11, 20, 13, 6, 18, 21) + (16, 21, 5, 19, 2, 21)	$=$	(1, 15, 18, 25, 20, 16)
ctr_4	$=$	$ctr_3 + 7$	$=$	28
v_4	$=$	48 192 814 944	$=$	(6, 4, 7, 4, 9, 14)
w_4	$=$	(6, 4, 7, 4, 9, 14) + (10, 17, 24, 15, 19, 14)	$=$	(16, 21, 5, 19, 2, 2)
y_4	$=$	(4, 17, 5, 0, 7, 17) + (16, 21, 5, 19, 1, 2)	$=$	(20, 12, 10, 19, 9, 19)
ctr_5	$=$	$ctr_4 + 7$	$=$	35
v_5	$=$	48 192 814 951 = (6, 4, 7, 4, 9, 21)		
w_5	$=$	(6, 4, 7, 4, 9, 21) + (10, 17, 24, 15, 19, 14)	$=$	(16, 21, 5, 19, 2, 9)
y_5	$=$	(4, 13, 18, 22, 4, 6) + (16, 21, 5, 19, 2, 9)	$=$	(20, 8, 23, 15, 6, 15)

und wir erhalten hier das Chiffrat

$$y = (11, 25, 22, 11, 3, 14, 1, 15, 9, 11, 20, 18, 1, 15, 18, 25, 20, 16, 20, 12, 10, 19, 9, 19, 20, 8, 23, 15, 6, 15)$$

also

$$y = \text{lawldobpjplusbpsauqumktjtuixpgp}$$

2.3. Stromchiffrierung

Wir beschränken uns in diesem Abschnitt auf den Fall $\mathbb{A} = \{0, 1\} = \mathbb{F}_2$. Bei einer Stromchiffrierung kann ein Klartext beliebiger Länge verschlüsselt werden. Der Klartext ist in diesem Fall gegeben durch einen Bitstrom

$$m = (m_1, m_2, m_3, \dots) \in \mathbb{F}_2^*$$

wobei \mathbb{F}_2^* dafür steht, dass der Bitstrom beliebig lang sein kann. Der Schlüssel ist gegeben durch einen Schlüsselstrom

$$k = (k_1, k_2, k_3, \dots) \in \mathbb{F}_2^*$$

Die Verschlüsselung erfolgt in diesem Fall Bit für Bit und Alice erzeugt einen Chiffrastrom

$$y = (y_1, y_2, y_3, \dots) \in \mathbb{F}_2^*$$

mit $y_i = e_k(m_i) = m_i + k_i \bmod 2 = x_i \oplus k_i$.

Die Entschlüsselung erfolgt ebenfalls Bit für Bit und Bob rekonstruiert den Nachrichtenstrom m via

$$m_i = d_k(y_i) = y_i + k_i \bmod 2 = y_i \oplus k_i$$

Da $k_i \oplus k_i = 0$, gilt in der Tat

$$d_k(e_k(m_i)) = d_k(m_i \oplus k_i) = (m_i \oplus k_i) \oplus k_i = m_i$$

Der Operator \oplus ist der XOR-Operator aus der theoretischen Informatik, der komponentenweise auf den Bits wie folgt operiert,

\oplus	0	1
0	0	1
1	1	0

der also der Addition im Körper \mathbb{F}_2 entspricht. Wie in der Mathematik werden wir daher für \oplus auch in dieser Vorlesung $+$ schreiben und betrachten $\oplus = +$ als Addition von Vektoren in \mathbb{F}_2^n .

Die Sicherheit der Stromchiffrierung hängt wesentlich vom Schlüssel k ab. Am besten sind Schlüssel ohne erkennbares Muster und ohne nachvollziehbarer Struktur (echte Zufallsfolgen). Echte Zufallszahlen bzw. Folgen von Zufallszahlen können jedoch nicht algorithmisch erzeugt werden. Deshalb benutzt man in der Praxis Pseudozufallszahlengeneratoren (PRNG für **Pseudo Random Number Generator**), die mit einem Samen

$$s = (s_1, s_2, \dots, s_t) \in \mathbb{F}_2^t$$

und einer Erzeugerfunktion f arbeiten.

Definition 2.2. Ein Pseudozufallszahlengenerator besteht aus einer Erzeugerfunktion

$$f : \mathbb{F}_2^t \longrightarrow \mathbb{F}_2$$

die aus einem Samen

$$s = (s_1, s_2, \dots, s_t) \in \mathbb{F}_2^{t+1}$$

einen Schlüsselstrom k wie folgt erzeugt:

Für $i = 1, \dots, t$:

$$k_i = s_i$$

Für $i \geq t + 1$:

$$k_i = f(k_{i-t}, k_{i-t+1}, \dots, k_{i-1})$$

Das bekannteste Verfahren ist das linear-rückgekoppelte Schieberegister **LFSR** (Linear Feedback Shift Register). In diesem Fall ist f gegeben durch ein t -Tupel

$$f = (f_1, \dots, f_t) \in \mathbb{F}_2^t$$

mit $f_t \neq 0$, und f definiert durch

$$f(a_1, a_2, \dots, a_t) = \sum_{i=1}^t f_i \cdot a_i$$

Beispiel 2.5. Das LFSR mit $t = 5$, $f = (1, 0, 1, 0, 1)$ und $s = (1, 1, 0, 0, 0)$ erzeugt den Schlüssel

$$\begin{aligned} k = & (1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, \\ & 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, \\ & 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, \\ & 1, 1, 0, 0, 0, 1, 0, 0, \dots) \end{aligned}$$

Insbesondere ist k periodisch mit einer Periode der Länge 15.

In der Tat werden alle durch ein LFSR erzeugten Schlüssel irgendwann periodisch (und sind schon deshalb keine echten Zufallsfolgen).

Regel 2.1. Ein von einem linear-rückgekoppeltem Schieberegister mit $s, f \in \mathbb{F}_2^t$ erzeugter Schlüssel k wird periodisch mit Periodenlänge $p \leq 2^t - 1$.

Beweis: Wir betrachten die Teilabschnitte $K_i = (k_i, \dots, k_{i+t-1})$ des erzeugten Schlüssels k . Falls $s = 0$, also $K_1 = 0$, so ist auch k der 0-Strom und damit periodische von jeder Periode. Andernfalls ist $K_i \neq 0$ für jedes $i \geq 1$ (wegen $f_t \neq 0$). Da außerdem $K_i \in \mathbb{F}_2^t$, gibt es insgesamt $2^t - 1$ verschiedene Möglichkeiten für K_i , dh. spätestens nach $2^t - 1$ Schritten kommt es zu einer Wiederholung $K_i = K_j$ ($1 \leq i < j \leq 2^t$), also

$$(k_i, k_{i+1}, \dots, k_{i+t-1}) = (k_j, k_{j+1}, \dots, k_{j+t-1})$$

Dann gilt aber

$$k_{i+t} = f(k_i, k_{i+1}, \dots, k_{i+t-1}) = f(k_j, k_{j+1}, \dots, k_{j+t-1}) = k_{j+t}$$

also $K_{i+1} = K_{j+1}$ usw. Damit wiederholt sich ab dieser Stelle alles im Abstand $j - i$ und daher wird k periodisch mit Periodenlänge $p = j - i \leq 2^t - 1$.

Falls $f_1 = 1$ kann man sogar zeigen, dass die gesamte Schlüsselfolge periodisch ist.

Definition 2.3. Ist die Erzeugerfunktion f eines LSFR gegeben durch $f = (f_1, \dots, f_t)$, so heißt

$$f(X) = 1 + \sum_{i=1}^t f_i \cdot X^i$$

Erzeugerpolynom des LSFR.

Beispiel 2.6. Das LFSR mit $t = 5$ und $f = (1, 0, 1, 0, 1)$ hat das Erzeugerpolynom

$$f(X) = 1 + X + X^3 + X^5$$

Bemerkung 2.6. Man kann zeigen, dass ein LSFR-erzeugter Schlüsselstrom mit $f \in \mathbb{F}_2^t$ genau dann periodisch von der (primitiven) Periode $2^t - 1$ wird, wenn das Erzeugerpolynom $f(X)$ primitiv ist, also eine Relation von \mathbb{F}_{2^t} definiert.

Insbesondere gibt es für jedes t ein LSFR mit Erzeugerpolynom $f(X)$ vom Grad t , sodass der erzeugte Schlüsselstrom die primitive Periode $2^t - 1$ hat.

Bemerkung 2.7. Im ersten Sicherheitsstandard des Mobilfunkstandards GSM wurde die 1987 entwickelte Verschlüsselungstechnologie A5/1 eingesetzt, die den Schlüsselstrom aus einer Kombination von drei LSFR erzeugt, und zwar mit den Erzeugerpolynomen

$$\begin{aligned} f_1(X) &= X^{19} + X^{18} + X^{17} + X^{14} + 1, \\ f_2(X) &= X^{22} + X^{21} + 1 \\ f_3(X) &= X^{23} + X^{22} + X^{21} + X^8 + 1 \end{aligned}$$

Alle drei Erzeugerpolynome sind primitiv, erzeugen also für sich Schlüsselströme mit maximal möglicher Periodenlänge.

Diese Verschlüsselungstechnologie ist allerdings nicht mehr sicher und kann gegenwärtig in Realzeit gebrochen werden.

Aktuell wird der Blockchiffrenalgorithmus A5/3 verwendet. Dieses Verfahren gilt zwar inzwischen theoretisch ebenfalls als angreifbar, ein praktisch durchführbarer Angriff ist allerdings nicht bekannt.

2.4. Asymmetrische Verschlüsselung

Die in Abschnitt 2.1 behandelten Verschlüsselungsverfahren zeichnen sich dadurch aus, dass es **einen** Schlüssel gibt, der sowohl für die Ver- als auch für die Entschlüsselung benutzt wird. So wird etwa bei der Stromchiffrierung der Schlüssel zum Nachrichtenstrom addiert, und bei der Entschlüsselung wird wieder der Schlüsselstrom zum Chiffarat addiert.

Der Umkehralgorithmus d zum Entschlüsseln von e lässt sich also sofort aus der Kenntnis des prinzipiellen Verfahrens und des Schlüssels k ableiten. Deshalb ist es für diese Verfahren auch essentiell, dass der Schlüssel k geheim bleibt und nur den beiden kommunizierenden Parteien bekannt ist.

Anders aufgebaut sind die sogenannten asymmetrischen Verschlüsselungsverfahren. Diese sind so konstruiert, dass sich der Entschlüsselungsalgorithmus d nicht (oder zumindest nicht in einer offensichtlichen Weise) aus der Kenntnis des Verschlüsselungsalgorithmus e und des Schlüssels k , der zum Verschlüsseln benutzt wird, ermitteln lässt.

Beispiel 2.7.

- a) Betrachte $e =$ Potenzierung, $k =$ die Potenz, zu der erhoben wird, d.h. der Text m wird in Form einer Zahl angegeben und

$$e_k(m) = m^k =: y$$

Dann gilt

$$d_k(y) = \sqrt[k]{y}$$

In vielen Zahlsystemen kann das nicht (in kontrollierter Zeit) aus der Kenntnis von y und k ermittelt werden.

- b) Betrachte $e =$ Potenzierung, $k =$ die Basis, die bei der Potenzierung verwendet wird, d.h. der Text m wird in Form einer Zahl angegeben und

$$e_k(m) = k^m =: y$$

Dann gilt

$$d_k(y) = \log_k(y)$$

In vielen Zahlsystemen kann das nicht (in kontrollierter Zeit) aus der Kenntnis von y und k ermittelt werden.

Da der Empfänger der Nachricht diese natürlich wieder in Klartext umwandeln muss, muss es bei den für asymmetrische Verfahren in Betracht kommenden Methoden aber möglich sein, mithilfe geeigneter Zusatzinformation p (die von e und k abhängt aber nicht unmittelbar aus der Kenntnis von e und k gewonnen werden kann) e wieder umzukehren und m zurückzugewinnen,

$$m = d_p(y)$$

Der „Schlüssel“ bei einem asymmetrischen Verfahren besteht also aus einem Schlüsselpaar (k, p) wobei p nicht aus k berechenbar ist.

1. Die Verschlüsselung benötigt nur den Schlüssel k ,

$$y = e_k(m)$$

2. Die Entschlüsselung benötigt nur den Schlüssel p ,

$$m = d_p(y)$$

Definition 2.4. Der Schlüssel k heißt der **öffentliche Schlüssel**, der Schlüssel p heißt der **private Schlüssel** des Verfahrens.

Bemerkung 2.8. Für den öffentlichen Schlüssel k werden wir oft k_{pub} schreiben, für den privaten Schlüssel p dann k_{priv} .

Bemerkung 2.9. Bei einem asymmetrischen Verfahren muss der öffentliche Schlüssel k nicht geheim gehalten werden. Wichtig ist aber, dass der private Schlüssel p geheim bleibt.

2.5. Kryptoanalyse

Die Kryptoanalyse beschäftigt sich damit, Kryptosysteme zu brechen, dh. Methoden zu finden, um einen Verschlüsselungsalgorithmus

$$y = e_k(m)$$

in algorithmischer Form und in überschaubarer Zeit nach m aufzulösen. Zu beachten ist dabei immer das sogenannten Kerckhoffsche Prinzip:

Prinzip (Kerckhoffsches Prinzip). *Ein kryptographisches System muss so konstruiert sein, dass seine Sicherheit nicht von der Geheimhaltung des Verschlüsselungssystems als solches sondern nur von der Geheimhaltung des Schlüssels abhängt.*

In der Kryptoanalyse arbeitet man daher mit den folgenden Grundannahmen:

1. Der Angreifer kennt das verwendete Verschlüsselungsverfahren.
2. Der Angreifer hat Zugang zu den chiffrierten Texten.
3. Der Schlüsselaustausch ist sicher, der Angreifer kann sich also keinen Zugang zu dem Schlüssel während des Schlüsselaustauschs verschaffen.
4. Die Schlüsselaufbewahrung ist sicher, der Angreifer hat also keinen Zugang zu dem geheim aufbewahrten Schlüssel.

Bei Angriffen auf ein kryptographisches System werden vor allem folgende Typen betrachtet:

ciphertext–only–Angriffe (passive Angreifer)

Catherine kennt chiffrierte Texte. Von dieser Situation muss ausgegangen werden, denn der von Alice und Bob benutzte Übertragungskanal ist unsicher.

Mögliche Ciphertext–only–Angriffe:

1. Vollständige Schlüsselsuche:

Der Angreifer probiert alle möglichen Schlüssel aus (brute force–Methode). Unter den wenigen sinnvollen Texten, die sich dabei aus den bekannten Chiffren ergeben, befindet sich der Klartext. Das funktioniert sehr gut, wenn der Schlüsselraum klein genug ist (wie z.B. bei der Caesar–Verschlüsselung). Alice und Bob können sich dagegen mit einem Verfahren mit einem hinreichend großen Schlüsselraum wehren. Ein Schlüsselraum mit 2^{112} Schlüsseln gilt momentan als sicher bis 2030.

2. Statistische Analysen:

Catherine kann Häufigkeiten bestimmter Buchstaben und Buchstabenkombinationen analysieren, um daraus auf den Schlüssel zu schließen (wie etwa bei der monoalphabetischen Substitution oder beim Vigenère–Verfahren). Alice und Bob können sich dagegen mit einem Verfahren wehren, das es schwer macht, Muster zu erkennen.

known–plaintext–Angriffe

Catherine kennt in diesem Fall einige Klartexte und die zugehörigen verschlüsselten Texte. Das kann z.B. der Fall sein, wenn Catherine weiß, dass an bestimmten Stellen des Textes immer spezielle Textpassagen, z.B. Namen, Absenderangaben oder Bezeichnungen stehen, oder wenn sie über andere Quellen an Klartextabschnitte der übertragenen Nachricht kommt. In diesem Fall kann Catherine versuchen, einen mathematischen Angriff auf das Verfahren zu starten. Ist der Algorithmus, der hinter dem Verschlüsselungsverfahren steckt, zu einfach aufgebaut (z.B. zu nahe an einem linearen Verfahren), so ist das häufig möglich.

chosen–plaintext–Angriffe

Catherine hat in diesem Fall die Möglichkeit, einen Klartext selbst zu wählen und hierzu das Chiffrat zu bekommen, z.B. indem sie Zugang zum Chiffriergerät bekommt oder indem sie die Möglichkeit hat, Texte ihrer Wahl in das System einzuschleusen. Möglich ist auch, dass Catherine durch eine eigene Aktion (die nichts mit der Verschlüsselung zu tun hat) das Senden einer Nachricht erzwingen kann.

chosen-ciphertext-Angriffe

Catherine hat in diesem Fall die Möglichkeit, ein Chiffrat selbst zu wählen und hierzu den Klartext zu bekommen, z.B. indem sie Zugang zum Dechiffriergerät bekommt, oder wenn sie jemanden kennt, der diesen Zugang hat.

Die letzten beiden Angriffe sind zwar relativ schwer zu realisieren, trotzdem muss man damit rechnen und ein verlässliches kryptographisches System sollte dagegen geschützt sein.

Definition 2.5. Ein kryptographisches Verfahren heißt beweisbar sicher, wenn es auch dann nicht gebrochen werden kann, wenn dem Angreifer beliebige Rechenleistung zur Verfügung steht.

Definition 2.6. Eine Stromchiffrierung mit Schlüsselstrom k , bei der

1. der Schlüsselstrom $k = (k_1, k_2, k_3, \dots)$ durch einen echten Zufallsgenerator erzeugt wird,
2. der Schlüsselstrom nur Alice und Bob bekannt ist,
3. jedes Schlüsselstrombit k_j nur für die Verschlüsselung eines einzigen Klartextbits m_j verwendet wird

heißt **One-Time-Pad (OTP)**.

Satz 2.2. Ein One-Time-Pad ist beweisbar sicher.

Bemerkung 2.10. Bei diesem Satz ist wichtig, dass der Schlüsselstrom nur einmal verwendet wird. Wird ein Schlüsselstrom zweimal verwendet um zwei Nachrichten m und n zu verschlüsseln, so erhalten wir zwei Chiffrate

$$y = m + k, \quad z = n + k$$

woraus

$$y + z = (m + k) + (n + k) = m + n + k + k = m + n$$

folgt. Aus der Kenntnis von $m + n$ kann Catherine in vielen Fällen auf n und m rückschließen, da es oft nicht viele sinnvolle Nachrichten gibt, deren Summe einen gegebenen Bitstrom ergibt.

Bemerkung 2.11. Wichtig in dem Satz ist auch, dass es sich bei k um einen echten und unendlich langen Zufallsstrom handelt. Kein System mit Schlüsseln fester endlicher Länge oder mit Schlüsseln unendlicher Länge, die sich mittels einer Erzeugerfunktion aus endlichen Samen einer festen Länge berechnen lassen, ist beweisbar sicher.

known-plaintext-Angriff bei einem LFSR

Wir betrachten ein LFSR mit einer Erzeugerfunktion $f : \mathbb{A}^t \rightarrow \mathbb{A}$, gegeben durch $f = (f_1, \dots, f_t)$ und einen Samen $s = (s_1, \dots, s_t)$, sodass also

$$\begin{aligned} k_i &= s_i && \text{für } i = 1, \dots, t \\ k_{i+t} &= f_1 \cdot k_i + f_2 \cdot k_{i+1} + \dots + f_t \cdot k_{i+t} && \text{für } i \geq 0 \end{aligned}$$

Wir gehen davon aus, dass Catherine die Länge t des Samens (und der Erzeugerfunktion) kennt und dass ihr ein Klartext–Ciphertext–Paar $m = (m_1, \dots, m_l)$, $y = (y_1, \dots, y_l)$ der Länge $l \geq 2t$ bekannt ist. Damit erhält Catherine aus

$$y_i = m_i + k_i$$

zunächst

$$k_i = y_i + m_i \quad \text{für } i = 1, \dots, m$$

und daher kennt sie k_i für alle $i \in \{0, \dots, m\}$.

Ferner weiß sie, dass

$$\begin{aligned} k_1 \cdot f_1 + k_2 \cdot f_2 + \dots + k_t \cdot f_t &= k_{t+1} \\ k_2 \cdot f_1 + k_3 \cdot f_2 + \dots + k_{t+1} \cdot f_t &= k_{t+2} \\ &\dots \\ k_{l-t} \cdot f_1 + k_{l-t+1} \cdot f_2 + \dots + k_{l-1} \cdot f_t &= k_l \end{aligned}$$

(mit ihr noch unbekannten f_1, \dots, f_t , wohingegen sie ja k_1, \dots, k_l kennt) gilt. Da $l \geq 2t$ hat Catherine ein lineares Gleichungssystem mit mindestens t Gleichungen in den t Unbekannten f_1, \dots, f_t . Dieses Gleichungssystem kann nun mit Mitteln der linearen Algebra gelöst werden. Falls es eindeutig lösbar ist, sind f_1, \dots, f_t damit bestimmt, andernfalls zumindest stark eingegrenzt.

Wir nehmen nun an, dass $\mathbb{A} = \mathbb{F}_q$ ein endlicher Körper mit q Elementen ist.

Definition 2.7. Ein Blockverschlüsselungsverfahren der Länge r heißt affin, wenn es (zu einem gegebenen Schlüssel k) eine invertierbare $r \times r$ -Matrix U mit Koeffizienten in \mathbb{F}_q und einen Vektor $\vec{v} \in \mathbb{F}_q^r$ gibt mit

$$y = e_k(m) = U \cdot \vec{m} + \vec{v}$$

Bemerkung 2.12. Allgemeiner reicht es, als Alphabet einen endlichen Ring zu betrachten; wichtig ist aber, dass die Matrix U invertierbar ist.

Bemerkung 2.13. Der Entschlüsselungsalgorithmus zu einem affinen Verfahren mit

$$y = e_k(m) = U \cdot \vec{m} + \vec{v}$$

ist gegeben durch

$$m = d_k(y) = U^{-1} \cdot (\vec{y} - \vec{v})$$

denn es gilt

$$\begin{aligned} d_k(e_k(m)) &= d_k(U \cdot \vec{m} + \vec{v}) \\ &= U^{-1} \cdot ((U \cdot \vec{m} + \vec{v}) - \vec{v}) \\ &= U^{-1} \cdot U \cdot \vec{m} + U^{-1} \cdot \vec{v} - U^{-1} \cdot \vec{v} \\ &= U^{-1} \cdot U \cdot \vec{m} \\ &= \vec{m} \end{aligned}$$

Beispiel 2.8. Das Vigenère–Verfahren ist affin.

Bemerkung 2.14. Zu $m = (m_1, \dots, m_r) \in \mathbb{F}_q^n$ bezeichnet \vec{m} den zugehörigen Spaltenvektor, dh.

$$\vec{m} = \begin{pmatrix} m_1 \\ \vdots \\ m_r \end{pmatrix}$$

Bemerkung 2.15. Der Schlüsselraum der affinen Verschlüsselungsverfahren ist $\mathrm{Gl}_r(\mathbb{F}_q) \times \mathbb{F}_q^n$. Ein Schlüssel besteht aus einem Paar $k = (U, \vec{v})$ mit einer invertierbaren Matrix U und einem Vektor $\vec{v} \in \mathbb{F}_q^r$.

Das Entschlüsselungsverfahren zu diesem Schlüssel k ist gegeben durch

$$d_k(y) = U^{-1} \cdot (\vec{y} - \vec{v}) = U^{-1} \cdot \vec{y} - U^{-1} \cdot \vec{v}$$

Satz 2.3. Sind $(m_0, y_0), (m_1, y_1), \dots, (m_r, y_r)$ Klartext–Ciphertext–Paare der Länge r , und sind $\vec{m}_1 - \vec{m}_0, \vec{m}_2 - \vec{m}_0, \dots, \vec{m}_r - \vec{m}_0 \in \mathbb{F}_q^r$ linear unabhängig, so kann aus diesen Daten der Schlüssel $k = (U, \vec{v})$ berechnet werden.

Beweis: Beachten Sie zunächst, dass

$$\begin{aligned} \vec{y}_j - \vec{y}_0 &= U \cdot \vec{m}_j + \vec{v} - (U \cdot \vec{m}_0 + \vec{v}) \\ &= U \cdot \vec{m}_j - U \cdot \vec{m}_0 \\ &= U \cdot (\vec{m}_j - \vec{m}_0) \end{aligned}$$

Setze $\vec{g}_j = \vec{m}_j - \vec{m}_0$ und $\vec{h}_j = \vec{y}_j - \vec{y}_0$. Dann ist $\vec{g}_1, \dots, \vec{g}_r$ eine Basis von \mathbb{F}_q^r , und daher ist die Matrix $A = (\vec{g}_1 \quad \dots \quad \vec{g}_r)$ mit $\vec{g}_1, \dots, \vec{g}_r$ als Spalten eine invertierbare Matrix. Ist nun $M = (\vec{h}_1 \quad \dots \quad \vec{h}_r)$ die Matrix mit $\vec{h}_1, \dots, \vec{h}_r$ als Spalten, so gilt

$$U = M \cdot A^{-1}$$

denn für alle $j \in \{1, \dots, r\}$ gilt

$$M \cdot A^{-1} \cdot \overrightarrow{g_j} = M \cdot \overrightarrow{e_j} = \overrightarrow{h_j} = U \cdot \overrightarrow{g_j}$$

Daher ist zunächst U bestimmt. Aus

$$\overrightarrow{v} = U \cdot \overrightarrow{m_0} + \overrightarrow{v} - U \cdot \overrightarrow{m_0} = \overrightarrow{y_0} - U \cdot \overrightarrow{m_0}$$

erhalten wir nun auch \overrightarrow{v} .

Beispiel 2.9. Für ein affines Chiffrierungsverfahren $e : \mathbb{F}_2^3 \longrightarrow \mathbb{F}_2^3$ gilt

$$\begin{aligned} e(1, 1, 0) &= (0, 0, 0) \\ e(1, 0, 1) &= (1, 0, 1) \\ e(0, 1, 1) &= (1, 1, 1) \\ e(1, 1, 1) &= (1, 1, 0) \end{aligned}$$

Mit $m_0 = (1, 1, 1)$, $m_1 = (0, 1, 1)$, $m_2 = (1, 0, 1)$ und $m_3 = (1, 1, 0)$ haben wir hier, dass

$$m_1 - m_0 = g_1, \quad m_2 - m_0 = g_2, \quad m_3 - m_0 = g_3$$

die Standardbasisvektoren von \mathbb{F}_2^3 sind. Hierfür gilt nun

$$\begin{aligned} U \cdot \overrightarrow{g_1} &= e(m_1) - e(m_0) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\ U \cdot \overrightarrow{g_2} &= e(m_2) - e(m_0) = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \\ U \cdot \overrightarrow{g_3} &= e(m_3) - e(m_0) = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \end{aligned}$$

und damit ist

$$U = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

und es ist

$$\overrightarrow{b} = U \cdot \overrightarrow{m_1} - e(m_1) = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

Damit ist das Verfahren geknackt.

Folgerung 2.4. *Jedes affin-lineare Verschlüsselungsverfahren ist durch hinreichend viele generische Klartext-Ciphertext-Paare entschlüsselbar.*

Damit ist auch gezeigt, dass ein modernes, verlässliches Verschlüsselungsverfahren nicht-lineare Komponenten enthalten muss.

Einige Anforderungen die sich daraus für Verschlüsselungsverfahren ergeben:

Konfusion

Zwischen Klartext und Chiffrat sollen keine Beziehungen erkennbar sein. Insbesondere sollen gleiche Änderungen an unterschiedlichen Klartexten zu unterschiedlichen Änderungen der Chiffre führen.

Diffusion

Der Einfluss eines Zeichens des Klartexts soll sich auf möglichst viele Zeichen des Chiffrets erstrecken, dh. wird ein Zeichen im Klartext geändert, so sollen sich viele Zeichen im Chiffret ändern.

3. Data Encryption Standard DES

3.1. Feistel–Chiffrierung

Der **Data Encryption Standard (DES)** war bis zur Jahrtausendwende das führende Verfahren zur Verschlüsselung im digitalen Datenverkehr. Initiiert 1972 vom NBS (National Bureau of Standards, heute NIST, National Institute of Standards and Technology), ist DES eine Blockchiffrierung

$$\text{DES}_k : \mathbb{F}_2^{64} \longrightarrow \mathbb{F}_2^{64}$$

wobei $k \in \mathbb{F}_2^{64}$ ein 64–Bitschlüssel ist, effektiv allerdings nur 56 Bit lang, da jedes achte Bit ein Paritätsprüfbet ist. Wir arbeiten hier also nur mit Binärdaten.

Das DES–Verfahren besteht aus insgesamt 18 Schritten:

- Aus einer Eingangspermutation IP .
- Aus 16 Verschlüsselungsrunden.
- Aus einer Ausgangspermutation $FP = IP^{-1}$.

Dabei funktionieren alle 16 Verschlüsselungsrunden nach dem gleichen Muster, benutzen jedoch jeweils unterschiedliche Rundenschlüssel $k^{(1)}, \dots, k^{(16)}$, die nach einem festen Schlüsselfahrplan aus dem Schlüssel k abgeleitet werden.

Diese 16 Verschlüsselungsrunden sind nach dem Prinzip eines **Feistel–Netzwerks** aufgebaut. Dazu wird in jeder Runde i der eingehende 64–Bit–Datenstrom x_i in zwei gleichlange 32–Bit–Anteile aufgeteilt,

$$x_i = L_i \| R_i$$

und so verarbeitet, dass an die $i + 1$ –Runde (bzw. an die Ausgangspermutation) ein Datenstrom

$$x_{i+1} = L_{i+1} \| R_{i+1}$$

weitergegeben wird, wobei

$$\begin{aligned} L_{i+1} &= R_i \\ R_{i+1} &= L_i + f(R_i, k^{(i)}) \end{aligned}$$

Dieses Verfahren hat den großen Vorteil, dass es sich (bei Kenntnis der Rundenschlüssel) sehr leicht umkehren lässt. Da $f(R_i, k^{(i)}) + f(R_i, k^{(i)}) = 0$, gilt nämlich

$$\begin{aligned} R_i &= L_{i+1} \\ L_i = R_{i+1} + f(R_i, k^{(i)}) &= R_{i+1} + f(L_{i+1}, k^{(i)}) \end{aligned}$$

was die Entschlüsselung (für eine Besitzer des Schlüssels) stark vereinfacht.

Bemerkung 3.1. Die Eingangs- und die Ausgangspermutation sind fest vorgegeben und allgemein bekannt. Sie tragen nicht zur Sicherheit der Verschlüsselung bei und es ist nicht bekannt, welche Rolle sie bei dem Verfahren spielen. Es wird allerdings vermutet, dass sie auf die Prozessortechnologie der Zeit der Einführung von DES Bezug nehmen und die prozedurale Verarbeitung vereinfachen sollen.

3.2. Die S -Boxen

Eine spezielle Rolle in der DES-Verarbeitung (und auch in jedem anderen bekannten Blockchiffierungsverfahren) bilden Zwischenschritte, bei denen Abschnitte des zu verschlüsselnden Textes gegen andere Abschnitte und Passagen ausgetauscht werden, und zwar nach Regeln, die keine klare Struktur erkennen lassen (speziell keine lineare Struktur) und nicht durch (einfache) Formeln beschrieben werden können (die sich unter Umständen auflösen und rückwärtsrechnen lassen). Sie dienen vor allem der Konfusion, sorgen also dafür, dass die Beziehung zwischen Schlüssel und Chiffraut verschleiert wird. Bei DES übernehmen diesen Part die acht S -Boxen $S_1 \dots, S_8$. Das sind Abbildungen

$$S_j : \mathbb{F}_2^6 = \mathbb{F}_2^2 \times F_2^4 \longrightarrow F_2^4$$

die wie folgt aufgebaut sind:

Zunächst wird jedes $x = (x_1, x_2, x_3, x_4, x_5, x_6) \in \mathbb{F}_2^6$ in einen Randteil a und einen Mittelteil b wie folgt zerlegt

$$a = (x_1, x_6), \quad b = (x_2, x_3, x_4, x_5)$$

Dann werden Rand- und Mittelteil als Binärdarstellungen natürlicher Zahlen betrachtet, dh.

$$a = x_1 \cdot 2^1 + x_6 \cdot 2^0, \quad b = x_2 \cdot 2^3 + x_3 \cdot 2^2 + x_4 \cdot 2^1 + x_5 \cdot 2^0$$

Die Abbildung S_1 wird dann durch die folgende Tabelle beschrieben:

S_1	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

wobei diese Tabelle so zu lesen ist, dass ein Element (a, b) auf das Element c aus dieser Tabelle mit Zeilenindex a und Spaltenindex b abzubilden und dieses dann in seine Binärdarstellung aus \mathbb{F}_2^4 zu übersetzen ist.

Beispiel 3.1. Das Element $x = (1, 0, 1, 1, 0, 1) \in \mathbb{F}_2^6$ zerfällt in den äußeren Anteil

$$a = (x_1, x_6) = (1, 1) = 1 \cdot 2^1 + 1 \cdot 2^0 = 3$$

und den mittleren Anteil

$$b = (x_2, x_3, x_4, x_5) = 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 06$$

und damit gilt

$$S_1(x) = S(3, 06) = 01 = (0, 0, 0, 1)$$

Beispiel 3.2. Für das Element $x = (1, 1, 1, 0, 1, 0)$ gilt

$$a = (1, 0) = 2, \quad b = (1, 1, 0, 1) = 13$$

und damit

$$S_1(x) = S_1(2, 13) = 10 = (1, 0, 1, 0)$$

Die S -Boxen S_2, \dots, S_8 sind nach dem gleichen Grundprinzip aufgebaut, wobei hier die folgenden Abbildungstabellen zugrunde liegen:

S_2 -Box

S_2	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	15	01	08	14	06	11	03	04	09	07	02	13	12	00	05	10
1	03	13	04	07	15	02	08	14	12	00	01	10	06	09	11	05
2	00	14	07	11	10	04	13	01	05	08	12	06	09	03	02	15
3	13	08	10	01	03	15	04	02	11	06	07	12	00	05	14	09

S_3 -Box

S_3	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	10	00	09	14	06	03	15	05	01	13	12	07	11	04	02	08
1	13	07	00	09	03	04	06	10	02	08	05	14	12	11	15	01
2	13	06	04	09	08	15	03	00	11	01	02	12	05	10	14	07
3	01	10	13	00	06	09	08	07	04	15	14	03	11	05	02	12

S_4 -Box

S_4	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	07	13	14	03	00	06	09	10	01	02	08	05	11	12	04	15
1	13	08	11	05	06	15	00	03	04	07	02	12	01	10	14	09
2	10	06	09	00	12	11	07	13	15	01	03	14	05	02	08	04
3	03	15	00	06	10	01	13	08	09	04	05	11	12	07	02	14

 S_5 -Box

S_5	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	02	12	04	01	07	10	11	06	08	05	03	15	13	00	14	09
1	14	11	02	12	04	07	13	01	05	00	15	10	03	09	08	06
2	04	02	01	11	10	13	07	08	15	09	12	05	06	03	00	14
3	11	08	12	07	01	14	02	13	06	15	00	09	10	04	05	03

 S_6 -Box

S_6	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	12	01	10	15	09	02	06	08	00	13	03	04	14	07	05	11
1	10	15	04	02	07	12	09	05	06	01	13	14	00	11	03	08
2	09	14	15	05	02	08	12	03	07	00	04	10	01	13	11	06
3	04	03	02	12	09	05	15	10	11	14	01	07	06	00	08	13

 S_7 -Box

S_7	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	04	11	02	14	15	00	08	13	03	12	09	07	05	10	06	01
1	13	00	11	07	04	09	01	10	14	03	05	12	02	15	08	06
2	01	04	11	13	12	03	07	14	10	15	06	08	00	05	09	02
3	06	11	13	08	01	04	10	07	09	05	00	15	14	02	03	12

S_8 -Box

S_8	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	13	02	08	04	06	15	11	01	10	09	03	14	05	00	12	07
1	01	15	13	08	10	03	07	04	12	05	06	11	00	14	09	02
2	07	11	04	01	09	12	14	02	00	06	10	13	15	03	05	08
3	02	01	14	07	04	10	08	13	15	12	09	00	03	05	06	11

Beispiel 3.3. Für das Element $x = (1, 1, 1, 0, 0, 0)$ gilt

$$a = (1, 0) = 2, \quad b = (1, 1, 0, 0) = 12$$

und damit

$$S_1(x) = S_1(2, 12) = 3 = (0, 0, 1, 1)$$

und

$$S_3(x) = S_3(2, 12) = 5 = (0, 1, 0, 1)$$

Die S -Boxen sind so konstruiert, dass sie gewisse Sicherheitsanforderungen erfüllen. Zu den wichtigsten Konstruktionsmerkmalen der S -Boxen gehören:

Bemerkung 3.2.

1. Jede S -Box hat sechs Eingangs- und vier Ausgangsbits.
2. Die S -Boxen sind so konstruiert, dass sie nicht linear und auch nicht affin linear sind, dh. es gibt keine 4×6 -Matrix A über \mathbb{F}_2 und keinen Vektor $b \in \mathbb{F}_2^4$ so dass

$$S(x) = A \cdot x + b$$

Das gilt auch nicht näherungsweise, dh. eine solche Beziehung gilt nicht einmal für eine Mehrheit der Elemente von \mathbb{F}_2^6 .

3. Unterscheiden sich $x, x' \in \mathbb{F}_2^6$ an genau einer Stelle, so unterscheiden sich $S(x)$ und $S(x')$ an mindestens zwei Stellen. Damit tragen die S -Boxen entscheidend zur Diffusion bei.
4. Unterscheiden sich $x, x' \in \mathbb{F}_2^6$ an den beiden mittleren Stellen, so unterscheiden sich $S(x)$ und $S(x')$ an mindestens zwei Stellen.
5. Für einen festen Randteil $a \in \mathbb{F}_2^2$ werden alle Elemente von \mathbb{F}_2^4 genau einmal angenommen, dh. in jeder Zeile der S -Boxtabellen steht jedes Element von \mathbb{F}_2^4 genau einmal.

3.3. Der Schlüsselfahrplan und Vorbereitung der Runden

1. Vorbereitung der Schlüsselerzeugung

Gegeben ist ein 64-Bit-Schlüssel

$$k = (k_1, k_2, \dots, k_{64}) \in \mathbb{F}_2^{64}$$

wobei allerdings jedes achte Bit ein Paritätsprüfbits ist, d.h. für $l = 1, \dots, 8$ gilt

$$k_{8 \cdot l} = \sum_{i=1}^7 k_{8 \cdot l - i}$$

Beachten Sie dabei, dass die Summe in \mathbb{F}_2 gebildet wird, sodass dort gilt

$$\sum_{i=0}^7 k_{8 \cdot l - i} = 0$$

Dieser Schlüssel wird wie folgt für die Verarbeitung vorbereitet:

1. Wende die Abbildung

$$PC-1 : \mathbb{F}_2^{64} \longrightarrow \mathbb{F}_2^{28} \times \mathbb{F}_2^{28}$$

auf den Schlüssel k an, wobei $PC-1$ durch die folgende Tabelle gegeben ist:

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Diese Abbildung (*Permuted Choice-1*) lässt also jedes achte Bit (die Paritätsprüfbits) weg und vertauscht die verbleibenden Bits nach dem angegebenen Muster, d.h. wenn

$$k = (k_1, k_2, \dots, k_{63}, k_{64})$$

ist, so ist

$$k^{(0)} := PC-1(k) = (k_{57}, k_{49}, k_{41}, \dots, k_{20}, k_{12}, k_4)$$

2. Der permutierte und reduzierte Schlüssel $k^{(0)} = (k_1^{(0)}, k_2^{(0)}, \dots, k_{56}^{(0)})$ wird in der Mitte (also entlang der durchgezogenen Linie in der Tabelle von PC-1) in zwei Hälften

$$k^{(0)} = C_0 \| D_0$$

aufgeteilt, wobei $C_0 = (k_1^{(0)}, k_2^{(0)}, \dots, k_{28}^{(0)})$ und $D_0 = (k_{29}^{(0)}, k_{30}^{(0)}, \dots, k_{56}^{(0)})$. Ausgehend vom Originalschlüssel k bedeutet das

$$C_0 = (k_{57}, k_{49}, \dots, k_{44}, k_{36}), \quad D_0 = (k_{63}, k_{55}, \dots, k_{12}, k_4)$$

3. Definiere Zahlen (Verschiebungsindeks) v_i wie folgt:

$$v_i = \begin{cases} 1 & \text{falls } i = 1, 2, 9 \text{ oder } 16 \\ 2 & \text{sonst} \end{cases}$$

2. Erzeugung des i -ten Rundenschlüssels $k^{(i)}$

In der Vorbereitung wurde $k^{(0)} = C_0 \| D_0$ bereitgestellt.

Für $i = 1, \dots, 16$ erzeuge zunächst C_i und D_i aus C_{i-1} und D_{i-1} und dann $k^{(i)}$ aus $C_i \| D_i$. Gehe dabei vor wie folgt:

1. Empfange C_{i-1} und D_{i-1} aus der Vorrunde (bzw. der Vorbereitung).
2. Verschiebe C_{i-1} bzw. D_{i-1} jeweils um v_i Positionen nach links, um C_i bzw. D_i zu erhalten, dh. wenn

$$C_{i-1} = (c_1^{(i-1)}, c_2^{(i-1)}, c_3^{(i-1)}, \dots, c_{28}^{(i-1)})$$

und wenn $i = 1, 2, 9$ oder 16 , so ist

$$C_i = (c_2^{(i-1)}, c_3^{(i-1)}, \dots, c_{27}^{(i-1)}, c_{28}^{(i-1)}, c_1^{(i-1)})$$

und für alle anderen i ist

$$C_i = (c_3^{(i-1)}, c_4^{(i-1)}, \dots, c_{28}^{(i-1)}, c_1^{(i-1)}, c_2^{(i-1)})$$

Entsprechend für D_i .

3. Setze

$$k^{(i)} = PC\text{-}2(C_i \| D_i)$$

wobei $PC\text{-}2$ aus $C_i \| D_i$ genau 48 Bits in der folgenden Reihenfolge auswählt:

14	17	11	24	1	5
2	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	52	50	36	29	32

dh. ist $C_i \| D_i = (a_1, a_2, \dots, a_{56})$, so ist

$$k^{(i)} = (a_{14}, a_{17}, a_{11}, \dots, a_{36}, a_{29}, a_{32})$$

Dieser Schlüssel $k^{(i)}$ wird in Runde i verwendet.

Bemerkung 3.3. Die Verschiebungen sind so strukturiert, dass

$$C_{16} \| D_{16} = C_0 \| D_0$$

Bemerkung 3.4. Das Verfahren stellt sicher, dass jedes relevante Bit des Schlüssel in genau 14 Rundenschlüsseln vorkommt.

Beispiel 3.4. Wir betrachten den Schlüssel $k = 0x\ 78aab48dff59c6d3$ (in Hexadezimal-darstellung), binär also

$$k = 011110001010101010110100100011011111111010110011100011011010011$$

Wenden wir $PC\text{-}1$ darauf an, so erhalten wir

$$k^{(0)} = 1101111011110001000101111011101001001011100001110110101$$

dh.

$$C_0 = 1101111011110001000101111011, \quad D_0 = 1101001001011100001110110101$$

In der ersten Runde weden C_0 und D_0 jeweils um eine Position nach links verschoben, also

$$C_1 = 1011110111100010001011110111, \quad D_1 = 1010010010111000011101101011$$

Wenden wir nun auf

$$C_1 \| D_1 = 10111101111000100010111101111010010010111000011101101011$$

die Auswahlfunktion $PC\text{-}2$ an, so erhalten wir

$$k^{(1)} = 0011111111110111001000101111011001001111101010$$

bzw. in Hexadezimaldarstellung

$$k^{(1)} = 0x 3ffdc8bd93ea$$

Setzen wir das jetzt für 16 Runden um, so erhalten wir die folgenden Rundenschlüssel (alle in Hexadezimaldarstellung):

$$\begin{array}{ll} k^{(1)} = 0x 3ffdc8bd93ea & k^{(2)} = 0x 8739dd4f672c \\ k^{(3)} = 0x 1f6ef57859cc & k^{(4)} = 0x df7da8c0d0bf \\ k^{(5)} = 0x daa7edc73ea9 & k^{(6)} = 0x d9de0fba1b79 \\ k^{(7)} = 0x 61bbe13db36 & k^{(8)} = 0x b0fce7552db0 \\ k^{(9)} = 0x dc9e5694f2d5 & k^{(10)} = 0x 66fa7ef386e1 \\ k^{(11)} = 0x aef5669aaaf0b & k^{(12)} = 0x ea4f7b3e7714 \\ k^{(13)} = 0x edf3397961e2 & k^{(14)} = 0x 879fdbbe4e80b \\ k^{(15)} = 0x 7f7ad3e6365e & k^{(16)} = 0x 3d5b3dcb9513 \end{array}$$

Beispiel 3.5. Um das Prinzip jetzt nochmal zu verdeutlichen, wollen wir ein DES–ähnliches Feistel–Netzwerk mit vier Runden und Schlüsseln und Nachrichten der Länge 16 Bit aufbauen und untersuchen. Wir gehen dabei davon aus, dass der Schlüssel k keine redundanten Stellen und keine Paritätsprüfbits enthält, sodass also alle 16 Zeichen relevant sind.

Daher entfällt auch die Notwendigkeit einer Abbildung $PC\text{-}1$, und wir können direkt mit $k^{(0)} = k$ beginnen und schreiben

$$k^{(0)} = C_0 \| D_0$$

mit $C_0 = (k_1, \dots, k_8)$ und $D_0 = (k_9, \dots, k_{16})$.

Für die Runden 1 bis 4 erzeugen wir Rundenschlüssel $k^{(i)}$ wie folgt:

1. Aus Runde $i - 1$ erhalten wir $C_{i-1} \| D_{i-1}$.
2. Erzeuge C_i aus C_{i-1} und D_i aus D_{i-1} durch eine Verschiebung um 2 Positionen nach links.

3. Wende die Abbildung PC auf $C_i \| D_i$ an und setze

$$k^{(i)} = PC(C_i \| D_i)$$

wobei PC gegeben ist durch

$$PC = (14 \quad 8 \quad 1 \quad 9 \quad 12 \quad 3 \quad 2 \quad 16 \quad 5 \quad 7 \quad 15 \quad 10)$$

Beachten Sie, dass auch bei diesem Verfahren sichergestellt ist, dass

$$C_4 = C_0, \quad D_4 = D_0$$

Beispiel 3.6. In der Situation von Beispiel 3.5 betrachten wir den Schlüssel

$$k = 0x abcd$$

in der Hexadezimaldarstellung, dh. in der binären Schreibweise

$$k = 1010101111001101$$

sodass also $C_0 = 10101011$ und $D_0 = 11001101$.

Der erste Rundenschlüssel entsteht dann wie folgt:

Zunächst verschieben wir C_0 und D_0 um zwei Positionen nach links und erhalten

$$C_1 = 10101110, \quad D_1 = 00110111$$

und somit ist

$$w := C_1 \| D_1 = 1010111000110111$$

Damit erhalten wir

$$\begin{aligned} k^{(1)} &= PC(C_1 \| D_1) \\ &= w_{14}w_8w_1w_9w_{12}w_3w_2w_{16}w_5w_7w_{15}w_{10} \\ &= 101011011110 \end{aligned}$$

bzw. hexadezimal geschrieben

$$k^{(1)} = ade$$

Der zweite Rundenschlüssel entsteht, indem wir zunächst C_1 und D_1 um zwei Positionen nach links verschieben und

$$C_2 = 10111010, \quad D_2 = 11011100$$

erhalten. Somit ist

$$w := C_2 \| D_2 = 1011101011011100$$

Damit erhalten wir

$$\begin{aligned} k^{(2)} &= PC(C_2 \| D_2) \\ &= w_{14}w_8w_1w_9w_{12}w_3w_2w_{16}w_5w_7w_{15}w_{10} \\ &= 101111001101 \end{aligned}$$

bzw. hexadezimal geschrieben

$$k^{(2)} = \text{bcd}$$

Der dritte Rundenschlüssel entsteht, indem wir zunächst C_2 und D_2 um zwei Positionen nach links verschieben und

$$C_3 = 11101010, \quad D_3 = 01110011$$

erhalten. Somit ist

$$w := C_3 \| D_3 = 1110101001110011$$

Damit erhalten wir

$$\begin{aligned} k^{(3)} &= PC(C_3 \| D_3) \\ &= w_{14}w_8w_1w_9w_{12}w_3w_2w_{16}w_5w_7w_{15}w_{10} \\ &= 001011111111 \end{aligned}$$

bzw. hexadezimal geschrieben

$$k^{(3)} = \text{2ff}$$

Der vierte Rundenschlüssel entsteht, indem wir zunächst C_3 und D_3 um zwei Positionen nach links verschieben und

$$C_4 = 10101011, \quad D_4 = 11001101$$

erhalten. Somit ist

$$w := C_4 \| D_4 = 1010101111001101$$

Beachten Sie dabei, dass in der Tat $C_4 \| D_4 = C_0 \| D_0 = k^{(0)}$. Damit erhalten wir

$$\begin{aligned} k^{(4)} &= PC(C_4 \| D_4) \\ &= w_{14}w_8w_1w_9w_{12}w_3w_2w_{16}w_5w_7w_{15}w_{10} \\ &= 111101011101 \end{aligned}$$

bzw. hexadezimal geschrieben

$$k^{(4)} = \text{f5d}$$

Beispiel 3.7. Für den Schlüssel

$$k = \text{b3f9}$$

erhalten wir mit der Methode aus Beispiel 3.5 die Rundenschlüssel

$$\begin{aligned} k^{(1)} &= \text{b3f} \\ k^{(2)} &= \text{dde} \\ k^{(3)} &= \text{aeb} \\ k^{(4)} &= \text{7d5} \end{aligned}$$

3. Vorbereitung der Runden

Nachdem die Schlüssel erzeugt sind, wird ein Nachrichtenblock

$$m = (m_1, m_2, \dots, m_{63}, m_{64}) \in \mathbb{F}_2^{64}$$

wie folgt für die Verarbeitung vorbereitet:

- Wende die Eingangspermutation IP auf m an, wobei IP gegeben ist durch die folgende Tabelle

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

sodass also

$$\tilde{m} := IP(m) = (m_{58}, m_{50}, m_{42}, \dots, m_{16}, m_8, m_{57}, m_{49}, \dots, m_{15}, m_7)$$

- Teile $\tilde{m} = IP(m)$ auf in einen linken Block L_0 und einen rechten Block R_0 ,

$$\tilde{m} = P(m) = L_0 \| R_0$$

wobei

$$L_0 = (\tilde{m}_1, \dots, \tilde{m}_{32}) = (m_{58}, m_{50}, m_{42}, \dots, m_{16}, m_8)$$

und

$$R_0 = (\tilde{m}_{33}, \dots, \tilde{m}_{64}) = (m_{57}, m_{49}, m_{41}, \dots, m_{15}, m_7)$$

- Übergebe $L_0 \| R_0$ an Runde 1 der Rundenverarbeitung

3.4. Rundenverarbeitung und Abschluss

Der wesentliche Teil der Verschlüsselung, und der Teil, auf dem die Sicherheit des DES-Verfahrens beruht, erfolgt in 16 Runden, die wie folgt aufgebaut sind:

In Runde i ($i = 1, \dots, 16$) gehe vor wie folgt:

1. Empfange $L_{i-1} \| R_{i-1}$ aus Runde $i - 1$ (bzw. aus der Vorbereitung, wenn $i = 1$).
2. Expandiere R_{i-1} mit der Expansionsfunktion

$$E : \mathbb{F}_2^{32} \longrightarrow \mathbb{F}_2^{48}$$

die wie folgt gegeben ist

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

dh. falls

$$R_{i-1} = (r_1, r_2, \dots, r_{31}, r_{32})$$

so ist

$$E(R_{i-1}) = (r_{32}, r_1, r_2, \dots, r_8, r_9, r_8, r_9, r_{10}, \dots, r_{31}, r_{32}, r_1)$$

3. Berechne

$$T = E(R_{i-1}) + k^{(i)}$$

mit dem i -ten Rundenschlüssel $k^{(i)}$ (und $+$ in \mathbb{F}_2^{48}).

4. Teile $T = (t_1, \dots, t_{48})$ auf in 8 Blöcke B_1, \dots, B_8 zu je 6 Bits, sodass also

$$B_l = (t_{(l-1)\cdot 6+1}, t_{(l-1)\cdot 6+2}, \dots, t_{l\cdot 6})$$

und wende auf jeden Block B_l die S-Box S_l (aus Abschnitt 3.2) an. Erhalte dadurch

$$N = (S_1(B_1), S_2(B_2), \dots, S_8(B_8)) \in \mathbb{F}_2^{32}$$

5. Auf N wende die Permutation π an, die gegeben ist durch die Tabelle

10	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

d.h. ist $N = (n_1, n_2, \dots, n_{32})$, so ist

$$\pi(N) = (n_{10}, n_7, n_{20}, n_{21}, \dots, n_{22}, n_{11}, n_4, n_{25})$$

6. Setze $f_{k^{(i)}}(R_{i-1}) = \pi(N)$.

7. Setze

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} + f_{k^{(i)}}(R_{i-1}) \end{aligned}$$

Nach der sechzehnten Runde erfolgt noch eine Abschlussbearbeitung der Daten wie folgt:

1. Vertausche in $L_{16} \| R_{16}$ die linke und die rechte Hälfte und setze

$$\tilde{y} = R_{16} \| L_{16}$$

2. Wende auf \tilde{y} die Abschlusspermutation $FP = IP^{-1}$ an, die gegeben ist durch

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

sodass also

$$FP(\tilde{y}) = (\tilde{y}_{40}, \tilde{y}_8, \tilde{y}_{48}, \dots, \tilde{y}_{64}, \tilde{y}_{32}, \tilde{y}_{39}, \tilde{y}_7, \dots, \tilde{y}_{57}, \tilde{y}_{25})$$

(Überzeugen Sie sich, dass FP tatsächlich die Umkehrpermutation zu IP ist.)

3. Das Chiffrat ist $y = FP(\tilde{y})$.

Beispiel 3.8. Wir betrachten DES mit dem Schlüssel $k = 0x\ 78aab48dff59c6d3$ in Hexadezimaldarstellung (aus Beispiel 3.4) und den Klartext $m = 0x\ 27f180da6309b5c9$, dh.

$$m = 001001111110001100000001101101001100011000010011011010111001001$$

Zunächst wenden wir IP an und bekommen

$$\tilde{m} = 00000111110010010111000001111111110011010000111000110000100111$$

und damit

$$\begin{aligned} L_0 &= 10011010010010100100000111110011, \\ R_0 &= 11001110010100111010100000011001 \end{aligned}$$

In der ersten Runde wenden wir auf R_0 die Expansion an und erhalten

$$E = 11100101110000101010011110101010000000011110011$$

Den ersten Rundenschlüssel haben wir schon in Beispiel 3.4 berechnet:

$$k^{(1)} = 001111011111101110010001011101100100111101010$$

Damit erhalten wir

$$T = E + k^{(1)} = 1101100000111110110111011010001001001100011001$$

Anwendung der S -Boxen liefert jetzt

$$N = 01111000001010000000011110000000$$

Nun wird N noch mit π permutiert und wir bekommen

$$f_{(k^{(1)})}(R_0) = \pi(N) = 0000000001010001010001001001001111$$

und damit

$$\begin{aligned} L_1 &= R_0 &= 11001110010100111010100000011001 \\ R_1 &= L_0 + f_{(k^{(1)})}(R_0) &= 10011010011000101110001110111100 \end{aligned}$$

Es ergibt sich (in Hexadezimaldarstellung)

$$L_1 \| R_1 = 0x\ ce53a8199a62e3bc$$

Führen wir das nach diesem Muster 16 Runden durch, so erhalten wir

$$\begin{aligned} L_{16} &= 00000100110011101101001011011011 \\ R_{16} &= 0011011010010111111010011001011 \end{aligned}$$

und damit (hexadezimal)

$$L_{16} \| R_{16} = 0x\ 04ced2db3697f4cb$$

Nach Vertauschen und Anwenden von FP erhalten wir das Chiffraut

$$y = 0001001101111011111101000010001101011110010001000010111100111111$$

bzw. (hexadezimal geschrieben)

$$y = 0x\ 137bf4235e442f3f$$

Beispiel 3.9. Die prinzipiellen Techniken von DES sollen wieder in einer vereinfachten Situation erläutert werden. Dazu greifen wir wieder das Beispiel 3.5 auf.

Bei der Verschlüsselung verzichten wir dabei auf die Eingangs- und Ausgangspermutation IP und FP , da die nichts zur Sicherheit betragen. Die Vorbereitung der Rundenverarbeitung besteht also lediglich darin, den Klartext $m \in \mathbb{F}_2^{16}$ in Empfang zu nehmen und in zwei Teile aufzuspalten

$$m = L_0 \| R_0$$

mit $L_0 = (m_1, \dots, m_8)$ und $R_0 = (m_9, \dots, m_{16})$.

In Runde i ($i = 1, \dots, 4$) gehen wir vor wie folgt:

1. Empfange $L_{i-1} \| R_{i-1}$ aus Runde $i - 1$ (bzw. der Vorbereitung, falls $i = 1$).
2. Expandiere R_{i-1} mit der Expansionsfunktion

$$E : \mathbb{F}_2^8 \longrightarrow \mathbb{F}_2^{12}$$

die gegeben ist durch

$$E = (6 \ 3 \ 4 \ 7 \ 6 \ 2 \ 8 \ 3 \ 7 \ 1 \ 5 \ 2)$$

3. Berechne

$$T = E(R_{i-1}) + k^{(i)}$$

mit dem Rundenschlüssel $k^{(i)}$.

4. Teile T in zwei Blöcke auf, $T = B_1 \| B_2$ und wende die S -Boxen S_1 und S_2 aus der DES-Verschlüsselung an,

$$N = S_1(B_1) \| S_2(B_2)$$

5. Wende die Permutation π gegeben durch

$$\pi = (3 \ 7 \ 6 \ 2 \ 8 \ 5 \ 1 \ 4)$$

und erhalten

$$f_{k^{(i)}}(R_{i-1}) = \pi(N)$$

6. Setze

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} + f_{k^{(i)}}(R_{i-1}) \end{aligned}$$

Die Abschlussbearbeitung besteht darin, die Blöcke L_4 und R_4 zu vertauschen,

$$y = R_4 \| L_4$$

Beispiel 3.10. Wir greifen nochmals das Beispiel 3.6 mit dem Schlüssel $k = 0x abcd$ auf. In Beispiel 3.6 haben wir schon die vier Rundenschlüssel berechnet,

$$k^{(1)} = 0x ade, \quad k^{(2)} = 0x bcd, \quad k^{(3)} = 0x 2ff, \quad k^{(4)} = 0x f5d$$

Der Klartext, den wir betrachten, ist $m = 0x 4321$, hexadezimal geschrieben, also binär

$$m = 0100001100100001$$

Damit erhalten wir

$$L_0 = 01000011, \quad R_0 = 00100001$$

Für Runde 1 berechnen wir

$$E(R_0) = 010000110000$$

Folglich ist

$$T = E(R_0) + k^{(1)} = 111011101110$$

mit Blöcken $B_1 = 111011$ und $B_2 = 101110$. Anwendung der S -Boxen liefert

$$N = 00000001$$

und die Permutation π führt zu

$$f_{k^{(1)}}(R_0) = \pi(N) = 00001000$$

Also erhalten wir

$$\begin{aligned} L_1 &= R_0 &= 00100001 \\ R_1 &= L_0 + f_{k^{(1)}}(R_0) &= 01001011 \end{aligned}$$

Das wird nun so auch für die Runden zwei bis vier durchgeführt.

Für Runde 2 ergibt das

$$\begin{aligned}
 E(R_1) &= 000101101011 \\
 T &= E(R_1) + k^{(2)} = 101010100110 \\
 N &= 00011011 \\
 f_{k^{(2)}}(R_1) &= \pi(N) = 11011100
 \end{aligned}$$

und damit

$$\begin{aligned}
 L_2 &= R_1 = 01001011 \\
 R_2 &= L_1 + f_{k^{(2)}}(R_1) = 11111101
 \end{aligned}$$

Für Runde 3 erhalten wir

$$\begin{aligned}
 E(R_2) &= 111011110111 \\
 T &= E(R_2) + k^{(3)} = 110000001000 \\
 N &= 11110110 \\
 f_{k^{(3)}}(R_2) &= \pi(N) = 11110011
 \end{aligned}$$

und damit

$$\begin{aligned}
 L_3 &= R_2 = 11111101 \\
 R_3 &= L_2 + f_{k^{(3)}}(R_1) = 10111000
 \end{aligned}$$

Runde 4 schließlich führt zu

$$\begin{aligned}
 E(R_3) &= 011000010110 \\
 T &= E(R_3) + k^{(4)} = 100101001011 \\
 N &= 10000010 \\
 f_{k^{(4)}}(R_3) &= \pi(N) = 01000010
 \end{aligned}$$

und damit

$$\begin{aligned}
 L_4 &= R_3 = 10111000 \\
 R_4 &= L_3 + f_{k^{(4)}}(R_1) = 10111111
 \end{aligned}$$

Zum Abschluss werden noch L_4 und R_4 vertauscht und wir erhalten das Chiffraut

$$y = 101111110111000$$

bzw. $y = 0x\ bfb8$ in der Hexadezimaldarstellung.

Beispiel 3.11. Für den Schlüssel $k = 0x\ b3f9$ aus Beispiel 3.7 und den Klartext

$$m = 0x\ 8675$$

erhalten wir

$$\begin{aligned} L_1 \| R_1 &= 01110101 \| 00100101 \\ L_2 \| R_2 &= 00100101 \| 11101100 \\ L_3 \| R_3 &= 11101100 \| 01110100 \\ L_4 \| R_4 &= 01110100 \| 01111010 \end{aligned}$$

und damit das Chiffrat

$$y = 0111101001110100$$

bzw. $y = 0x\ 7a74$.

3.5. Dechiffrierung von DES

Die Dechiffrierung von DES ist prinzipiell sehr ähnlich zur Chiffrierung, da es sich bei DES um ein Feistelnetzwerk handelt. Durch die spezielle Struktur des Netzwerkes bei DES kann das Verfahren sogar noch weiter vereinfacht werden.

Dazu erzeugen wir zunächst Rundenschlüssel für die Entschlüsselung wie folgt:

Für den gegebenen Schlüssel k setze

$$\tilde{k}^{(0)} = PC-1(k)$$

(mit $PC-1$ aus dem Verschlüsselungsalgorithmus) und teile \tilde{k}_0 auf in eine linke und eine rechte Hälfte

$$\tilde{k}^{(0)} = \tilde{C}_0 \| \tilde{D}_0$$

Beachte dabei, dass gilt

$$\tilde{C}_0 \| \tilde{D}_0 = C_0 \| D_0 = C_{16} \| D_{16}$$

Ferner setze

$$w_i = \begin{cases} 0 & \text{für } i = 1 \\ 1 & \text{für } i = 2, 9, 16 \\ 2 & \text{sonst} \end{cases}$$

Damit erzeugen wir für $i = 1, \dots, 16$ Rundenschlüssel \tilde{k}_i sukzessive wie folgt:

1. Erzeuge \tilde{C}_i aus \tilde{C}_{i-1} durch eine zyklische Verschiebung um w_i Positionen nach rechts (also in der anderen Richtung als bei der Schlüsselerzeugung für die Chiffrierung), und analog erzeuge \tilde{D}_i aus \tilde{D}_{i-1} durch eine zyklische Verschiebung um w_i Positionen nach rechts.
2. Setze

$$\tilde{k}^{(i)} = PC-2(\tilde{C}_i \| \tilde{D}_i)$$

mit $PC-2$ aus dem Verschlüsselungsverfahren.

Hilfssatz 3.1. Für alle $i = 1, \dots, 16$ gilt

$$\tilde{k}^{(i)} = k^{(17-i)}$$

(mit den Rundenschlüsseln $k^{(17-i)}$ aus der Verschlüsselung).

Beweis: Da $w_1 = 0$ gilt $\tilde{C}_1 = \tilde{C}_0 = C_{16}$ und analog $\tilde{D}_1 = \tilde{D}_0 = D_{16}$. Damit gilt

$$\tilde{k}^{(1)} = PC\text{-}2(C_{16} \| D_{16}) = k^{(16)}$$

und die Aussage stimmt für $i = 1$. Hieraus leiten wir nun induktiv ab, dass auch für alle $i > 1$ gilt

$$\tilde{C}_i = C_{17-i}, \quad \tilde{D}_i = D_{17-i}, \quad \tilde{k}^{(i)} = k^{(17-i)}$$

Ist nämlich schon gezeigt, dass $\tilde{C}_{i-1} = C_{17-(i-1)}$ und $\tilde{D}_{i-1} = D_{17-(i-1)}$, so folgt aus der Definition von w_i und der Tatsache, dass wir hier nach rechts verschieben, sofort, dass auch $\tilde{C}_i = C_{17-i}$ und $\tilde{D}_i = D_{17-i}$. Damit ist dann aber

$$\tilde{k}^{(i)} = PC\text{-}2(\tilde{C}_i \| \tilde{D}_i) = PC\text{-}2(C_{17-i} \| D_{17-i}) = k^{(17-i)}$$

Zur Vorbereitung der Entschlüsselungsrunden gehen wir vor wie folgt:

1. Auf das empfangene Chiffrat $y = (y_1, \dots, y_{64})$ wende die Eingangspermutation IP an und erhalte

$$\tilde{y} = IP(y) = (\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_{64})$$

Beachten Sie dabei, dass \tilde{y} mit dem Tupel \tilde{y} aus dem vorletzten Schritt der Chiffrierung übereinstimmt, denn IP ist die zu FP inverse Permutation.

2. Teile \tilde{y} auf in einen linken und eine rechten Block,

$$\tilde{y} = \tilde{L}_0 \| \tilde{R}_0$$

wobei $\tilde{L}_0 = (\tilde{y}_1, \dots, \tilde{y}_{32})$ und $\tilde{R}_0 = (\tilde{y}_{33}, \dots, \tilde{y}_{64})$.

Hilfssatz 3.2. Es ist $\tilde{L}_0 = R_{16}$ und $\tilde{R}_0 = L_{16}$.

Beweis: Das ist klar, denn

$$\tilde{L}_0 \| \tilde{R}_0 = IP(y) = IP(FP(R_{16} \| L_{16})) = R_{16} \| L_{16}$$

(beachten Sie dabei, dass am Ende der Verschlüsselung noch einmal vertauscht wurde.)

Die Entschlüsselungsrunden $i = 1, \dots, 16$ laufen nun ab wie folgt:

1. In Runde i empfange $\tilde{L}_{i-1} \parallel \tilde{R}_{i-1}$ aus der Vorrunde (bzw. aus der Vorbereitung, wenn $i = 1$).
2. Berechne $f_{\tilde{k}(i)}(\tilde{R}_{i-1})$.
3. Setze

$$\tilde{L}_i = \tilde{R}_{i-1}, \quad \tilde{R}_i = \tilde{L}_{i-1} + f_{\tilde{k}(i)}(\tilde{R}_{i-1})$$

und übergebe $\tilde{L}_i \parallel \tilde{R}_i$ an die nächste Runde (bzw. an die Abschlussverarbeitung).

Hilfssatz 3.3. Für alle $i = 0, \dots, 16$ gilt

$$\tilde{L}_i = R_{16-i}, \quad \tilde{R}_i = L_{16-i}$$

Beweis: Durch Induktion nach i :

Für $i = 0$ ist das gerade der vorangegangene Hilfssatz 3.2.

Ist nun schon gezeigt, dass

$$\tilde{L}_i = R_{16-i}, \quad \tilde{R}_i = L_{16-i}$$

so gilt

$$\tilde{L}_{i+1} = \tilde{R}_i \stackrel{IV}{=} L_{16-i} = R_{16-i-1} = R_{16-(i+1)}$$

und

$$\begin{aligned} \tilde{R}_{i+1} &= \tilde{L}_i + f_{\tilde{k}(i+1)}(\tilde{R}_i) \\ &\stackrel{IV}{=} R_{16-i} + f_{k^{(17-(i+1))}}(L_{16-i}) \\ &= R_{16-i} + f_{k^{(16-i)}}(L_{16-i}) \\ &= L_{16-i-1} + f_{k^{(16-i)}}(R_{16-i-1}) + f_{k^{(16-i)}}(R_{16-i-1}) \\ &= L_{16-(i+1)} \end{aligned}$$

und damit ist die Behauptung allgemein gezeigt.

Speziell gilt nach diesem Lemma

$$\tilde{L}_{16} = R_0, \quad \tilde{R}_{16} = L_0$$

Ein Abschlusschritt beendet nun die Dechiffrierung:

1. Vertausche $\tilde{L}_{16} \parallel \tilde{R}_{16}$ zu

$$\tilde{m} = \tilde{R}_{16} \parallel \tilde{L}_{16}$$

2. Wende die Abschlusspermutation $FP = IP^{-1}$ an und setze

$$m' = FP(\tilde{m})$$

Hilfssatz 3.4. Die Nachricht m' ist der Klartext m zu y .

Beweis: Es ist

$$m' = FP(\tilde{m}) = FP(\tilde{R}_{16} \parallel \tilde{L}_{16}) = FP(L_0 \parallel R_0) = FP(IP(m)) = m$$

Damit entschlüsselt dieses Verfahren tatsächlich das Chiffrat.

Beispiel 3.12. Wir betrachten DES mit dem Schlüssel $k = 0x\text{78aab48dff59c6d3}$ in Hexadezimaldarstellung (aus Beispiel 3.4 und das Chiffrat $y = 0x\text{137bf4235e442f3f}$, das wir in Beispiel 3.8 erzeugt haben).

Wir erhalten dann als Rundenschlüssel für die Dechiffrierung genau die Rundenschlüssel der Chiffrierung in umgekehrter Reihenfolge, also

$$\begin{aligned}\tilde{k}^{(1)} &= k^{(16)} = 0x\text{3d5b3dcb9513} \\ \tilde{k}^{(2)} &= k^{(15)} = 0x\text{7f7ad3e6365e} \\ &\vdots \\ \tilde{k}^{(16)} &= k^{(1)} = 0x\text{3ffdc8bd93ea}\end{aligned}$$

Führen wir damit die Dechiffrierung gemäß obigem Schema durch, so erhalten wir tatsächlich wieder

$$m = 0x\text{27f180da6309b5c9}$$

Beispiel 3.13. Das vereinfachte DES–Verfahren, das wir in Beispiel 3.5 aufgebaut haben, hat die Eigenschaft, dass $C_4 \parallel D_4 = C_0 \parallel D_0$. Damit überträgt sich das DES–Dechiffrierungsverfahren auf diesen Kontext. Wir definieren hierfür

$$w_i = \begin{cases} 0 & \text{für } i = 1 \\ 2 & \text{sonst} \end{cases}$$

und gehen bei der Rundenschlüsselerzeugung für die Dechiffrierung so vor, wie bei der Rundenschlüsselerzeugung für die Chiffrierung, wobei wir lediglich dort, wo wir bei der Chiffrierung in Runde i um zwei Positionen nach links verschoben haben, jetzt um w_i Positionen nach rechts verschieben.

Damit ist dann das Verfahren zur Dechiffrierung eines Chiffrats y komplett identisch mit dem in Beispiel 3.9 beschriebenen Verfahren zur Chiffrierung (lediglich mit $\tilde{k}^{(i)}$ ($= k^{(5-i)}$) anstelle von $k^{(i)}$).

Beispiel 3.14. Für den Schlüssel $k = 0x\text{abcd}$ aus Beispiel 3.6 ergibt der Schlüsselfahrplan für die Entschlüsselung tatsächlich die Rundenschlüssel

$$\tilde{k}^{(1)} = k^{(4)} = 0x\text{f5d}, \tilde{k}^{(2)} = k^{(3)} = 0x\text{2ff}, \tilde{k}^{(3)} = k^{(2)} = 0x\text{bcd}, \tilde{k}^{(4)} = k^{(1)} = 0x\text{ade}$$

In Beispiel 3.10 hatten wir das Wort $m = 4321$ verschlüsselt und das Chiffrat $y = 0x\ bfb8$ erhalten. Wenn wir nun die Entschlüsselung auf y an, so erhalten wir

$$\begin{aligned}\widetilde{L}_1 \parallel \widetilde{R}_1 &= 10111000 \parallel 11111101 \\ \widetilde{L}_2 \parallel \widetilde{R}_2 &= 11111101 \parallel 01001011 \\ \widetilde{L}_3 \parallel \widetilde{R}_3 &= 01001011 \parallel 00100001 \\ \widetilde{L}_4 \parallel \widetilde{R}_4 &= 00100001 \parallel 01000011\end{aligned}$$

und damit

$$m = 0100001100100001$$

bzw. $m = 0x\ 4321$, sodass wir also tatsächlich m zurückgewonnen haben.

Dechiffrieren wir mit dieser Methode $y = 0x\ a2f5$, so erhalten wir

$$\begin{aligned}\widetilde{L}_1 \parallel \widetilde{R}_1 &= 11110101 \parallel 11100111 \\ \widetilde{L}_2 \parallel \widetilde{R}_2 &= 11100111 \parallel 11101100 \\ \widetilde{L}_3 \parallel \widetilde{R}_3 &= 11101100 \parallel 11100111 \\ \widetilde{L}_4 \parallel \widetilde{R}_4 &= 11100111 \parallel 11101000\end{aligned}$$

also

$$m = 1110100011100111$$

bzw $m = 0x\ e8e7$.

Beispiel 3.15. Für den Schlüssel $k = 0x\ b3f9$ aus Beispiel 3.7 und das Chiffrat $y = 0x\ 9d17$ erhalten wir

$$\begin{aligned}\widetilde{L}_1 \parallel \widetilde{R}_1 &= 00010111 \parallel 11100011 \\ \widetilde{L}_2 \parallel \widetilde{R}_2 &= 11100011 \parallel 00111011 \\ \widetilde{L}_3 \parallel \widetilde{R}_3 &= 00111011 \parallel 10001000 \\ \widetilde{L}_4 \parallel \widetilde{R}_4 &= 10001000 \parallel 10111011\end{aligned}$$

also

$$m = 1011101110001000$$

bzw $m = 0x\ bb88$.

3.6. Die Sicherheit von DES

Aufgrund der Paritätsprüfbits haben die Schlüssel k von DES eine effektive Länge von 56 Bits und der Schlüsselraum eine Größe von 2^{56} . Das wurde zu Zeiten der Konzeption von DES noch als ausreichend betrachtet (obwohl es auch damals schon Stimmen gab, die einen längeren Schlüssel forderten), um eine vollständige Schlüsselsuche effektiv

auszuschließen. Mit zunehmender Rechenleistung wurde das aber immer mehr zu einem Kritikpunkt an DES.

Das Prinzip der vollständigen Schlüsselsuche für ein Blockverschlüsselungsverfahren mit (endlichem) Schlüsselraum K und Verschlüsselung-/Entschlüsselungsalgorithmus $e = e_k$, $d = d_k$ kann wie folgt beschrieben werden:

1. Gegeben ist eine Anordnung des Schlüsselraums $K = \{k_1, k_2, \dots, k_n\}$.
2. Gegeben ist ein Klartext / Geheimtextpaar $(m, y) \in \mathbb{F}_2^{64} \times \mathbb{F}_2^{64}$.
3. Für $i = 1, \dots, n$ überprüfe, ob

$$d_{k_i}(y) = m$$

Wenn ja, STOPP.

Zu beachten ist dabei, dass es nicht ausgeschlossen ist, dass es mehrere Schlüssel $k \in K$ gibt, für die $d_k(y) = m$ gilt (*false positives*). Beim DES–Verfahren liegt die Wahrscheinlichkeit dafür bei $\frac{1}{2^8}$. Um sich also davon zu überzeugen, dass der gefundene Schlüssel korrekt ist, sollte er noch an weiteren Klartext / Geheimtextpaaren überprüft werden. Im Extremfall sind also 2^{56} Schlüssel zu testen, im Mittel sind es 2^{55} . Übliche Rechner eignen sich für solche Probleme nicht sonderlich gut, allerdings wurden schon früh spezielle Schlüsselsuchmaschinen entwickelt, die für diese Aufgabe optimiert sind. Im Jahr 1998 gelang es dann der Maschine *deep crack* einen brute–force–Angriff auf DES in 56 Stunden erfolgreich durchzuführen. Seither haben sich die Methoden und Techniken der Hardware weiter verbessert (und vor allem verbilligt), sodass Maschinen für weniger als 10 000 \$ gebaut werden können, die einen erfolgreichen Angriff auf DES in wenigen Tagen durchführen (*deep crack* kostete etwa 250 000 \$).

Seit Einführung von DES hält sich auch das Gerücht, dass die NSA dafür gesorgt habe, dass DES ein Hintertürchen enthalte, welches es der NSA ermögliche, alle Nachrichten mitzulesen. Deshalb hat man auch früh begonnen, nach analytischen und mathematischen Wegen zu Suchen, um DES zu brechen. In der Tat geht ein Großteil der Entwicklung der Kryptoanalyse (als wissenschaftliche Disziplin) auf die Beschäftigung mit DES zurück.

Ein erster Versuch war die von Biham und Shamir 1990 vorgeschlagene **differentielle Kryptoanalyse**. Hierbei handelt es sich um einen chosen–plaintext–Angriff. Dabei wird untersucht, wie sich Änderungen an Klartexten auf das Chifferrat auswirken, speziell geringfügige gleichartige Änderungen an vielen Klartexten. So werden etwa von vielen Klartexten immer die gleichen Stellen bitweise geändert und dann untersucht, wie sich

dabei Chiffrat ändert. Gelingt es, in den Änderungen des Chiffrats Strukturen und Muster zu erkennen, etwa unterschiedliche Häufigkeitsmuster bei den Ausgangsdifferenzen, so erlauben diese eventuell Rückschlüsse auf die Struktur des Schlüssels. Beim DES–Verfahren wird dabei speziell untersucht, ob man die Eingänge der S –Boxen dadurch bestimmen kann, da die Struktur der S –Boxen bekannt sind und man daher untersuchen kann, welche S –Box–Eingänge bestimmte Differenzen der Ausgänge bewirken können. Das DES–Verfahren erweist sich allerdings als ziemlich resistent gegen diese Art von Angriffen. Wie inzwischen bekannt ist, war es eines der Design–Kriterien für das DES–Verfahren und speziell für den Aufbau der S –Boxen, dass gleiche Änderungen an unterschiedlichen Klartexten möglichst zu unterschiedlichen Änderungen an den Chiffrauen führen sollen. Für eine erfolgreiche differentielle Kryptoanalyse des DES–Verfahrens müsste für mindestens 2^{48} selbstgewählte Klartexte (von 2^{64} möglichen Klartexten) das Chiffrat bekannt sein, und das kann als unrealistisch betrachtet werden.

Von Matsui wurde 1993 ein weiterer analytischer Angriff, die **lineare Kryptoanalyse** vorgestellt. Die Annahme hier ist, dass DES, obwohl es aufgrund der S –Boxen nicht linear ist, trotzdem gewisse lineare Bestandteile enthält und daher teilweise linear angenähert werden kann. Deshalb wird bei diesem Angriff versucht, Linearkombinationen von Schlüsselbits linear durch Klartext- und Ciphertextbits darzustellen und hierdurch die Anzahl der zu untersuchenden Schlüssel zu reduzieren. Für eine erfolgreiche lineare Kryptoanalyse des DES–Verfahrens müsste aber immer noch für mindestens 2^{43} selbstgewählte Klartexte (von 2^{64} möglichen Klartexten) das Chiffrat bekannt sein, und daher ist auch dieser Angriff keine realistische Gefahr für das DES–Prinzip.

Da DES inzwischen einem brute force–Angriff nicht mehr gewachsen ist, wird es in der Praxis in sicherheitsrelevanten Bereichen nicht mehr eingesetzt, ist aber immer noch Vorlage für benutzte Verfahren. Eine Variante, die sich noch im Einsatz befindet, ist 3DES, das üblicherweise in der Form

$$y = e_{k_3} (d_{k_2} (e_{k_1} (m)))$$

mit drei Schlüsseln $k_1, k_2, k_3 \in \mathbb{F}_2^{64}$ verwendet wird (*encryption–decryption–encryption–mode*). Dieses Verfahren gilt als sicher (wenn man gewisse Schlüsselkombinationen vermeidet) und wird auch praktisch verwendet (etwa im Finanzwesen). Als neuer Standard hat sich jedoch das im nächsten Abschnitt vorgestellte AES–Verfahren durchgesetzt.

4. Advanced Encryption Standard AES

Nachdem DES ab ca. 1995 nicht mehr als sicher galt, wurde es durch AES abgelöst. Im Gegensatz zu DES war AES keine Auftragsentwicklung. Im Jahr 1997 schrieb NIST (*National Institute of Standards and Technology*) einen Wettbewerb zur Entwicklung von AES aus, dessen Gewinner der (belgische) Algorithmus Rijndael war. AES steht in drei Varianten zur Verfügung:

Schlüssellänge	Rundenanzahl
128	10
192	12
256	14

Üblich (z.B. bei TLS, dem Internetstandard IPsec oder dem WLAN–Vreschlüsselungsstandard IEEE 802.11i) ist AES128, auf das wir uns hier beschränken.

Die Anzahl der Verschlüsselungsrunden bei AES ist kleiner als bei DES. Trotzdem beeinträchtigt das die Sicherheit des Verfahrens nicht, da AES nicht nach dem Prinzip eines Feistel–Netzwerks aufgebaut ist und in jeder Runde mit dem kompletten Wort arbeitet. Ein weiterer entscheidender Unterschied zu DES ist die Tatsache, dass AES mit dem endlichen Körper $F_{2^8} = \mathbb{F}_{256}$ arbeitet vergleiche dazu auch Anhang C).

4.1. Der Körper mit 256 Elementen

Für den Körper mit $256 = 2^8$ Elementen gibt es verschiedene Beschreibungen, die in der Praxis verwendet werden. In der Codierungstheorie benutzt man etwa die Beschreibung durch die Relation

$$\alpha^8 = \alpha^4 + \alpha^3 + \alpha^2 + 1 \quad (4.1)$$

Für AES dagegen betrachten wir diesen Körper mit der **Rijndael–Relation**

$$\alpha^8 = \alpha^4 + \alpha^3 + \alpha + 1 \quad (4.2)$$

Es handelt sich dabei beide Male um den gleichen Körper, es wird lediglich eine andere Sicht darauf gegeben. Wir arbeiten in diesem Abschnitt immer mit der Relation (4.2).

Die Elemente x des Körpers $k = \mathbb{F}_{256}$ schreiben sich alle in der Form

$$x = b_7 \cdot \alpha^7 + b_6 \cdot \alpha^6 + b_5 \cdot \alpha^5 + b_4 \cdot \alpha^4 + b_3 \cdot \alpha^3 + b_2 \cdot \alpha^2 + b_1 \cdot \alpha + b_0$$

mit eindeutig bestimmten Zahlen $b_i \in \mathbb{F}_2 = \{0, 1\}$, und dieses Element wird mit dem binären 8–Tupel

$$x = (b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0) \in \mathbb{F}_2^8$$

identifiziert. Die Addition ist dann in offensichtlicher Weise gegeben: Für

$$x = (b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0), \quad y = (c_7, c_6, c_5, c_4, c_3, c_2, c_1, c_0)$$

ist

$$x + y = (b_7 + c_7, b_6 + c_6, b_5 + c_5, b_4 + c_4, b_3 + c_3, b_2 + c_2, b_1 + c_1, b_0 + c_0)$$

Die Multiplikation nutzt (wie in Anhang C ausführlich beschrieben) die Relation aus. Dazu werden zunächst α -Potenzen in der üblichen Weise multipliziert und dann werden alle α^n mit $n \geq 8$ mithilfe der Relation auf α -Potenzen vom Grad höchstens 7 reduziert, also etwa

$$\alpha^4 \cdot \alpha^5 = \alpha^9 = \alpha \cdot \alpha^8 \stackrel{(*)}{=} \alpha \cdot (\alpha^4 + \alpha^3 + \alpha + 1) = \alpha^5 + \alpha^4 + \alpha^2 + \alpha$$

wobei wir für die Gleichheit (*) die Relation (4.2) eingesetzt haben.

Gegebenenfalls muss diese Relation öfter als einmal angewendet werden, etwa bei

$$\begin{aligned} \alpha^7 \cdot \alpha^6 &= \alpha^{13} = \alpha^5 \cdot \alpha^8 \\ &= \alpha^5 \cdot (\alpha^4 + \alpha^3 + \alpha + 1) \\ &= \alpha^9 + \alpha^8 + \alpha^6 + \alpha^5 \\ &= \alpha \cdot (\alpha^4 + \alpha^3 + \alpha + 1) + (\alpha^4 + \alpha^3 + \alpha + 1) + \alpha^6 + \alpha^5 \\ &= \alpha^5 + \alpha^4 + \alpha^2 + \alpha + \alpha^4 + \alpha^3 + \alpha + 1 + \alpha^6 + \alpha^5 \\ &= \alpha^6 + \alpha^3 + \alpha^2 + 1 \end{aligned}$$

Für allgemeine Elemente wird erst nach dem Distributivgesetz ausmultipliziert und dann obige Regel angewandt, also etwa

$$\begin{aligned} (\alpha^5 + \alpha) \cdot (\alpha^4 + \alpha^2) &= \alpha^9 + \alpha^7 + \alpha^5 + \alpha^3 = \alpha^5 + \alpha^4 + \alpha^2 + \alpha + \alpha^7 + \alpha^5 + \alpha^3 \\ &= \alpha^7 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha \end{aligned}$$

Dadurch wird eine Multiplikation auf der Menge \mathbb{F}_2^8 der Bytes erklärt, und es ist etwa

$$\begin{aligned} (0, 0, 1, 0, 0, 0, 1, 0) \cdot (0, 0, 0, 1, 0, 1, 0, 0) &= (\alpha^5 + \alpha) \cdot (\alpha^4 + \alpha^2) \\ &= \alpha^7 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha \\ &= (1, 0, 0, 1, 1, 1, 1, 0) \end{aligned}$$

Beispiel 4.1. Wir betrachten die beiden Elemente $x, y \in \mathbb{F}_{256}$ mit

$$x = (1, 0, 0, 1, 0, 0, 0, 1), \quad y = (0, 0, 1, 1, 0, 0, 1, 1)$$

Hierfür gilt

$$\begin{aligned} x + y &= (1 + 0, 0 + 0, 0 + 1, 1 + 1, 0 + 0, 0 + 0, 0 + 1, 1 + 1) \\ &= (1, 0, 1, 0, 0, 0, 1, 0) \end{aligned}$$

und

$$\begin{aligned}
x \cdot y &= ((1, 0, 0, 1, 0, 0, 0, 1) \cdot (0, 0, 1, 1, 0, 0, 1, 1)) \\
&= (\alpha^7 + \alpha^4 + 1) \cdot (\alpha^5 + \alpha^4 + \alpha + 1) \\
&= \alpha^{12} + \alpha^{11} + \alpha^8 + \alpha^7 + \alpha^9 + \alpha^8 + \alpha^5 + \alpha^4 + \alpha^5 + \alpha^4 + \alpha + 1 \\
&= \alpha^{12} + \alpha^{11} + \alpha^9 + \alpha^7 + \alpha + 1 \\
&= \alpha^4 \cdot (\alpha^4 + \alpha^3 + \alpha + 1) + \alpha^3 \cdot (\alpha^4 + \alpha^3 + \alpha + 1) \\
&\quad + \alpha \cdot (\alpha^4 + \alpha^3 + \alpha + 1) + \alpha^7 + \alpha + 1 \\
&= \alpha^8 + \alpha^7 + \alpha^5 + \alpha^4 + \alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 \\
&\quad + \alpha^5 + \alpha^4 + \alpha^2 + \alpha + \alpha^7 + \alpha + 1 \\
&= \alpha^8 + \alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha^2 + 1 \\
&= \alpha^4 + \alpha^3 + \alpha + 1 + \alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha^2 + 1 \\
&= \alpha^7 + \alpha^6 + \alpha^2 + \alpha + \\
&= (1, 1, 0, 0, 0, 1, 1, 0)
\end{aligned}$$

Die Multiplikation in \mathbb{F}_{256} lässt sich technisch recht einfach realisieren. Dazu betrachten wir den „Überlauf“

$$u = (0, 0, 0, 1, 1, 0, 1, 1)$$

(der für $\alpha^4 + \alpha^3 + \alpha + 1$ steht). Dann gilt

$$\begin{aligned}
&\alpha \cdot (b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0) \\
&= (0, 0, 0, 0, 0, 0, 1, 0) \cdot (b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0) \\
&= (b_6, b_5, b_4, b_3, b_2, b_1, b_0, 0) + b_7 \cdot (0, 0, 0, 1, 1, 0, 1, 1) \\
&= (b_6, b_5, b_4, b_3, b_2, b_1, b_0, 0) + b_7 \cdot u
\end{aligned}$$

Da

$$(0, 0, 0, 0, 0, 1, 0, 0) = \alpha^2 = \alpha \cdot \alpha = (0, 0, 0, 0, 0, 1, 0) \cdot (0, 0, 0, 0, 0, 1, 0)$$

und entsprechend für

$$\begin{aligned}
(0, 0, 0, 0, 1, 0, 0, 0) &= \alpha^3 = \alpha \cdot \alpha^2 = (0, 0, 0, 0, 0, 1, 0) \cdot (0, 0, 0, 0, 0, 1, 0, 0) \\
&\vdots \\
(1, 0, 0, 0, 0, 0, 0, 0) &= \alpha^7 = \alpha \cdot \alpha^6 = (0, 0, 0, 0, 0, 1, 0) \cdot (0, 1, 0, 0, 0, 0, 0, 0)
\end{aligned}$$

kann dadurch (und mit dem Distributivgesetz) die gesamte Multiplikation beschrieben werden.

Beispiel 4.2. Es ist

$$(1, 1, 0, 1, 0, 1, 1, 1) \cdot (0, 1, 1, 1, 0, 0, 1, 1, 0) = (0, 1, 0, 0, 1, 0, 0, 1)$$

Beispiel 4.3. Es ist

$$(1, 1, 0, 1, 0, 1, 1, 1)^2 = (0, 0, 1, 1, 1, 1, 1, 1)$$

und

$$(1, 0, 1, 0, 1, 0, 1, 0) \cdot (0, 1, 0, 1, 0, 1, 0, 1) = (0, 1, 0, 1, 1, 0, 0, 1)$$

Die Bestimmung der inversen Elemente ist komplexer und nicht mit einfachen algebraischen Mitteln möglich. Dazu ist es in diesem Fall erforderlich, die gesamte Multiplikationstabelle von \mathbb{F}_{2^8} (mit dieser Relation) aufzustellen und für ein gegebenes $x \in \mathbb{F}_{2^8}$ in dieser Tabelle nach dem y mit $x \cdot y = 1$ zu suchen. Wir stellen hier eine Tabelle mit den inversen Elementen zur Verfügung. Dazu betrachten wir für jedes Byte $x \in \mathbb{F}_2^8$ dessen Hexadezimaldarstellung, d.h. wir schreiben

$$x = L \| R$$

(mit $L, R \in \mathbb{F}_2^4$) und betrachten L und R als Hexadezimalzahlen. Dann erhalten wir die inversen Elemente aus der folgenden Tabelle:

$L \backslash R$	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	-	01	8d	f6	cb	52	7b	d1	e8	4f	29	c0	b0	e1	e5	c7
1	74	b4	aa	4b	99	2b	60	5f	58	3f	fd	cc	ff	40	ee	b2
2	3a	6e	5a	f1	55	4d	a8	c9	c1	0a	98	15	30	44	a2	c2
3	2c	45	92	6c	f3	39	66	42	f2	35	20	6f	77	bb	59	19
4	1d	fe	37	67	2d	31	f5	69	a7	64	ab	13	54	25	e9	09
5	ed	5c	05	ca	4c	24	87	bf	18	3e	22	f0	51	ec	61	17
6	16	5e	af	d3	49	a6	36	43	f4	47	91	df	33	93	21	3b
7	79	b7	97	85	10	b5	ba	3c	b6	70	d0	06	a1	fa	81	82
8	83	7e	7f	80	96	73	be	56	9b	9e	95	d9	f7	02	b9	a4
9	de	6a	32	6d	d8	8a	84	72	2a	14	9f	88	f9	dc	89	9a
a	fb	7c	2e	c3	8f	b8	65	48	26	c8	12	4a	ce	e7	d2	62
b	0c	e0	1f	ef	11	75	78	71	a5	8e	76	3d	bd	bc	86	57
c	0b	28	2f	a3	da	d4	e4	0f	a9	27	53	04	1b	fc	ac	e6
d	7a	07	ae	63	c5	db	e2	ea	94	8b	c4	d5	9d	f8	90	6b
e	b1	0d	d6	eb	c6	0e	cf	ad	08	4e	d7	e3	5d	50	1e	b3
f	5b	23	38	34	68	46	03	8c	dd	9c	7d	a0	cd	1a	41	1c

Diese Tabelle ist so zu lesen: Zu einem x mit Hexadezimaldarstellung $x = L \| R$ ist das Element aus der Zeile mit Index L und der Spalte mit Index R das inverse Element.

Beispiel 4.4. Für $x = (1, 1, 0, 1, 0, 1, 1, 0)$ ist $L = (1, 1, 0, 1) = d$ und $R = (0, 1, 1, 0) = 6$ und daher ist das Inverse in der Zeile mit Index d und der Spalte mit Index 6 zu finden, also

$$x^{-1} = 0x\text{e}2 = (1, 1, 1, 0, 0, 0, 1, 0)$$

Durch Multiplikation können wir tatsächlich verifizieren, dass

$$(1, 1, 0, 1, 0, 1, 1, 0) \cdot (1, 1, 1, 0, 0, 0, 1, 0) = (0, 0, 0, 0, 0, 0, 0, 1)$$

4.2. S–Boxen und Byte–Substitution in AES

Die S–Boxoperation in AES ist einheitlich durch eine Abbildung

$$\text{Sub} : \mathbb{F}_2^8 \longrightarrow \mathbb{F}_2^8$$

gegeben, die wie folgt definiert ist:

Setze

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

Zu einem

$$x = (b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0) \in F_2^8 = F_{256}$$

bestimme sein Inverses in F_{256} (gemäß obiger Tabelle), schreibe

$$x^{-1} = (u_7, u_6, u_5, u_4, u_3, u_2, u_1, u_0)$$

(falls $x = 0$, so nehmen wir hier statt x^{-1} auch $(0, 0, 0, 0, 0, 0, 0, 0)$) und setze

$$u = \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{pmatrix}$$

Berechne

$$v = \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{pmatrix} := A \cdot u + c$$

Dann ist

$$\text{Sub}(x) = (v_7, v_6, v_5, v_4, v_3, v_2, v_1, v_0)$$

Beispiel 4.5.

1. Für $x = 0$ gilt

$$v = A \cdot \vec{0} + c = c$$

und damit

$$\text{Sub}(x) = (0, 1, 1, 0, 0, 0, 1, 1)$$

2. Für $x = 1 = (0, 0, 0, 0, 0, 0, 0, 1)$ ist $x^{-1} = (0, 0, 0, 0, 0, 0, 0, 1)$ und damit

$$v = A \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + c = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

also

$$\text{Sub}(x) = (0, 1, 1, 1, 1, 1, 0, 0)$$

3. Für $x = (0, 1, 1, 1, 0, 1, 0, 1) = 0x75$ ist $x^{-1} = 0xb5 = (1, 0, 1, 1, 1, 0, 0, 1)$ und

damit

$$v = A \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} + c = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

also

$$\text{Sub}(x) = (0, 0, 1, 0, 0, 0, 0, 1)$$

Beispiel 4.6. Für $x = (1, 0, 0, 1, 1, 1, 1, 1) = 0x9f$ ist $x^{-1} = 0x9a = (1, 0, 0, 1, 1, 0, 1, 0)$ und damit

$$v = A \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} + c = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

also

$$\text{Sub}(x) = (1, 1, 0, 1, 1, 0, 1, 1)$$

4.3. Der AES–Schlüsselfahrplan

Das AES–128 Verfahren benötigt insgesamt 11 Schlüssel, einen Schlüssel k_0 für die Eingangsverarbeitung des Klartextes, und 10 Rundenschlüssel k_1, \dots, k_{10} für jede der 10 Verarbeitungsrunden. Dazu schreiben wir zunächst

$$k = K_0 \| K_1 \| \cdots \| K_{14} \| K_{15}$$

mit $K_i \in \mathbb{F}_2^8 = \mathbb{F}_{256}$ (also byteweise). Die 11 Rundenschlüssel k_0, k_1, \dots, k_{10} werden nun sukzessive über ein Schema der Form

$$\begin{array}{cccc} W_0 & W_1 & W_2 & W_3 \\ W_4 & W_5 & W_6 & W_7 \\ \vdots & \vdots & \vdots & \vdots \\ W_{40} & W_{41} & W_{42} & W_{43} \end{array}$$

mit $W_i \in \mathbb{F}_{256}^4$ aufgebaut, das rekursiv wie folgt gefüllt wird:

1. Die Elemente W_0, W_1, W_2 und W_3 der ersten Zeile werden mit den Elementen des Schlüssels k gefüllt,

$$\begin{aligned} W_0 &= K_0 \| K_1 \| K_2 \| K_3, & W_1 &= K_4 \| K_5 \| K_6 \| K_7, \\ W_2 &= K_8 \| K_9 \| K_{10} \| K_{11}, & W_3 &= K_{12} \| K_{13} \| K_{14} \| K_{15} \end{aligned}$$

2. Für $i \geq 1$ definiere rekursiv

- zunächst $W_{4 \cdot i}$ durch

$$W_{4 \cdot i} = W_{4 \cdot (i-1)} + g_i(W_{4 \cdot (i-1)})$$

wobei

$$g_i = \mathbb{F}_{256}^4 \longrightarrow \mathbb{F}_{256}^4$$

definiert ist als

$$g_i(b_0, b_1, b_2, b_3) = (\text{Sub}(b_1) + \alpha^{i-1}, \text{Sub}(b_2), \text{Sub}(b_3), \text{Sub}(b_0))$$

dh.

- a) Zunächst wird (b_0, b_1, b_2, b_3) byteweise um eine Position nach links verschoben um (b_1, b_2, b_3, b_0) zu erhalten.
- b) Nach dieser Verschiebung wird auf jedes einzelne Byte die S -Box Sub aus Abschnitt 4.2 angewendet, um $(\text{Sub}(b_1), \text{Sub}(b_2), \text{Sub}(b_3), \text{Sub}(b_4))$ zu erhalten.
- c) Im letzten Schritt wird zum linken Byte der sogenannte Rundenkoeffizient α^{i-1} (in \mathbb{F}_{256}) addiert.
- dann $W_{4 \cdot i+j}$ für $j = 1, 2, 3$ durch

$$W_{4 \cdot j+i} = W_{4 \cdot i+j-1} + W_{4 \cdot (i-1)+j}$$

Beispiel 4.7. Wir betrachten $u = (0x ab, 0x cd, 0x 32, 0x 45) \in \mathbb{F}_{256}^4$ (hexadezimal geschrieben) und wollen $g_1(u)$, $g_4(u)$ und $g_{10}(u)$ berechnen.

Linksverschiebung macht daraus

$$v = (0x cd, 0x 32, 0x 45, 0x ab)$$

Anwendung der S -Box auf jedes Element führt zu

$$w = (0x bd, 0x 23, 0x 6e, 0x 62)$$

Jetzt sind noch die jeweiligen Rundenkoeffizienten zu addieren.

Für $i = 1$ ist

$$\alpha^{i-1} = \alpha^0 = 1 = (0, 0, 0, 0, 0, 0, 0, 0, 1) = 0x\ 01$$

und das ergibt

$$(1, 0, 1, 1, 1, 1, 0, 1) + (0, 0, 0, 0, 0, 0, 0, 1) = (1, 0, 1, 1, 1, 1, 0, 0) = 0x\ ba$$

sodass also

$$g_1(u) = (0x\ ba, 0x\ 23, 0x\ 6e, 0x\ 62)$$

Für $i = 4$ ist das

$$\alpha^{i-1} = \alpha^3 = (0, 0, 0, 0, 1, 0, 0, 0) = 0x\ 08$$

und das ergibt

$$(1, 0, 1, 1, 1, 1, 0, 1) + (0, 0, 0, 0, 1, 0, 0, 0) = (1, 0, 1, 1, 0, 1, 0, 1) = 0x\ b5$$

sodass also

$$g_4(u) = (0x\ b5, 0x\ 23, 0x\ 6e, 0x\ 62)$$

Für $i = 10$ ist das

$$\alpha^{i-1} = \alpha^9 = (0, 0, 0, 1, 1, 0, 1, 0) = 0x\ 1a$$

und das ergibt

$$(1, 0, 1, 1, 1, 1, 0, 1) + (0, 0, 1, 1, 0, 1, 1, 0) = (1, 0, 0, 0, 1, 0, 1, 1) = 0x\ 8b$$

sodass also

$$g_{10}(u) = (0x\ 8b, 0x\ 23, 0x\ 6e, 0x\ 62)$$

Beispiel 4.8. Für $u = (0x\ ab, 0x\ cd, 0x\ 32, 0x\ 45) \in \mathbb{F}_{256}^4$ (hexadezimal geschrieben) gilt:

$$g_6(u) = (0x\ 3a, 0x\ 44, 0x\ 71, 0x\ 21)$$

und

$$g_9(u) = (0x\ 01, 0x\ 44, 0x\ 71, 0x\ 21)$$

Definition 4.1. Der i -te Rundenschlüssel für AES ist gegeben durch

$$k_i = W_{4 \cdot i} \| W_{4 \cdot i+1} \| W_{4 \cdot i+2} \| W_{4 \cdot i+3}$$

Bemerkung 4.1. Es ist $k_0 = k$ der ursprüngliche Schlüssel von AES.

Beispiel 4.9. Wir betrachten den Schlüssel,

$$k = 0x\ abcd324574a8b2d5f1f2f3f47abc6543$$

also

$$W_0 = 0x\ 7abc6543, \ W_1 = 0x\ 74a8b2d5, \ W_2 = 0x\ f1f2f3f4, \ W_3 = 0x\ abcd3245$$

Den Wert für $g_1(W_3)$ haben wir in Beispiel 4.7 schon berechnet,

$$g_1(W_3) = 0x\ ba236e62$$

Dann gilt

$$\begin{aligned} W_4 &= W_0 + g(W_3) &= 0x\ c09f0b21 \\ W_5 &= W_4 + W_1 &= 0x\ b437b9f4 \\ W_6 &= W_5 + W_2 &= 0x\ 45c54a00 \\ W_7 &= W_6 + W_3 &= 0x\ ee087845 \end{aligned}$$

Damit gilt

$$k_1 = 0x\ c09f0b21b437b9f445c54a00ee087845$$

4.4. Die AES–Diffusionsoperationen

Die Diffusionsoperationen haben den Zweck, den Einfluss des Klartextes auf das Chiffraut zu verschleiern und sollen dazu führen, dass kleine Änderungen am Klartext große und unkontrollierte Änderungen am Chiffraut bewirken.

4.4.1. ShiftRow–Operations

Bei diesen Operationen schreiben wir ein 128–Bit–Wort B in der Form

$$B = B_0 \| B_1 \| B_2 \| \dots \| B_{14} \| B_{15}$$

mit $B_i \in \mathbb{F}_{256}$ und ordnen diese Bytes an wie folgt

$$\begin{array}{cccc} B_0 & B_4 & B_8 & B_{12} \\ B_1 & B_5 & B_9 & B_{13} \\ B_2 & B_6 & B_{10} & B_{14} \\ B_3 & B_7 & B_{11} & B_{15} \end{array}$$

Dann ist

$$\text{ShiftRow}(B) = \begin{array}{cccc} B_0 & B_4 & B_8 & B_{12} \\ B_5 & B_9 & B_{13} & B_1 \\ B_{10} & B_{14} & B_2 & B_6 \\ B_{15} & B_3 & B_7 & B_{11} \end{array}$$

dh.

- In der ersten Zeile gibt es keine Verschiebung.
- In der zweiten Zeile kommt es zu einer Rechtsverschiebung um drei Bytes.
- In der dritten Zeile kommt es zu einer Rechtsverschiebung um zwei Bytes.
- In der vierten Zeile kommt es zu einer Rechtsverschiebung um ein Byte.

Das Ergebnis wird dann wieder in ein 128–Bit–Wort zurückgerechnet, dh.

$$\text{ShiftRow}(B) = B_0 \| B_5 \| B_{10} \| B_{15} \| B_4 \| B_9 \| \dots \| B_6 \| B_{11}$$

4.4.2. MixColumns–Operations

Hierfür schreiben wir ein 128–Bit–Wort C wieder in der Form

$$C = C_0 \| C_1 \| C_2 \| \dots \| C_{14} \| C_{15}$$

mit $C_i \in \mathbb{F}_{256}$ und ordnen diese Bytes erneut an wie folgt

$$\begin{matrix} C_0 & C_4 & C_8 & C_{12} \\ C_1 & C_5 & C_9 & C_{13} \\ C_2 & C_6 & C_{10} & C_{14} \\ C_3 & C_7 & C_{11} & C_{15} \end{matrix}$$

Dann ist

$$\text{MixCol}(C) = \begin{pmatrix} \alpha & \alpha + 1 & 1 & 1 \\ 1 & \alpha & \alpha + 1 & 1 \\ 1 & 1 & \alpha & \alpha + 1 \\ \alpha + 1 & 1 & 1 & \alpha \end{pmatrix} \cdot \begin{pmatrix} C_0 & C_4 & C_8 & C_{12} \\ C_1 & C_5 & C_9 & C_{13} \\ C_2 & C_6 & C_{10} & C_{14} \\ C_3 & C_7 & C_{11} & C_{15} \end{pmatrix}$$

dh. MixCol multipliziert die 4×4 –Matrix

$$A = \begin{pmatrix} \alpha & \alpha + 1 & 1 & 1 \\ 1 & \alpha & \alpha + 1 & 1 \\ 1 & 1 & \alpha & \alpha + 1 \\ \alpha + 1 & 1 & 1 & \alpha \end{pmatrix}$$

mit C (ebenfalls als 4×4 –Matrix mit Elementen aus \mathbb{F}_{256} aufgefasst), wobei die Multiplikationen in \mathbb{F}_{256} durchzuführen sind.

Beispiel 4.10. Wir betrachten

$$C = 0x\ ab12dc43ef5698010fe12dc34ba56987$$

angeordnet als

$$\begin{pmatrix} 0x\ ab & 0x\ ef & 0x\ 0f & 0x\ 4b \\ 0x\ 12 & 0x\ 56 & 0x\ e1 & 0x\ a5 \\ 0x\ dc & 0x\ 98 & 0x\ 2d & 0x\ 69 \\ 0x\ 43 & 0x\ 07 & 0x\ c3 & 0x\ 87 \end{pmatrix}$$

Dann ist

$$\text{MixCol}(C) = \begin{pmatrix} 0x\ e4 & 0x\ a0 & 0x\ c8 & 0x\ 8c \\ 0x\ b3 & 0x\ f7 & 0x\ 62 & 0x\ 26 \\ 0x\ df & 0x\ 9b & 0x\ ea & 0x\ ae \\ 0x\ ae & 0x\ ea & 0x\ 40 & 0x\ 04 \end{pmatrix}$$

also

$$\text{MixCol}(C) = 0x\ e4b3dfa0f79besc862ea408c26ae04$$

Beispiel 4.11. Für

$$C = 0x\ 0123456789abcdeffedcba9876543210$$

gilt

$$\text{MixCol}(C) = \begin{pmatrix} 0x\ 05 & 0x\ cd & 0x\ ba & 0x\ 32 \\ 0x\ 4d & 0x\ 67 & 0x\ 10 & 0x\ 98 \\ 0x\ b0 & 0x\ 89 & 0x\ fe & 0x\ 76 \\ 0x\ b8 & 0x\ 23 & 0x\ 54 & 0x\ dc \end{pmatrix}$$

also

$$\text{MixCol}(C) = 0x\ 054db0b8cd678923ba10fe54329876dc$$

4.5. Ablauf der AES–Verschlüsselung

Wie schon DES ist auch AES ein Blockverschlüsselungsverfahren (der Länge 128, 192 oder 256). Bei AES128 wird ein Klartext m der Länge 128 Bit mit einem Schlüssel k der Länge 128 Bit verarbeitet.

Vorbereitung:

Die Vorbereitung der Verschlüsselung umfasst die folgenden Schritte:

1. Erzeuge die Rundenschlüssel k_0, k_1, \dots, k_{10} .
2. Führe eine Schlüsseladdition durch (*key whitening*)

$$m^{(0)} = m + k_0$$

Rundenverarbeitung:

Jede Runde i ($i = 1, \dots, 10$) besteht aus drei Schichten. Dabei geht man vor wie folgt:

1. Empfange

$$m^{(i-1)} = (m_0^{(i-1)}, m_1^{(i-1)}, m_2^{(i-1)}, \dots, m_{14}^{(i-1)}, m_{15}^{(i-1)})$$

(mit $m_j^{(i-1)} \in \mathbb{F}_{256}$) aus Runde $i - 1$ (bzw. der Vorbereitung).

2. **Bytesubstitutionsschicht:** Führe eine Bytesubstitution durch

$$B = (B_0, B_1, \dots, B_{14}, B_{15}) = (\text{Sub}(m_0^{(i-1)}), \text{Sub}(m_1^{(i-1)}), \dots, \text{Sub}(m_{14}^{(i-1)}), \text{Sub}(m_{15}^{(i-1)}))$$

3. **Diffusionsschicht:**

- Führe eine ShiftRow–Operation durch:

$$C = (C_0, C_1, \dots, C_{14}, C_{15}) = \text{ShiftRow}(B_0, B_1, \dots, B_{14}, B_{15})$$

- Falls $i \leq 9$, so führe eine MixColumn–Operation durch:

$$D = (D_0, D_1, \dots, D_{14}, D_{15}) = \text{MixCol}(C_0, C_1, \dots, C_{14}, C_{15})$$

falls $i = 10$, so setze $D = C$.

4. **Schlüsseladditionsschicht:** Addiere den Rundenschlüssel:

$$m^{(i)} = D + k_i$$

Abschluss und Chiffrauszeugung:

Setze

$$y = AES(m) = m^{(10)}$$

Bemerkung 4.2. In Runde 10 entfällt die MixColumn–Operation.

4.6. Entschlüsselung von AES

Da AES nicht nach einem Feistelnetzwerk konstruiert ist, läuft die Entschlüsselung hier nicht komplett nach demselben Muster wie die Verschlüsselung (wie das etwa bei DES der Fall ist). Die Entschlüsselung arbeitet jedoch auch mit den Rundenschlüsseln k_0, k_1, \dots, k_{10} .

Der Fahrplan für die Entschlüsselung lässt sich etwa wie folgt beschreiben:

Vorbereitung:

Die Vorbereitung der Entschlüsselung umfasst die folgenden Schritte:

1. Empfange das Chiffra y .
2. Erzeuge die Rundenschlüssel k_0, k_1, \dots, k_{10} .
3. Setze $y^{(0)} = y$.

Rundenverarbeitung:

Jede Runde i ($i = 1, \dots, 10$) besteht aus drei Schichten. Dabei geht man vor wie folgt:

1. Empfange

$$y^{(i-1)} = (y_0^{(i-1)}, y_1^{(i-1)}, y_2^{(i-1)}, \dots, y_{14}^{(i-1)}, y_{15}^{(i-1)})$$

(mit $y_j^{(i-1)} \in \mathbb{F}_{256}$) aus Runde $i - 1$ (bzw. der Vorbereitung).

2. **inverse Schlüsseladditionsschicht:** Addiere den inversen Rundenschlüssel:

$$B = y^{(i-1)} + k_{11-i}$$

3. **inverse Diffusionsschicht:**

- Falls $i \geq 2$, so führe eine inverse MixColumn–Operation durch:

$$C = (C_0, C_1, \dots, C_{14}, c_{15}) = \text{MixCol}^{-1}(B_0, B_1, \dots, B_{14}, B_{15})$$

falls $i = 1$, so setze $C = B$.

- Führe eine inverse ShiftRow–Operation durch:

$$D = (D_0, D_1, \dots, D_{14}, D_{15}) = \text{ShiftRow}^{-1}(C_0, C_1, \dots, C_{14}, C_{15})$$

4. **inverse Bytesubstitutionsschicht:** Führe eine Byteresubstitution durch

$$y^{(i)} = (y_0^{(i)}, y_1^{(i)}, \dots, y_{14}^{(i)}, y_{15}^{(i)}) = (\text{Sub}^{-1}(D_0), \text{Sub}^{-1}(D_1), \dots, \text{Sub}^{-1}(D_{14}), \text{Sub}^{-1}(D_{15}))$$

Abschluss und Klartexterzeugung:

Führe eine Schlüsseladdition durch und setze

$$m = AES(y) = y^{(10)} + k_0$$

Bemerkung 4.3. Zu allen Operationen, die bei der Verschlüsselung benutzt werden, gibt es tatsächlich inverse Operationen. Das liegt daran, dass die Matrizen A , die in der Bytesubstitutionsschicht bzw. der MixColumn–Operation benutzt werden, invertierbar sind.

5. Grundlagen der asymmetrischen Verschlüsselung

Bei den bisher vorgestellten Verfahren zur Verschlüsselung handelt es sich ausschließlich um symmetrische Verfahren, dh.

1. derselbe geheime Schlüssel wird für die Ver- und die Entschlüsselung verwendet.
2. die Algorithmen zur Ver- und Entschlüsselung sind von der Struktur her sehr ähnlich (bei DES sogar identisch).

Moderne symmetrische Verfahren wie AES oder gewisse Weiterentwicklungen von DES gelten als sehr sicher und effizient, alle symmetrischen Verfahren besitzen aber einige grundsätzliche Nachteile, etwa:

- **Das Schüsselaustauschproblem:** Der symmetrische Schlüssel muss zwischen Alice und Bob sicher und auf einem geheimen Kanal ausgetauscht werden.
- **Das Schlüsselanzahlproblem:** In einem Netzwerk mit n Parteien, die alle paarweise miteinander sicher kommunizieren wollen, werden insgesamt $\binom{n}{2} = \frac{n \cdot (n+1)}{2}$ viele Schlüssel benötigt.
- **Das Betrugsproblem:** Da Alice und Bob beide die vollen Fähigkeiten zum Chiffrieren und Dechiffrieren besitzen, ist nicht nachweisbar, wer von beiden eine Nachricht erzeugt hat. Alice kann z.B. die Urheberschaft einer Nachricht leugnen und behaupten, Bob habe sie selbst erzeugt. Umgekehrt kann natürlich auch Bob eine Nachricht erzeugen und behaupten, sie käme von Alice.

Diese Probleme adressieren die asymmetrischen Verfahren. Die Grundidee dahinter, die auf Diffie, Hellman und Merkle zurückgeht ist die folgende:

Statt mit einem Schlüssel sowohl für die Chiffrierung als auch für die Dechiffrierung zu arbeiten, wird mit einem Schlüsselpaar gearbeitet, einem Schlüssel zum Chiffrieren, der nicht geheim sein muss, und einem Schlüssel zum Dechiffrieren, der nur dem Empfänger der Nachricht bekannt ist. Ein Schlüssel in einem asymmetrischen Verfahren besteht also aus einem Schlüsselpaar $(k_{\text{pub}}, k_{\text{pr}})$, bestehend aus einem öffentlichen Schlüssel (*public key*) k_{pub} und einem privaten Schlüssel (*private key*) k_{pr} .

Der öffentliche Schlüssel k_{pub} ist der Schlüssel, der für die Chiffrierung verwendet wird. Dieser muss nicht geheim gehalten werden und kann daher über unsichere Kanäle übertragen werden. Der private Schlüssel k_{pr} wird für die Entschlüsselung verwendet und ist geheim. Er wird nur vom Empfänger der Nachricht verwendet. Das bedeutet insbesondere, dass mit einem Schlüsselpaar $(k_{\text{pub}}, k_{\text{pr}})$ immer nur eine Kommunikation in **einer** Richtung möglich ist.

Ein Schlüssel zwischen Alice und Bob kann dann nach dem folgenden Grundmuster vereinbart werden:

1. Alice und Bob einigen sich auf ein symmetrisches Verschlüsselungsverfahren.
2. Bob schickt Alice seinen öffentlichen Schlüssel k_{pub} .
3. Alice wählt einen Schlüssel k für das symmetrische Verfahren aus und erzeugt mit dem öffentlichen Schlüssel von Bob ein Chiffraut von k ,

$$y = e_{k_{\text{pub}}}(k)$$

und schickt dieses an Bob.

4. Bob entschlüsselt das Chiffraut mit seinem privaten Schlüssel

$$k = d_{k_{\text{pr}}}(y)$$

5. Alice und Bob kommunizieren mit diesem Schlüssel k .

In dieser Variante ist das Verfahren allerdings noch nicht sicher und praxistauglich. Da der öffentliche Schlüssel von Bob allgemein bekannt ist, könnte ihn auch Catherine benutzen, um das Chiffraut eines Schlüssels an Bob zu schicken und sich dabei als Alice ausgeben. Damit könnte Sie mit Bob eine Kommunikation beginnen und Geheiminformationen empfangen, die Bob nur für Alice bestimmt hat. Daher ist zusätzlich noch ein Authentifizierungsschritt nötig, damit Bob sicher sein kann, dass der chiffrierte Schlüssel tatsächlich von Alice kommt. Wie wir noch sehen werden, können auch solche Authentifizierungsprozesse mit asymmetrischen Verfahren realisiert werden.

Grundvoraussetzung:

Damit ein asymmetrisches Verfahren funktionieren kann, müssen zwei Grundbedingungen erfüllt sein:

1. Der öffentliche Schlüssel k_{pub} und das verwendete Verfahren lassen keine Rückschlüsse auf k_{pr} zu.
2. Der öffentliche Schlüssel k_{pub} und das Chiffraut $y = E_{k_{\text{pub}}}(m)$ lassen keine Rückschlüsse auf die Nachricht m zu.

Zu diesem Zweck betrachten wir eine spezielle Klasse mathematischer Funktionen:
Eine **Einwegfunktion** ist eine Funktion $f : M \rightarrow N$ mit den folgenden Eigenschaften:

1. Für ein $m \in M$ ist $n = f(m)$ einfach zu berechnen, d.h. es gibt einen Algorithmus, der n aus m effizient ermittelt.
2. Für ein $n \in \text{Bild}(f)$ ist die Ermittlung eines $m \in M$ mit $n = f(m)$ schwer, dh. es gibt keinen effizienten (probabilistischen) Algorithmus, der zu einem gegebenen n mit nicht vernachlässigbarer Wahrscheinlichkeit ein m mit $f(m) = n$ berechnet.

Diese Begriffserklärung ist so noch etwas vage, reicht aber, um einen anschaulichen Zugang zum Begriff der Einwegfunktion zu bekommen. Für einen exakten Zugang betrachten wir

$$M = \mathbb{F}_2^* = \{(b_1, \dots, b_n) \mid n \in \mathbb{N}, b_i \in \mathbb{F}_2\}$$

Definition 5.1. Eine Einwegfunktion $f : \mathbb{F}_2^* \rightarrow N$ ist eine Funktion, für die gilt:

1. Es gibt einen Algorithmus, der für $m = (b_1, \dots, b_r)$ den Wert $f(m)$ in polynomialer Zeit in r berechnet.
2. Für jeden probabilistischen Algorithmus F gilt:

$$\forall c \in \mathbb{N} \exists r_0 \in N : \forall r \geq r_0 \quad p(\{m \in \mathbb{F}_2^r : f(F(f(m))) = f(m)\}) \leq \frac{1}{r^c}$$

Ein Algorithmus heißt dabei probabilistisch oder randomisiert, wenn er Zwischenergebnisse nach einem Zufallsprinzip auswählen oder berechnen kann. Man unterscheidet zwei Sorten von probabilistischen Algorithmen:

Las–Vegas–Algorithmen:

Diese Algorithmen führen immer zu einem korrekten Ergebnis, brechen allerdings unter Umständen ab. Ein Beispiel hierfür ist der Quicksort–Algorithmus mit Zufallsauswahl des Pivotelements.

Monte–Carlo–Algorithmen:

Diese Algorithmen liefern immer ein Ergebnis, dass allerdings mit einer gewissen (kontrollierten) Wahrscheinlichkeit falsch sein kann. Ein Beispiel hierfür ist der Miller–Rabin–Test auf die Primzahleigenschaft einer Zahl.

Es ist nicht bekannt, ob es tatsächlich Einwegfunktionen gibt. Falls das der Fall ist, so ist $P \neq NP$ und es gibt tatsächlich NP–schwere Probleme.

Bei asymmetrische Verfahren dienen Einwegfunktionen zur Verschlüsselung. Damit allerdings Bob wieder an die chiffrierte Nachricht kommt, ist es für ihn notwendig, das Umkehrproblem der Einwegfunktion zu lösen:

Eine Einwegfunktion f heißt **Einwegfunktion mit Falltür** (trapdoor–one–way–function), wenn es Zusatzinformationen Z zu der Einwegfunktion gibt, die es erlauben, einen deterministischen Algorithmus $F = F_Z$ zu finden, der in kontrollierter Zeit (dh. in polynomialer Zeit in r , falls $M = \mathbb{F}_2^*$ und $m \in \mathbb{F}_2^r$) zu einem gegebenen $n \in \text{Bild}(f)$ ein $m \in M$ findet mit $f(m) = n$.

Damit lassen sich sofort asymmetrische Verfahren konstruieren. Der öffentliche Schlüssel ist dann die Einwegfunktion, der private Schlüssel ist die Zusatzinformation (und der darauf basierende Algorithmus F_Z).

Mit den asymmetrischen Verfahren werden wichtige Probleme der Kryptographie gelöst:

1. Schlüsselaustausch:

Mit asymmetrischen Verfahren können Protokolle zum geheimen Schlüsselaustausch (über einen unsicheren Kanal) realisiert werden, z.B. DHKE (Diffie–Hellman–Schlüsselaustausch) oder das RSA–Schlüsseltransprotokoll.

2. Nachrichtenintegrität und Zurückweisbarkeit:

Mit asymmetrischen Verfahren können Protokolle für digitale Signaturen entwickelt werden, sodass der Empfänger sicher sein kann, dass die Nachricht während der Übertragung nicht manipuliert wurde und dass der Sender im Nachhinein nicht abstreiten kann, der Urheber der Nachricht zu sein.

3. Identifikation und Authentizierung:

Mit asymmetrischen Verfahren können challenge–and–response–Protokolle und Signaturverfahren entwickelt werden, die es erlauben Teilnehmer und Geräte eindeutig zu identifizieren.

4. Verschlüsselung:

Mit asymmetrischen Verfahren können schließlich auch Daten selbst verschlüsselt und sicher übertragen werden (wenn auch in der Regel unter sehr hohen Kosten).

Offen bleibt noch das Problem der Authentizität der öffentlichen Schlüssel, dh. die Frage ob der öffentliche Schlüssel, der zur Verfügung gestellt wird, tatsächlich von Bob stammt, oder ob es Catherine ist, die diesen Schlüssel unter falschem Namen und als Bobs öffentlichen Schlüssel verbreitet hat.

Dieses Problem wird in der Praxis über Zertifikate gelöst, die von einer neutralen Stelle (*trusted third party*) ausgestellt werden.

Formal lässt sich ein asymmetrisches Verschlüsselungsverfahren wie folgt definieren:

Definition 5.2. Ein **asymmetrisches** oder **public–key–Verschlüsselungsverfahren** ist ein Tupel $(\mathbf{K}, \mathbf{T}, \mathbf{C}, \mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ mit den folgenden Eigenschaften:

- Der **Schlüsselraum \mathbf{K}** ist eine Menge von Paaren (e, d) , den sogenannten Schlüsselpaaren. Dabei heißt die Komponente e der öffentliche und die Komponente d der private Schlüssel des Schlüsselpaares (e, d) .
- Der **Klartextraum \mathbf{T}** ist eine Menge, deren Elemente Klartexte heißen.
- Der **Chiffratraum \mathbf{C}** ist eine Menge, deren Elemente Chiffre heißen.
- Der **Schlüsselerzeugungsalgorithmus \mathbf{KGen}** ist ein probabilistischer Algorithmus, der aus der Eingabe eines zufälligen binären k –Tupels $(k \in \mathbb{N})$ z , ein Schlüsselpaar $(e, d) = \mathbf{KGen}(z)$.
- Der **Verschlüsselungsalgorithmus \mathbf{Enc}** ist ein probabilistischer Algorithmus, der aus einem Klartext $t \in T$ mithilfe des öffentlichen Schlüssels e eines Schlüsselpaares (e, d) (und einem zufälligen binären k –Tupel z) ein Chiffra $c \in C$ erzeugt, $c = \mathbf{Enc}_e(t)$.
- Der **Entschlüsselungsalgorithmus \mathbf{Dec}** ist ein deterministischer Algorithmus, der aus einem Chiffra $c \in C$ mithilfe des privaten Schlüssels d eines Schlüsselpaares (e, d) eine Klartext $t \in T$ erzeugt, $t = \mathbf{Dec}_d(c)$.

sodass die folgende Bedingung gilt:

Für jedes Schlüsselpaar (e, d) und jeden Klartext $t \in T$ gilt

$$\mathbf{Dec}_d(\mathbf{Enc}_e(t)) = t$$

dh. der Entschlüsselungsalgorithmus entschlüsselt korrekt (unabhängig von der beim Verschlüsseln verwendeten Zufallskomponente z).

6. Asymmetrische Verfahren basierend auf dem Faktorisierungsproblem

Der entscheidende Punkt bei der Konstruktion eines jeden asymmetrischen Verfahrens ist die Tatsache, dass aus dem öffentlichen Schlüssel e und dem Verschlüsselungsverfahren **Enc** nicht auf den privaten Schlüssel d geschlossen werden kann (das Entschlüsselungsverfahren **Dec** wird als im Prinzip bekannt angenommen). Das bedeutet, dass es sich bei der Verschlüsselungsfunktion

$$e = \mathbf{Enc}_e : T \longrightarrow C$$

um eine Einwegfunktion mit Falltür handeln muss, wobei es genau diese (dem Empfänger bekannte) Falltür ist, die die Ermittlung der privaten Komponente d des Schlüsselpaares (e, d) und damit die Entschlüsselung der Nachricht ermöglicht.

Eine dieser Falltüren ist die Primfaktorzerlegung großer Zahlen. Nach aktuellem Stand der Technik (und unter Vernachlässigung der Quantenalgorithmen) ist es Catherine nicht möglich, in überschaubarer Zeit die Primfaktorzerlegung einer großen Zahl n zu bestimmen (vergleiche dazu auch Anhang F). Bob dagegen hat diese Zahl n als Produkt

$$N = p \cdot q$$

von zwei (großen) Primzahlen ermittelt und hat daher die Falltür in die Zahl bereits mit eingebaut.

6.1. Das RSA–Verfahren

Das RSA–Verfahren, benannt nach seinen Erfindern Ron Rivest, Adi Shamir und Len Adleman nutzt genau das aus. Es ist das erste public-key–Verschlüsselungsverfahren, das (im Jahr 1976) veröffentlicht wurde und auch heute noch das wichtigste. In der Anwendung wird das RSA–Verfahren hauptsächlich eingesetzt

- zur Verschlüsselung kleinerer Datenmenge, insbesondere zum Schlüsseltransport.
- für digitale Signaturen.

Ver– und Entschlüsselung finden in dem Restklassenring $\mathbb{Z}/N\mathbb{Z}$ für ein geeignetes N statt. Der Klartextraum ist $T = \mathbb{F}_2^n$, wobei n so zu wählen ist, dass $2^n < N$. Ein Klartext t wird als binäre Darstellung einer ganzen Zahl $z < N$ aufgefasst, die dann mit ihrer Restklasse in $\mathbb{Z}/N\mathbb{Z}$ identifiziert wird.

Schlüsselerzeugung:

Bob erzeugt ein Schlüsselpaar $((e, N), d)$ wie folgt:

1. Bob wählt zwei große Primzahlen p und q mit $p \neq q$ und setzt $N = p \cdot q$.
2. Bob berechnet $\varphi(N) = (p - 1) \cdot (q - 1)$.
3. Bob wählt eine zu $\varphi(N)$ teilerfremde Zahl e .
4. Er bestimmt d mit $0 < d < N$ so, dass $d \cdot e = 1 \pmod{\varphi(N)}$.

Bezeichnung:

Die Zahl N heißt **RSA–Modul**.

Bemerkung 6.1. Die Bedingung $\text{ggT}(e, \varphi(N)) = 1$ stellt sicher, dass die Restklasse von e in $\mathbb{Z}/\varphi(N)\mathbb{Z}$ invertierbar ist, dass also tatsächlich ein d existiert mit $d \cdot e = 1 \pmod{\varphi(N)}$.

Mit diesen Daten ist (e, N) der öffentliche Schlüssel von Bob und d sein privater Schlüssel. Das erzeugte Schlüsselpaar ist damit $k = ((e, N), d)$.

Bemerkung 6.2. Für die Sicherheit des RSA–Verfahrens wird aktuell verlangt, dass es sich bei N (mindestens) um eine 1024–Bit–Zahl handelt (dass also N etwa 300 Stellen hat). Die beiden Primzahlen p und q sollten vergleichbar groß sein (also eine binäre Länge in der Nähe 512 haben), aber auch nicht zu nahe beieinander liegen. Ihre binäre Länge sollte sich mindestens um 1 aber höchstens um 30 unterscheiden. Dadurch wird eine effiziente Faktorisierung mit dem Fermat–Verfahren oder dem Quadratischen Sieb unterbunden und auch sichergestellt, dass keiner der Faktoren durch Probiedivision (oder eine andere Methode zum Abspalten kleinerer Primzahlen) ermittelt werden kann.

Beispiel 6.1. Bob wählt die beiden Primzahlen $p = 1171$ und $q = 983$ und berechnet

$$N = p \cdot q = 1171 \cdot 983 = 1151093.$$

$$\varphi(N) = (p - 1) \cdot (q - 1) = 1170 \cdot 982 = 1148940.$$

Für e wählt Bob die Zahl $e = 37$. Um zu überprüfen, dass e tatsächlich teilerfremd zu $\varphi(N)$ ist, und um die Zahl d zu berechnen, benutzt Bob den euklidischen Algorithmus. Dafür setzt er

$$r_0 = 1148940, \quad r_1 = 37, \quad i = 0$$

und führt sukzessive Division mit Rest durch:

$i = 0: 1\ 148\ 940 = 31\ 052 \cdot 37 + 16$. Wir setzen $r_2 = 16$.

$i = 1: 37 = 2 \cdot 16 + 5$. Wir setzen $r_3 = 5$.

$i = 2: 16 = 3 \cdot 5 + 1$. Wir setzen $r_4 = 1$.

$i = 3: 5 = 5 \cdot 1 + 0$. \longrightarrow **STOPP**.

Rückwärtsrechnen liefert

$$\begin{aligned} 1 &= 16 - 3 \cdot 5 &= 16 - 3 \cdot (37 - 2 \cdot 16) \\ &= 7 \cdot 16 - 3 \cdot 37 &= 7 \cdot (1\ 148\ 940 - 31\ 052 \cdot 37) - 3 \cdot 37 \\ &= 7 \cdot 1\ 148\ 940 + (-217\ 367) \cdot 31 \end{aligned}$$

Damit gilt modulo 1 148 940:

$$1 = -217\ 367 \cdot 37 = 931\ 573 \cdot 37$$

und daher ist die gesuchte Zahl

$$d = 931\ 573$$

Bob veröffentlicht seinen öffentlichen Schlüssel $k_{\text{pub}} = (37, 1\ 151\ 093)$. Den privaten Schlüssel $d = 931\ 573$ behält er für sich.

Die Wirkungsweise des RSA–Verfahrens beruht auf folgender Aussage, die wir schon in Anhang A hergeleitet haben, cf. Satz A.4.

Satz 6.1. Für jede ganze Zahl m mit $1 \leq m \leq N$ gilt

$$(m^e)^d = m \bmod N$$

Verschlüsselung:

Bob hat seine öffentlichen Schlüssel (e, N) veröffentlicht und Alice ist daher im Besitz dieser Daten. Um einen Klartext t (aufgefasst als eine Restklasse $t \in \mathbb{Z}/N\mathbb{Z}$, wie oben beschrieben) zu verschlüsseln, geht Alice nun vor wie folgt:

1. Alice benutzt den öffentlichen Schlüssel (e, N) und berechnet $c = t^e \bmod N$.
2. Alice schickt c über einen öffentlichen Kanal an Bob.

Beispiel 6.2. Alice will den Klartext $t = (1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1) \in \mathbb{F}_2^{18}$ geheim an Bob schicken. Sie wandelt die Binärdarstellung in eine ganze Zahl um und erhält

$$t = 172\ 275$$

Alice nutzt die Methode der iterierten Quadrate und berechnet

$$c = t^e = 172\ 275^{37} = 1\ 040\ 623 \bmod N$$

Alice schickt die Zahl $c = 1\ 040\ 623$ an Bob.

Entschlüsselung:

Bob nutzt seinen privaten Schlüssel d , um das von Alice empfangene Chiffrat c wie folgt zu entschlüsseln:

1. Bob benutzt seinen privaten Schlüssel d und berechnet $b = c^d \left(= (m^e)^d\right) \bmod N$.
2. Wegen $t = b \bmod N$ hat Bob die Nachricht entschlüsselt.

Beispiel 6.3. Bob hat von Alice das Chiffrat $c = 1\,040\,623$ empfangen. Er nutzt seinen privaten Schlüssel $d = 931\,573$ und ebenfalls die Methode der iterierten Quadrate und berechnet

$$b = c^d = 1\,040\,623^{931\,573} = 172\,275 \bmod N$$

Er wandelt b in eine Binärzahl um und erhält den Klartext

$$t = (1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1)$$

zurück.

Die Auswahl von e und d :

Trotz der Methode des iterierten Quadrieren ist die Potenzbildung teuer und rechenaufwendig. Das gilt speziell für den öffentlichen Exponenten e , da Alice oft Chipkarten für die Verschlüsselung benutzt (Kontokarten, Kreditkarten, Personalausweise) und die Prozessoren auf Chipkarten in der Regel nicht sonderlich mächtig sind. Deshalb ist es vernünftig, den Exponenten e möglichst klein zu wählen. Da e teilerfremd zu $\varphi(N) = (p - 1) \cdot (q - 1)$ sein muss, und $\varphi(N)$ immer gerade ist, scheidet $e = 2$ aus. Naheliegend wäre es nun, eine kleine Primzahl zu nehmen, etwa $e = 3$ oder $e = 7$, die teilerfremd zu $\varphi(N)$ ist (was in diesem Fall sehr schnell durch Probewidivision verifiziert werden kann). Das ist jedoch nicht unproblematisch, da es in bestimmten Situationen bei kleinen Exponenten e Angriffsmöglichkeiten gibt, die Rückschlüsse auf t oder gar die Bestimmung von t zulassen, ohne dass der Angreifer d kennt (**low-exponent-Angriff**). Eine häufige Wahl für e ist daher

$$e = 2^{16} + 1 = 65537$$

Das ist eine Primzahl und mit hoher Wahrscheinlichkeit teilerfremd zu $\varphi(N)$. Ferner ist e einerseits groß genug, um den low-exponent-Angriff abzuwehren und andererseits noch so klein und einfach strukturiert, um $t^e \bmod N$ effizient zu berechnen (mit 16 Quadrierungen und einer Multiplikation).

Da üblicherweise e als erstes bestimmt wird, ergibt sich d daraus zwangsläufig (über den euklidischen Algorithmus) und wir haben keinen Einfluss auf die Größenordnung von

d. Mit hoher Wahrscheinlichkeit ergibt sich daraus ein sehr hoher Wert für d . Es kann allerdings auch wünschenswert sein, für d einen kleinen Wert zu erhalten, etwa wenn die Entschlüsselung komplett auf einer Chipkarte stattfinden soll um das Risiko eines physischen Angriffs auf die Entschlüsselung zu minimieren. In dem Fall kann das RSA–Verfahren auch so aufgesetzt werden, dass zuerst ein d teilerfremd zu $\varphi(N)$ bestimmt wird und dann daraus das e mit $d \cdot e = 1 \pmod{N}$ berechnet wird. Das ist allerdings riskant, denn es gibt Angriffe, mit deren Hilfe das RSA–Verfahren gebrochen werden kann, wenn $d < N^{0.292}$.

Performanz:

Wählen wir, wie oben beschreiben $e = 2^{16} + 1$ (bzw. in dieser Größenordnung), so kann die Verschlüsselung effizient (mit insgesamt 17 Multiplikationen) durchgeführt werden. Da aber d in der Regel (bei den aktuellen Sicherheitsstandards) eine 1024–Bit Zahl ist, sind für die Entschlüsselung auf jeden Fall 1024 Quadrierungen erforderlich, im Mittel zusätzlich noch 512 Multiplikationen. Das ist speziell für Chipkarten mit in der Regel eher schwachen Prozessoren eine große Herausforderung. Eine deutliche Beschleunigung liefert hier der chinesische Restsatz CRT (vergleiche Anhang A, Satz A.5), der besagt, dass

$$\varepsilon : \mathbb{Z}/N\mathbb{Z} \longrightarrow \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$$

mit $\varepsilon(z \pmod{N}) = (z \pmod{p}, z \pmod{q})$ bijektiv ist. Anstelle c^d in $\mathbb{Z}/N\mathbb{Z}$ zu berechnen, reicht es, $(c \pmod{p})^d$ in $\mathbb{Z}/p\mathbb{Z} = \mathbb{F}_p$ und $(c \pmod{q})^d$ in $\mathbb{Z}/q\mathbb{Z} = \mathbb{F}_q$ zu berechnen. Setzen wir $c' = c \pmod{p}$ und schreiben wir

$$d = k \cdot (p - 1) + d' \quad \text{mit } 0 \leq d' \leq p - 2$$

so gilt in \mathbb{F}_p :

$$(c')^d = (c')^{k \cdot (p-1) + d'} = ((c')^k)^{p-1} \cdot (c')^{d'} = 1 \cdot (c')^{d'} = (c')^{d'}$$

(nach dem Satz von Fermat). Wir können für diese Berechnung also annehmen, dass $d' < p - 1$. Damit hat d' im allgemeinen nur noch halb so viele binäre Stellen wie d und daher sind zum d' –Potenzieren nur noch halb so viele Multiplikationen erforderlich wie zum d –Potenzieren. Entsprechend hat auch c' nur noch halb so viele binäre Stellen wie c . Da die Größenordnung der Operationen zum multiplizieren zweier k –Bit–Zahlen bei k^2 liegt, bedeutet das, dass für die Berechnung von $(c')^{d'}$ in \mathbb{F}_p nur ca. ein Achtel der Rechenoperationen notwendig ist, die für die Berechnung von c^d in $\mathbb{Z}/N\mathbb{Z}$ notwendig sind. Da die Potenzierung auch noch modulo q durchgeführt werden muss, reduziert der chinesische Restsatz daher insgesamt den Rechenaufwand auf etwa eine Viertel.

Beispiel 6.4. Wir greifen noch einmal das Beispiel 6.1 mit

$$p = 1171, \ q = 983, \ N = 1151093, \ e = 37, \ d = 931573$$

sowie das von Alice erzeugte Chiffrat $c = 1040623$ auf. Sowohl c als auch d haben hier 20 binäre Stellen, $d = 0x11100011011011110101$, bei der Urform der Entschlüsselung müssen also 20 Quadrate von binär 20-stelligen Zahlen berechnet werden, außerdem müssen hier 13 Multiplikationen durchgeführt werden. Es gilt

$$c_1 = c \bmod p = 775, \quad d_1 = d \bmod p - 1 = 253$$

und

$$c_2 = c \bmod q = 609, \quad d_2 = d \bmod q - 1 = 637$$

und diese Zahlen sind alle binär 8 bis 10-stellig. Wir erhalten durch iteriertes Quadrieren

$$x_1 = c_1^{d_1} \bmod p = 138 \bmod p$$

und

$$x_2 = c_2^{d_2} \bmod q = 250 \bmod q$$

Außerdem liefert der euklidische Algorithmus

$$1 = 218 \cdot 983 - 183 \cdot 1171$$

Wie in Satz A.5 setzen wir

$$t_0 = x_2 \cdot 218 \cdot 983 - x_1 \cdot 183 \cdot 1171 = -24000678$$

und erhalten

$$t_0 + 21 \cdot N = 172275$$

dh. $t_0 \bmod N = t$ ist wieder die ursprüngliche Nachricht.

Bemerkung 6.3. Zunächst scheint die Benutzung des chinesischen Restsatzes mit viel Arbeit verbunden zu sein, da z.B. eine Darstellung $1 = r \cdot p + s \cdot q$ ermittelt werden muss. Allerdings fällt dieser Aufwand nur einmal an. Da ein public-key-Verfahren in der Regel über einen längeren Zeitraum benutzt wird, lohnt es sich aber, diese vorbereitenden Rechnungen einmal durchzuführen und sie dann immer wieder zu benutzen.

RSA–Padding:

Die bis jetzt betrachtete Form des RSA–Verfahrens wird auch **Schulbuch–RSA** genannt. Diese Grundform hat jedoch einige Schwächen:

- Sie ist deterministisch.
- Sie ist multiplikativ.

Das das Verfahren deterministisch ist, bedeutet, dass ein Klartext immer zum selben Chiffra führt. Da RSA häufig benutzt wird, um z.B. PINs zu übertragen, und diese in der Regel sehr kurz sind, kann Catherine das für einen Angriff ausnutzen. Da sie den öffentlichen Schlüssel von Bob kennt, kann sie selbst Chiffre erzeugen, z.B. zu allen vierstelligen PINs. Dann kann sie das von Alice geschickte Chiffra mit dieser Liste vergleichen und daraus die PIN von Alice ablesen, ohne das Chiffra formal entschlüsselt zu haben.

Dass das Verfahren multiplikativ ist, bedeutet, dass für Klartexte $t_1, t_2 \in \mathbb{Z}/N\mathbb{Z}$ gilt

$$(t_1 \cdot t_2)^e = t_1^e \cdot t_2^e$$

und das wiederum bedeutet, dass

$$RSA(t_1 \cdot t_2) = RSA(t_1) \cdot RSA(t_2)$$

Diese Eigenschaften des Schulbuch–RSAs nennt man verformbar (**malleable**), und diese Eigenschaft kann Catherine ausnutzen. Sie hilft ihr zwar nicht bei der Entschlüsselung, aber sie ermöglicht es ihr, die Nachricht von Alice bewusst zu verfälschen. Wenn sie etwa das Chiffra c ($= RSA(t)$) von Alice abfängt, kann sie selbst mit dem public key von Bob einen Chiffrawert $c' = RSA(t')$ erzeugen und an Bob den Wert $c' \cdot c$ schicken. Da

$$c' \cdot c = RSA(t') \cdot RSA(t) = RSA(t' \cdot t)$$

ist, erhält Bob beim Entschlüsseln den Klartest $t^* = t' \cdot t$. Wenn Catherine also etwa weiß, dass Alice mit dem Chiffra einen Angebotspreis übermittelt, kann sie diesen bewusst modifizieren und daher die Chancen von Alice in einem Bieterverfahren reduzieren.

Die Verformbarkeit von Schulbuch–RSA kann von Catherine auch für einen chosen–ciphertext–Angriff benutzt werden. Bei einem chosen–ciphertext–Angriff geht man davon aus, dass Catherine ein „Orakel“ befragen kann, das ihr für eine Auswahl von ihr erzeugten Chiffren (die das zu entschlüsselnde Chiffra nicht enthalten) den Klartext liefert. Will Catherine das Chiffra c entschlüsseln, so wählt sie eine zu N teilerfremde Zahl $t_1 \in \mathbb{Z}/N\mathbb{Z}$ und füttet ihr Orakel mit dem Chiffra $c^* = RSA(t_1) \cdot c$. Hierfür

erhält sie von ihrem Orakel einen Klartext t^* , und sie erhält den gesuchten Klartext als $t = t_1^{-1} \cdot t^*$.

Um diese Schwächen des Schulbuch–RSAs auszugleichen, können diverse Padding–Verfahren benutzt werden. Eine Möglichkeit ist das **Optimal Asymmetric Encryption Padding OAEP**, das im Public Key Cryptography Standard PKCS#1 spezifiziert ist:

Wir nehmen an, dass die binäre Länge, des RSA–Moduls N gleich k ist, so dass also alle binären $k - 1$ –Tupel verschlüsselt werden können. Ferner nehmen wir an, dass der zu verschlüsselnde Klartext t lediglich n Bits belegt, wobei $l = k - 1 - n$ so groß ist, dass die Laufzeit eines Angriffs deutlich kleiner als 2^l sein muss. Ferner betrachten wir Einwegfunktionen $G : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^l$ und $H : \mathbb{F}_2^l \rightarrow \mathbb{F}_2^n$ (die allgemein bekannt sind). Dann funktioniert die Übertragung eines Klartextes t mit RSA–OAEP wie folgt

1. Alice wählt zufällig eine Zahl $r \in \mathbb{F}_2^n$.

2. Alice berechnet

$$\begin{aligned} x &= t + G(r) && \in \mathbb{F}_2^n \\ y &= r + H(t + G(r)) && \in \mathbb{F}_2^l \end{aligned}$$

(wobei die Additionen hier komponentenweise in \mathbb{F}_2^n bzw. \mathbb{F}_2^l durchzuführen sind) und setzt

$$m = x \| y$$

(die Konkatenation von x und y).

3. Alice berechnet $c = m^e \bmod N$ und schickt c an Bob.

4. Bob berechnet $m = c^d \bmod N$ und schreibt $m = x \| y$ mit $x \in \mathbb{F}_2^n$ und $y \in \mathbb{F}_2^l$.

5. Bob berechnet $H(x)$ und

$$y - H(x) = r + H(t + G(r)) - H(t + G(r)) = r$$

und hat damit die Zufallszahl r gefunden.

6. Bob berechnet $G(r)$ und

$$x - G(r) = t + G(r) - G(r) = t$$

und hat damit den Klartext t zurückgewonnen.

Beispiel 6.5. Wir benutzen wieder den RSA–Modul $N = 1\,151\,093$ mit öffentlichem Schlüssel $k_{\text{pub}} = (37, 1\,151\,093)$ und privatem Schlüssel $d = 931\,573$ aus Beispiel 6.1.

Statt mit Binärzahlen wollen wir aber mit Dezimalzahlen arbeiten. Mit diesem Verfahren können wir alle sechstelligen Zahlen verschlüsseln. Drei Stellen davon werden für den Klartext benutzt, drei für das Padding. Mit $M = (\mathbb{Z}/10 \cdot \mathbb{Z})^3$ bezeichnen wir die Menge aller dreistelligen Dezimalzahlen und definieren

$$G : M \longrightarrow M, \quad H : M \longrightarrow M$$

durch $G(x) = x^2 \bmod 1001$ (wobei wir beachten, dass 1000 kein Quadrat modulo 1001 ist) und $H(x) = x^2 \bmod 993$.

Alice will ihre Geheimzahl $t = 317$ mit diesem System an Bob schickt.

1. Alice wählt zufällig die Zahl $r = 724$.

2. Alice berechnet

$$\begin{aligned} x &= t + G(x) &= 317 + (724^2 \bmod 1001) &= 317 + 653 &= 960 \\ y &= r + H(t + G(x)) &= 724 + (963^2 \bmod 993) &= 724 + 96 &= 710 \end{aligned}$$

Beachten Sie dabei, dass die Addition komponentenweise in $(\mathbb{Z}/10 \cdot \mathbb{Z})^3$ durchgeführt wird, dass also etwa

$$724+96 = (7, 2, 4) + (0, 9, 6) = (7+0, 2+9, 4+6) = (7, 1, 0) = 710 \quad \text{in } (\mathbb{Z}/10 \cdot \mathbb{Z})^3$$

3. Alice setzt

$$m = x \| y = 960710$$

berechnet

$$c = m^e = 960710^{31} = 788239 \quad \bmod N$$

und schickt c an Bob.

4. Bob berechnet

$$m = 788239^{931573} = 960710$$

und spaltet diese Zahl auf in $x = 960$ und $y = 710$.

5. Bob berechnet

$$\begin{aligned} r &= y - H(x) &= 710 - (960^2 \bmod 993) &= 710 - 96 &= 724 \\ t &= x - G(r) &= 960 - (724^2 \bmod 1001) &= 960 - 653 &= 317 \end{aligned}$$

und hat damit die Geheimzahl t von Alice erhalten. Beachten Sie dabei, dass auch die Subtraktion komponentenweise in $(\mathbb{Z}/10 \cdot \mathbb{Z})^3$ erfolgt.

Angriffe:

Seit der Erstimplementierung des RSA–Verfahrens im Jahr 1977 wurden zahlreiche Angriffe gegen RSA vorgeschlagen. Bei den Ansätzen unterscheidet man dabei drei Grundtypen:

Protokollangriffe:

Dieser Typ von Angriff versucht Schwachstellen der Art und Weise, wie RSA eingesetzt ist, auszunutzen. Die bekanntesten Angriffe dieses Typs nutzen die Verformbarkeit oder die Deterministik von RSA aus. Durch richtig eingesetztes Padding können jedoch alle bekannten Protokollangriffe abgewehrt werden.

Mathematische Angriffe:

Mathematische Angriffe haben das Ziel aus den Daten (e, N) des öffentlichen Schlüssels den privaten Schlüssel d abzuleiten. Die bekanntesten davon zielen auf die Faktorisierung von N ab. Hier gab es seit Einführung von RSA im Jahr 1977 signifikante Fortschritte, aktuell gilt aber immer noch eine RSA–Modul der binären Länge 1024 als sicher gegen alle Faktorisierungsalgorithmen.

Seitenkanalangriffe:

Diese Art von Angriffen analysiert die technischen Seiten der Implementierung von RSA, etwa den Stromverbrauch. Hat Catherine Zugriff auf die Stromverbrauchskurve eines Prozessors, der ein RSA–Chiffrat entschlüsselt, so kann sie aus der Verlaufskurve des Stromverbrauchs auf die Struktur von d rückschließen. Die Berechnung von $c^d \bmod N$ wird durch einen Square–and–Multiply–Algorithmus realisiert, bei dem in einem Verarbeitungsschritt entweder nur quadriert oder quadriert und multipliziert wird. Der Stromverbrauch in einem Schritt, in dem quadriert und multipliziert wird, ist messbar höher als der in einem Schritt, in dem nur quadriert wird. Daraus kann Catherine schließen, ob an einer bestimmten Stelle in der Binärdarstellung eine 1 oder eine 0 steht und dadurch d ermitteln. Solche Angriffe können durch die Einführung von Dummy–Operationen oder durch Glättung des Stromverbrauchs durch zusätzliche Kondensatoren unterbunden oder zumindest erschwert werden.

Insgesamt sind die Verwendung und der Einsatz von RSA sehr genau spezifiziert. Unter Beachtung aller vorgeschlagenen Sicherheitsmaßnahmen gilt das Verfahren aktuell als sehr zuverlässig.

6.2. Das Rabin–Verschlüsselungsverfahren

Obwohl aktuell alle mathematischen Angriffe auf das RSA–Verfahren über die Faktorisierung des RSA–Moduls laufen, ist nicht bewiesen, dass dieser Weg notwendig ist und dass es keinen Ansatz gibt, den privaten Schlüssel d zu ermitteln, ohne die Faktoren von

N zu kennen. Anders ist das bei einem von Rabin vorgeschlagenen Verfahren.

Schlüsselerzeugung:

Bob wählt zwei Primzahlen p und q , wobei

$$p = 3 \pmod{4}, \quad q = 3 \pmod{4}$$

und berechnet $N = p \cdot q$. Dabei sollten p , q und N großenordnungsmäßig wie im RSA-Verfahren gewählt werden.

Bob veröffentlicht N als seinen öffentlichen Schlüssel. Die Primfaktorzerlegung $N = p \cdot q$ stellt seinen privaten Schlüssel dar.

Beispiel 6.6. Bob wählt die beiden Primzahlen $p = 1171$ und $q = 983$ (wie in Beispiel 6.1). Hierfür gilt

$$1171 = 3 \pmod{4}, \quad 983 = 3 \pmod{4}$$

Bob berechnet $N = p \cdot q = 1151093$ und veröffentlicht N als seinen öffentlichen Schlüssel.

Verschlüsselung:

Alice wählt einen Klartext $t \in \{0, \dots, N - 1\}$, berechnet das Chiffrat

$$c = t^2 \pmod{N}$$

und schickt c an Bob.

Beispiel 6.7. Wir greifen Beispiel 6.6 wieder auf. Alice verschlüsselt in diesem Kontext den Klartext $t = 317712$, berechnet

$$c = t^2 = 317712^2 = 418681 \pmod{N}$$

und schickt $c = 418681$ an Bob.

Entschlüsselung:

Bob empfängt c und berechnet zunächst

$$t_p = c^{\frac{p+1}{4}} \pmod{p}, \quad t_q = c^{\frac{q+1}{4}} \pmod{q}$$

Dadurch hat er eine Quadratwurzel aus c sowohl modulo p als auch modulo q ermittelt. Mit dem chinesischen Restsatz erhält er daraus eine Quadratwurzel t^* von c modulo N .

Warnung:

Quadratwurzeln modulo p und modulo q sind nicht eindeutig. Mit t_p ist auch $-t_p$ eine Wurzel aus $c \pmod{p}$ und mit t_q ist auch $-t_q$ eine Wurzel aus $c \pmod{q}$. Daraus ergeben sich im allgemeinen vier verschiedene Wurzeln aus c modulo N . Die von Bob gefundene

Lösung t^* entspricht also nicht unbedingt dem von Alice verschlüsselten Klartext. Deshalb muss Bob auch alle vier Wurzeln berechnen und daraus die richtige auswählen. Das kann dadurch erfolgen, dass der Klartext immer eine bestimmte Struktur haben muss, etwa an einer bestimmten Stelle eine vorgegebenen Textblock enthalten muss.

Beispiel 6.8. In Beispiel 6.7 hat Bob von Alice das Chiffraut $c = 418\,681$ erhalten. Er berechnet zunächst

$$c_1 = c \bmod p = 634, \quad c_2 = \bmod q = 906$$

Damit berechnet er

$$\begin{aligned} t_p &= c_1^{\frac{p+1}{4}} = 634^{293} = 371 \bmod p \\ t_q &= c_2^{\frac{q+1}{4}} = 906^{246} = 780 \bmod q \end{aligned}$$

Wir haben bereits ermittelt, dass

$$1 = 218 \cdot 983 - 183 \cdot 1171$$

Wie in Satz A.5 setzen wir daher

$$t_0 = t_p \cdot 218 \cdot 983 - t_q \cdot 183 \cdot 1171 = -87\,645\,466$$

und damit

$$t^* = t_1^* = t_0 \bmod N = 988\,695$$

Indem er nun alle Variationen von $\pm t_p$ und $\pm t_q$ durchspielt, erhält er noch drei weitere Wurzeln

$$t_2^* = 317\,712, \quad t_3^* = 162\,398, \quad t_4^* = 833\,381$$

Aufgrund geeigneter Zusatzbedingungen an den Klartext muss sich Bob jetzt für eine der vier Wurzeln entscheiden.

Bemerkung 6.4. Auch das Rabin–Verfahren sollte nur mit Padding verwendet werden, da es im Schulbuch–Format deterministisch und formbar und damit den gleichen Angriffen wie das Schulbuch–RSA ausgesetzt ist.

Das Rabin–Verfahren hat das Problem der Mehrdeutigkeit der Wurzeln, das durch entsprechende Einschränkungen an die Struktur des Klartextes gelöst werden muss, seine Sicherheit gegen mathematische Angriffe ist aber beweisbar äquivalent zum Faktorisierungsproblem:

Satz 6.2. Kann Catherine aus jedem Quadrat $c \in \mathbb{Z}/N\mathbb{Z}$ eine Quadratwurzel $Q(c) \in \mathbb{Z}/N\mathbb{Z}$ bestimmen, so kann Catherine auch die Faktorisierung von N ermitteln.

Beweis: Catherine wählt zufällig eine Zahl $x \in \{1, \dots, N-1\}$ aus. Falls $\text{ggT}(x, N) > 1$, so hat Catherine den gesuchten Teiler von N . Andernfalls berechnet sie

$$c = x^2 \pmod{N}, \quad y = Q(c)$$

Dann ist y , genauso wie x , eine Quadratwurzel von c , die beiden stimmen aber nicht notwendig überein sondern es tritt einer der folgenden vier Fälle auf:

$$y = x \pmod{p} \quad \text{und} \quad y = x \pmod{q} \tag{6.1}$$

$$y = -x \pmod{p} \quad \text{und} \quad y = x \pmod{q} \tag{6.2}$$

$$y = x \pmod{p} \quad \text{und} \quad y = -x \pmod{q} \tag{6.3}$$

$$y = -x \pmod{p} \quad \text{und} \quad y = -x \pmod{q} \tag{6.4}$$

Dabei entsprechen die Fälle (6.1) bzw. (6.4) dem Fall $x = y$ bzw. $x = -y \pmod{N}$. Daher gilt im Fall (6.1) $p|(y-x)$ und $q|(y-x)$, also $\text{ggT}(y-x, N) = N$ und im Fall (6.4) $p \nmid (y-x)$ und $q \nmid (y-x)$, also $\text{ggT}(y-x, N) = 1$, während im Fall (6.2) $p|(y-x)$ und $q \nmid (y-x)$, also $\text{ggT}(y-x, N) = p$ und im Fall (6.3) $p \nmid (y-x)$ und $q|(y-x)$, also $\text{ggT}(y-x, N) = q$. In den Fällen (6.2) und (6.3) hat Catherine also die Faktorisierung von N gefunden.

Da jeder Fall gleich oft auftritt, wird Catherine schließlich sicher die Zahl N zerlegen können. Mit hoher Wahrscheinlichkeit reichen dafür sogar schon recht wenige Versuche.

Bemerkung 6.5. In einem endlichen Körper \mathbb{F}_p ist das Ziehen einer Quadratwurzel aus einer Quadratzahl algorithmisch sehr einfach durchzuführen, in einem allgemeinen Ring $\mathbb{Z}/N\mathbb{Z}$ gibt es dagegen kein bekanntes Verfahren dafür, dass keine genauere Kenntnis über N und seine Primfaktoren voraussetzt.

7. Asymmetrische Verfahren basierend auf dem diskreten Logarithmusproblem

Neben den im Abschnitt 6 vorgestellten public-key-Verschlüsselungsverfahren, deren Sicherheit auf der Schwierigkeit der Primfaktorzerlegung großer Zahlen beruht, gibt es noch eine große Klasse von public-key-Verfahren, deren Sicherheit auf dem diskreten Logarithmus-Problem (vergleiche Anhang G) beruht.

7.1. Diffie–Hellman–Schlüsselaustausch DHKE

Eines der großen ungeklärten Probleme der symmetrischen Verschlüsselungsverfahren ist der geheime Austausch der Schlüssel. Zu Beginn der Kommunikation müssen sich Alice und Bob auf einen Schlüssel für ihren Nachrichtenaustausch verstündigen und für die Sicherheit der Verfahren ist es auch erforderlich, die Schlüssel in regelmäßigen Abständen zu ändern um einem Angreifer die Analyse der Struktur der Kommunikation zu erschweren.

Diese Frage adressiert das von Whitfield Diffie und Martin Hellman im Jahr 1976 publizierte **Diffie–Hellman–Verfahren DHKE** (oft auch **Diffie–Hellman–Merkle–Verfahren** in Anerkennung von Beiträgen von Ralph Merkle genannt) zum Schlüsselaustausch.

Vorbereitung:

Bob und Alice erzeugen ein Schlüsselpaar (p, g) wie folgt:

1. Bob und Alice einigen sich auf eine große Primzahl p .
2. Bob und Alice einigen sich auf eine Zahl $g \in \{1, \dots, p-1\}$, die die Einheitengruppe $E(\mathbb{F}_p)$ erzeugt (oder zumindest eine sehr hohe Ordnung in dieser Gruppe hat).

Das Paar (p, g) kann öffentlich bekannt sein.

Beispiel 7.1. Bob und Alice einigen sich auch die Primzahl $p = 2027$ und auf den Erzeuger $g = 7$ der zyklischen Gruppe $E(\mathbb{F}_p)$. Das Paar $(2027, 7)$ ist die Basis ihres Schlüsselaustauschverfahrens.

Schlüsselaustausch:

Bob und Alice einigen sich auf einen gemeinsamen Schlüssel K wie folgt:

1. Alice wählt zufällig eine Zahl $a \in \{1, \dots, p-1\}$ und berechnet $\alpha = g^a \pmod{p}$ ($k_{\text{pr},A} = a$ ist der private Schlüssel von Alice, $k_{\text{pub},A} = \alpha$ ist ihr öffentlicher Schlüssel).
2. Alice schickt α an Bob.
3. Bob wählt zufällig eine Zahl $b \in \{1, \dots, p-1\}$ und berechnet $\beta = g^b \pmod{p}$ ($k_{\text{pr},B} = b$ ist der private Schlüssel von Bob, $k_{\text{pub},B} = \beta$ ist sein öffentlicher Schlüssel).
4. Bob schickt β an Alice.

Aus diesen Daten α und β berechnen Alice und Bob ihren neuen gemeinsamen Schlüssel K wie folgt:

1. Alice berechnet

$$\beta^a = (g^b)^a = g^{b \cdot a} \pmod{p}$$

2. Bob berechnet

$$\alpha^b = (g^a)^b = g^{a \cdot b} \pmod{p}$$

3. Alice und Bob nutzen aus, dass

$$\beta^a = g^{b \cdot a} = g^{a \cdot b} = \alpha^b$$

und benutzen

$$k = \beta^a = \alpha^b$$

als gemeinsamen Schlüssel.

Beispiel 7.2. Wir greifen die Situation aus Beispiel 7.1 mit $p = 2027$ und $g = 7$ noch einmal auf.

1. Alice wählt $a = 1213$ und berechnet

$$\alpha = g^a = 7^{1213} = 442 \pmod{p}$$

2. Alice schickt $\alpha = 442$ an Bob.

3. Bob wählt $b = 1531$ und berechnet

$$\beta = g^b = 7^{1531} = 1455 \pmod{p}$$

4. Bob schickt $\beta = 1455$ an Alice.

Daraus leiten Alice und Bob nun ihren gemeinsamen Schlüssel ab:

1. Alice berechnet

$$k_A = \beta^a = 1455^{1231} = 803 \mod p$$

2. Bob berechnet

$$k_B = \alpha^b = 442^{1531} = 803 \mod p$$

3. Alice und Bob führen ihre Kommunikation mit einem symmetrischen Verfahren mit dem gemeinsamen Schlüssel

$$k = k_A = k_B = 803$$

durch.

Bemerkung 7.1. Die Frage nach der Sicherheit des Diffie–Hellman–Verfahren ist äquivalent zur Frage, ob aus $\alpha = g^a$ und $\beta = g^b$ die Zahl $\gamma = g^{a \cdot b}$ effizient berechnet werden kann (*Diffie–Hellman–Problem*).

Kann ein Angreifer das diskrete Logarithmus–Problem modulo p lösen, so kann er offensichtlich das Diffie–Hellman–Problem lösen. Andere Angriffe auf das Diffie–Hellman–Verfahren sind aktuell nicht bekannt, es ist aber nicht bewiesen, ob das Diffie–Hellman–Problem tatsächlich äquivalent zum diskreten Logarithmus–Problem ist.

Bemerkung 7.2. Etwas einfacher als das Diffie–Hellman–Problem scheint das *Diffie–Hellman–Entscheidungsproblem* zu sein. Dabei geht es um die Frage, ob ein Angreifer bei Vorliegen eines Tripels (α, β, γ) modulo p mit $\alpha = g^a$ und $\beta = g^b$ effizient entscheiden kann, ob $\gamma = g^{a \cdot b}$. Es ist klar, dass ein Angreifer, der das Diffie–Hellman–Problem lösen kann, auch das Diffie–Hellman–Entscheidungsproblem lösen kann. Ob davon auch die Umkehrung gilt, ist nicht bekannt.

Auch für das Diffie–Hellman–Entscheidungsproblem ist kein effizienter Angriff bekannt (der nicht über den diskreten Logarithmus geht), allerdings ist es möglich, mit vergleichsweise einfachen Überlegungen zu quadratischen Resten die Wahrscheinlichkeit für eine korrekte Aussage zu erhöhen.

Bemerkung 7.3. Notwendig für die Sicherheit des Diffie–Hellman–Verfahrens ist auf jeden Fall die Wahl einer sehr großen Primzahl p . Das alleine reicht jedoch noch nicht aus. Treten nämlich in der Primfaktorzerlegung von $p - 1$ nur kleine Primzahlen auf, so lässt sich der diskrete Logarithmus einer Zahl $z \in E(\mathbb{F}_p)$ mit dem Pohlig–Hellman–Algorithmus relativ schnell bestimmen. Die Primzahl $p_1 = 2027$ aus Beispiel 7.1 ist daher für den DHKE sehr viel besser geeignet als es die Primzahl $p_2 = 2017$, denn

$$p_1 - 1 = 2026 = 2 \cdot 1013, \quad p_2 - 1 = 2016 = 2^5 \cdot 3^2 \cdot 7$$

für p_2 ist das diskrete Logarithmus–Problem also sehr viel einfacher als für p_1 .

Aktuell sollte daher $p - 1$ mindestens einen Primteiler der binären Länge 224 enthalten um sicher gegen Pohlig–Hellman–Angriffe zu sein.

In der praktischen Anwendung sind weitere technische Bedingungen zu beachten. So sollte es etwa kein Polynom mit kleinen ganzzahligen Koeffizienten geben, dass sich über \mathbb{Q} nicht als Produkt von zwei Polynomen schreiben lässt, modulo p aber eine Nullstelle hat, denn dann kann der diskrete Logarithmus relativ effizient mit einem Zahlkörpersieb bestimmt werden.

Bemerkung 7.4. Im Vorbereitungsschritt des Verfahrens wurde g zunächst als Erzeuger der zyklischen Gruppe $E(\mathbb{F}_p)$ gewählt. Ein solches Element ist nicht immer einfach zu finden, un es ist jedoch nicht unbedingt erforderlich, mit einem solchen g zu arbeiten. Es reicht, ein Element $g \in E(\mathbb{F}_p)$ zu wählen, dass eine sehr hohe Ordnung hat und dessen Ordnung von einer großen Primzahl geteilt wird. Häufig wird daher nicht mit der ganzen zyklischen Gruppe $E(\mathbb{F}_p)$ gearbeitet sondern mit einer Untergruppe $U \subseteq E(\mathbb{F}_p)$ hinreichend hoher Primzahlordnung. Das scheint auch die Angriffe auf des Diffie–Hellman–Entscheidungsproblem zu erschweren.

Bemerkung 7.5. Besonders gut geeignet für den Diffie–Hellman–Schlüsselaustausch sind Primzahlen p mit $p - 1 = 2 \cdot q$ mit einer Primzahl $q \geq 3$. Dadurch wird zunächst sichergestellt, dass $p - 1$ einen sehr großen Primteiler hat. Außerdem ist jedes Element $g \in E(\mathbb{F}_p)$ mit $g^2 \neq 1$ schon ein Element der Ordnung q bzw. $2 \cdot q$, also gut geeignet für DHKE.

Will man mit einem Element der Ordnung q arbeiten (um auch eventuelle Angriffe auf das Diffie–Hellman–Entscheidungsproblem zu erschweren), so reicht es, g durch g^2 zu ersetzen. Dieses Element hat authomatisch die Ordnung q (egal, ob g selbst die Ordnung q oder $2 \cdot q$ hatte).

Das diskrete Logarithmus–Problem für g und für g^2 ist dabei gleich schwer.

Bemerkung 7.6. Eine praktische Vorgehensweise zur Bestimmung eines Paars (p, g) mit einer k –Bit–Primzahl p , sodass das Element $g \in E(\mathbb{F}_p)$ eine Ordnung hat, die von einer hinreichend k_1 –Bit–Primzahl q geteilt wird, ist die folgende:

1. Bob bestimmt eine Primzahl q der gewünschten Länge von k_1 Bit (vergleiche Anhang E).
2. Bob wählt zufällig eine $k - k_1$ –Bit–Zahl m aus und setzt $p = m \cdot q + 1$.
 - p ist eine Primzahl, → STOPP.

- p ist keine Primzahl \rightarrow gehe zu (1).

Sind die Primzahlen q und $p = m \cdot q + 1$ gefunden, so kann g wie folgt bestimmt werden:

1. Bob wählt zufällig eine Zahl $x \in \{2, \dots, p-2\}$.
2. Bob berechnet $g = x^m$.
 - Falls $g \neq 1 \pmod{p}$, \rightarrow STOPP.
 - Falls $g = 1 \pmod{p}$, \rightarrow gehe zu (1).

Auf diese Art und Weise erhält Bob auf jeden Fall ein Element der Ordnung q . Da nämlich

$$g^q = x^{mq} = x^{p-1} = 1$$

nach dem Satz von Fermat hat g entweder die Ordnung 1 oder die Ordnung q . Falls also $g \neq 1$, so hat g die Ordnung q .

Bemerkung 7.7. Neben der direkten Ableitung des Schlüssels k aus den Zahlen α und β hat Catherine noch eine weitere, sehr viel einfache Methode, die Kommunikation zwischen Alice und Bob mitzuhören:

Da sie das Paar (p, g) kennt, kann sie mit Alice in Verbindung treten, sich als Bob ausgeben und mit Alice einen Schlüssel k_{AC} vereinbaren. Gleichzeitig kann sie mit Bob in Verbindung treten, sich ihm gegenüber als Alice ausgeben und mit ihm eine Schlüssel k_{CB} vereinbaren. Alice und Bob glauben jeweils, miteinander einen Schlüssel ausgetauscht zu haben.

Schickt nun Alice eine mit k_{AC} verschlüsselte Nachricht an Bob, so wird diese von Catherine abgefangen, mit dem ihr bekannten Schlüssel k_{AC} entschlüsselt und gelesen und dann mit k_{CB} wieder verschlüsselt und an Bob weitergeleitet. So kann Catherine den Datenverkehr mitlesen, ohne dass Alice und Bob dies bemerken.

Diese Art des Angriffs bezeichnet man als **Man–In–The–Middle–Attacke**. Um diese Art von Angriff abzuwehren, müssen Alice und Bob in der Lage sein, zu verifizieren, ob die ursprüngliche Nachricht tatsächlich vom jeweils anderen stammt. Das kann etwa über digitale Signaturen erfolgen.

7.2. Das ElGamal–Verschlüsselungsverfahren

Das Diffie–Hellman–Verfahren eignet sich sehr gut, um Schlüssel auszutauschen, liefert aber keine Möglichkeit, eine Nachricht geheim zu übermitteln. Das ist durch eine von Taher ElGamal im Jahr 1985 vorgeschlagene Erweiterung der Diffie–Hellman–Methode

möglich. Grundlage ist wiederum eine (sehr große Primzahl p) und eine Element $g \in E(\mathbb{F}_p)$, das entweder diese Gruppe erzeugt oder zumindest eine sehr hohe Ordnung in dieser Gruppe hat und dessen Ordnung selbst einen sehr großen Primteiler q hat.

Schlüsselerzeugung:

Bob erzeugt ein Schlüsselpaar $((p, g, B), b)$ wie folgt:

1. Bob wählt eine sehr große Primzahl p , für die $p - 1$ einen großen Primteiler q hat.
2. Bob wählt eine Zahl $g \in \{2, \dots, p - 2\}$ aus, dessen Ordnung in $E(\mathbb{F}_p)$ von q geteilt wird.
3. Bob wählt zufällig eine Zahl $b \in \{2, \dots, p - 2\}$ aus und berechnet $B = g^b \pmod{p}$ und $d = p - 1 - b$.

Mit diesen Daten ist $k_{\text{pub}} = (p, g, B)$ der öffentliche Schlüssel von Bob und $k_{\text{pr}} = d$ sein privater Schlüssel.

Beispiel 7.3. Bob wählt die Primzahl $p = 2027$ (für die $p - 1$ den großen Primteiler $q = 1013$ hat) und die Zahl $g = 49$ (da die Zahl 7 ein Erzeuger von $E(\mathbb{F}_p)$ ist ist $g = 7^2$ ein Element der Ordnung $q = 1013$). Ferner wählt er zufällig die Zahl $b = 714$ und berechnet

$$B = g^b = 49^{714} = 763 \pmod{p}$$

Damit veröffentlicht Bob den Schlüssel $k_{\text{pub}} = (2027, 49, 763)$. Sein privater Schlüssel ist $d = p - 1 - b = 1312$.

Verschlüsselung:

Alice verschlüsselt einen Klartext $t \in \{1, \dots, p - 1\}$ mit Bobs öffentlichem Schlüssel (p, g, B) wie folgt:

1. Alice wählt zufällig ein $a \in \{0, \dots, p - 1\}$.
2. Alice berechnet $A = g^a \pmod{p}$.
3. Alice berechnet $C = B^a \pmod{p}$.
4. Alice berechnet $c = C \cdot t \pmod{p}$.
5. Alice schickt das Paar (A, c) an Bob.

Beispiel 7.4. Wir greifen Beispiel 7.3. Alice will mit Bobs öffentlichen Schlüssel $(2027, 49, 763)$ den Klartext $t = 1717$ verschlüsseln und an Bob schicken.

1. Alice wählt zufällig die Zahl $a = 1423 \in \{0, \dots, 2026\}$.
2. Alice berechnet $A = g^a = 49^{1423} = 758 \pmod{2027}$.
3. Alice berechnet $C = B^a = 1962^{1423} = 1699 \pmod{2027}$.
4. Alice berechnet $c = C \cdot t = 1699 \cdot 1717 = 330 \pmod{p}$.
5. Alice schickt das Paar $(A, c) = (758, 330)$ an Bob.

Entschlüsselung:

Bob entschlüsselt das Chiffrat (B, c) mithilfe seines privaten Schlüssels d wie folgt:

1. Bob berechnet $D = A^d \pmod{p}$.
2. Bob berechnet $t = D \cdot c \pmod{p}$.

Auf diesen Weg hat Bob tatsächlich die Nachricht entschlüsselt, denn es gilt

$$\begin{aligned} A^d \cdot c &= (g^a)^{p-1-b} \cdot (g^b)^a \cdot t \\ &= g^{a \cdot (p-1-b) + b \cdot a} \cdot t \\ &= g^{a \cdot (p-1) - a \cdot b + b \cdot a} \cdot t \\ &= (g^{p-1})^a \cdot t \\ &= t \end{aligned}$$

denn $g^{p-1} = 1$.

Beispiel 7.5. Wir führen Beispiel 7.5 fort. Bob hat von Alice das Paar $(758, 330)$ empfangen und will das Chiffrat nun mit seinem privaten Schlüssel $d = 1312$ entschlüsseln.

1. Bob berechnet $D = A^d = 758^{1312} = 859 \pmod{2027}$.
2. Bob berechnet $t = D \cdot c = 859 \cdot 330 = 1717 \pmod{2027}$.

Bemerkung 7.8. Die Funktionsweise des ElGamal-Verfahrens beruht darauf, dass Alice mit der Übertragung des Chiffrats gleichzeitig auch noch einen Schlüsselaustausch durchführt. Aus der von Alice übermittelten Zahl A kann Bob via

$$C = A^b = (g^a)^b = g^{a \cdot b}$$

den gemeinsamen Schlüssel $g^{a \cdot b}$ berechnen, den Alice selbst schon in der Form

$$C = B^a = (g^b)^a = g^{a \cdot b}$$

ermittelt und auch gleich benutzt hat, um t in der Form $c = C \cdot t$ zu verschlüsseln.

Da Bob den gemeinsamen Schlüssel C kennt, kann er C^{-1} ermitteln und damit c entschlüsseln. Die einfachste Form der Bestimmung von C^{-1} ist aber nicht über die Berechnung von $C = A^b$ und dann die Ermittlung des Inversen (etwa über Euklid) sondern die Berechnung von

$$A^{(p-1-b)} = g^{(p-1-b) \cdot a} = (g^{p-1})^a \cdot g^{-b \cdot a} = g^{-a \cdot b} = C^{-1}$$

da $g^{p-1} = 1$.

Bemerkung 7.9. Jeder Angriff auf das Diffie–Hellman–Schlüsselaustauschverfahren ist auch ein Angriff auf das ElGamal–Verfahren. Umgekehrt stellt auch jeder Angriff auf das ElGamal–Verfahren einen Angriff auf DHKE dar, denn wenn aus c der Klartext $t = C^{-1} \cdot c$ ermittelt werden kann, kann daraus auch der gemeinsame Schlüssel $C = c \cdot t^{-1}$ bestimmt werden.

8. Digitale Signaturen

Wie wir im letzten Abschnitt 7 gesehen haben, kann das von Diffie und Hellman entwickelte asymmetrische Verfahren genutzt werden, um den Schlüsselaustausch für symmetrische Kryptographie zu realisieren. Dabei hat sich allerdings herausgestellt, dass dieses Verfahren sehr anfällig für Man–In–The–Middle–Attacken ist, denn wir haben bis jetzt noch keine Möglichkeit, zu überprüfen, ob eine Nachricht tatsächlich von der Person kommt, die behauptet, der Absender zu sein.

Diese Frage gibt es natürlich nicht nur in der digitalen Welt. Auch in der analogen Kommunikation ist es notwendig, die Herkunft eines Dokuments zu verifizieren, was üblicherweise durch eine Unterschrift erfolgt. Eine Übertragung auf die digitale Kommunikation liefern die digitalen Signaturen, die bei digitalen Zertifikaten, etwa zur Absicherung von Webbrowsersn, eingesetzt werden, für bindende Signaturen digital abgeschlossener Verträge oder für die sichere Aktualisierung von Software.

Alle gängigen digitalen Signaturen beruhen auf asymmetrischen kryptographischen Verfahren. Das grundsätzliche Prinzip ist dabei relativ einfach zu beschreiben:

Alice will einen (verschlüsselten) Text c an Bob schicken und digital unterschreiben. Sie benutzt dazu eine asymmetrisches Verschlüsselungsverfahren, mit dem sie ein Schlüsselpaar $(k_{\text{pr},A}, k_{\text{pub},A})$ erzeugt. Den öffentlichen Schlüssel $k_{\text{pub},A}$ gibt sie bekannt, sodass in auf jeden Fall auch Bob kennt, den privaten Schlüssel $k_{\text{pr},A}$ hält sie wie üblich geheim. Mit ihrem privaten Schlüssel $k_{\text{pr},A}$ verschlüsselt sie nun ihren Signaturtext sig und erzeugt daraus ihre digitale Signatur dig_sig . An Bob schickt sie jetzt das Paar $(c, \text{dig_sig})$. Da Bob den öffentlichen Schlüssel von Alice kennt, kann er damit dig_sig zu sig entschlüsseln. Da Alice die einzige Person ist, die den privaten Schlüssel dieses Schlüsselpaares kennt, kann die Signatur nur von ihr erzeugt worden sein. Dass auch Catherine die Signatur überprüfen kann, ist hier kein Problem, da sie dadurch lediglich feststellen kann, dass die Signatur tatsächlich von Alice erzeugt wurde, daraus aber noch keine Information über das gesendete Chiffraut erhält.

Dieser Ansatz wirft natürlich unmittelbar einige Fragen auf. Die Signatur dig_sig als solches kann zwar nur von Alice erzeugt werden, aber bezieht sich die Signatur tatsächlich auf die Nachricht c ? Würde Alice etwa nur ihren Namen in ihrer digitalen Signatur verschlüsseln, so kann Catherine das Paar $(c, \text{dig_sig})$ abfangen, den Text c durch eine andere Nachricht c' ersetzen und $(c', \text{dig_sig})$ an Bob schicken. Bob würde dann immer noch einen korrekt signierten Text erhalten und Catherine könnte ihm damit beliebige Texte (die sie z.B. mit dem öffentlichen Schlüssel von Bob erzeugt hat) unterschieben. Entsprechend könnte auch Alice später behaupten, dass sich ihre Signatur auf einen ganz anderen Text c' bezogen hat. Daher müssen alle digitalen Signaturverfahren den

signierten Text c mit in die Signatur dig_sig miteinfließen lassen, sodass einwandfrei nachgewiesen werden kann, dass Alice genau diese Nachricht c signiert hat. Dadurch werden zwei Sicherheitsaspekte abgedeckt:

1. **Identifikation:** Bob kann sich sicher sein, dass tatsächlich Alice diese Nachricht geschickt hat.
2. **Beweisbarkeit:** Alice kann nicht abstreiten, dass sie tatsächlich diese Nachricht geschickt hat.

8.1. RSA-Signatur

Das Verfahren der RSA-Signatur beruht auf der RSA-Verschlüsselung, wie sie im Abschnitt 6.1 beschrieben wurde. Wir nehmen dazu an, dass Bob eine Nachricht c (die schon im chiffrierten Format vorliegt) an Alice schicken und signieren möchte. Ferner nehmen wir an, dass diese Nachricht c einen Klartext in dem von Bob benutzten RSA-Verfahren bildet.

Schlüsselerzeugung:

Alice erzeugt ein Schlüsselpaar $(k_{\text{pr},A}, k_{\text{pub},A}) = ((e_A, N_A), d_A)$ genauso wie beim RSA-Verfahren in Abschnitt 6.1.

Beispiel 8.1. Alice wählt die beiden Primzahlen $p = 1153$ und $q = 1019$ und berechnet

$$N_A = p \cdot q = 1174\,907.$$

$$\varphi(N_A) = (p - 1) \cdot (q - 1) = 1172\,736.$$

Für e wählt Alice die Zahl $e_A = 53$ und leitet aus dem euklidischen Algorithmus die Beziehung

$$1 = (-21) \cdot 1172\,736 + 464\,669 \cdot 53$$

ab. Daher bestimmt sie $d_A = 464\,669$ als ihren privaten Schlüssel.

Alice publiziert ihren öffentlichen Schlüssel $k_{\text{pub},A} = (53, 1174\,907)$. Den privaten Schlüssel $d_A = 464\,669$ behält sie für sich.

Signatur durch Alice:

Die zu signierende Nachricht c liegt als Klartext $c \in \mathbb{Z}/N\mathbb{Z}$ vor. Alice signiert die Nachricht c nun wie folgt:

1. Alice benutzt ihren privaten Schlüssel d_A und berechnet $\text{sig}_A(c) = c^{d_A} \bmod N_A$.

2. Alice schickt das Paar $(c, \text{sig}_A(c))$ über einen öffentlichen Kanal an Bob.

Beispiel 8.2. Wir nehmen jetzt an, dass Bob selbst auch ein RSA-Verfahren aufgesetzt hat, und zwar wie in Beispiel 6.1 mit dem öffentlichen Schlüssel $k_{\text{pub},B} = (37, 1\,151\,093)$ und dem privaten Schlüssel $d_B = 931\,573$.

Alice will den Klartext $t = 972\,486$ sicher und signiert an Bob schicken. Dazu benutzt sie zunächst den öffentlichen Schlüssel $(e_B, N_B) = (37, 1\,151\,093)$ von Bob und berechnet

$$c = t^{e_B} = 972\,486^{37} = 1\,124\,183 \mod 1\,151\,093$$

Nun benutzt sie ihren privaten Schlüssel $d_A = 464\,669$ und berechnet

$$\text{sig}_A(c) = c^{d_A} = 1\,124\,183^{464\,669} = 325\,413 \mod 1\,174\,907$$

Das Paar $(c, \text{sig}_A(c)) = (1\,124\,183, 325\,413)$ schickt sie jetzt an Bob.

Verifikation durch Bob:

Bob hat jetzt ein Paar (c, s) erhalten. Mithilfe des öffentlichen Schlüssels (e_A, N_A) von Alice überprüft er, ob es sich bei s um die Signatur $\text{sig}_A(c)$ von Alice handelt indem er vorgeht wie folgt:

1. Bob berechnet $\tilde{c} = s^{e_A} \mod N_A$.
2. Bob akzeptiert die Signatur, falls $\tilde{c} = c$, andernfalls lehnt er sie ab.

Bemerkung 8.1. Das Verfahren funktioniert korrekt, denn da $d_A \cdot e_A = 1 \mod N_A$, ist

$$(\text{sig}_A(c))^{e_A} = (c^{d_A})^{e_A} = c^{d_A \cdot e_A} = c^1 = c \mod N_A$$

dh. falls s die Signatur von Alice ist, entscheidet Bob korrekt.

Beispiel 8.3. In Beispiel 8.2 hat Bob das Paar $(1\,124\,183, 464\,669)$ empfangen und will jetzt überprüfen, ob die Nachricht tatsächlich von Alice signiert wurde. Dazu benutzt er den öffentlichen Schlüssel $(e_A, N_A) = (53, 1\,174\,907)$ von Alice und berechnet

$$\tilde{c} = 464\,669^{53} = 1\,124\,183 \mod 1\,174\,907$$

Er überprüft, dass tatsächlich $\tilde{c} = c$ und akzeptiert daher die Nachricht.

Nun benutzt er seinen privaten Schlüssel $d_B = 931\,573$ und berechnet

$$t = c^{d_B} = 1\,124\,183^{931\,573} = 972\,486$$

Damit hat er den Klartext zurückbekommen und kann sich sicher sein, dass diese Nachricht auch von Alice stammt. Darüberhinaus kann Alice nicht abstreiten, diesen Text geschickt zu haben, denn nur sie kann diese Signatur erzeugen.

Sicherheitsaspekte:

Dieses Verschlüsselungsprinzip wird **Schulbuch–RSA–Signatur** genannt. Wie das Schulbuch–RSA–Verfahren selbst hat es einige entscheidende Nachteile. Der wichtigste ist die existentielle Fälschung. Dabei fängt Catherine die signierte Nachricht von Alice ab und schiebt Bob eine andere Nachricht als von Alice signiert unter, und zwar wie folgt:

1. Catherine wählt eine Signatur $s \in Z/N_A\mathbb{Z}$ aus.
2. Catherine berechnet $c = s^{e_A} \bmod N_A$ und schickt (c, s) an Alice.

Bob wird $\tilde{c} = s^{e_A} \bmod N_A$ berechnen, feststellen, dass es sich dabei um c handelt und wird die Nachricht als von Alice signiert akzeptieren. Catherine kann zwar in fder Regel nicht kontrollieren, was sie dadurch an Bob schickt, sie kann aber auf jeden Fall Verwirrung stiften.

Beispiel 8.4. In Beispiel 8.2 fängt Catherine die Daten $(1\ 124\ 183, 464\ 669)$ ab uns wählt als Signatur $s = 764\ 356$. Sie berechnet

$$c_1 = s^{e_A} = 764\ 356^{53} = 92\ 922 \bmod 1\ 174\ 907$$

und schickt $(92\ 922, 764\ 356)$ an Bob.

Bob überprüft, dass

$$764\ 356^{53} = 92\ 922$$

und akzeptiert die Nachricht als von Alice signiert. Er entschlüsselt daher c und erhält

$$t_1 = c_1^{d_B} = 92\ 922^{931\ 573} = 772\ 799$$

Bob würde also mit der Nachricht t_1 weiterarbeiten.

Eine weitere Schwäche der Schulbuch–RSA–Signatur ist die Formbarkeit des RSA–Verfahrens. Kennt Catherine ein valides Nachrichten–Signatur–Paar (c_1, s_1) , und hat sie das Paar (c, s) von Bob abgefangen, so kann sie durch $(c_1 \cdot c, s_1 \cdot s)$ ein gültiges Nachrichten–Signatur–Paar erzeugen. Dadurch kann sie gegebenenfalls eine Nachricht bewusst in eine bestimmte Richtung verschieben. Falls es ihr also gelingt, zu gewissen Texten c_i korrekte Signaturen s_i zu bekommen, kann Sie daraus weitere korrekte Nachrichten–Signatur–Paare konstruieren (diese Art von Angriff wird auch **chosen–message–Angriff** genannt).

Diese Angriffe können dadurch vermieden werden, dass nur Nachrichten einer gewissen Struktur erlaubt sind, dass etwa der zugrundeliegende Klartext t immer von der Form $t = t_1 \| t_1$ sein muss. Es ist praktisch ausgeschlossen, dass diese Form auch bei der existentiellen Fälschung oder bei Ausnutzung der Formbarkeit erreicht wird.

8.2. ElGamal-Signatur

Dieses Signaturverfahren beruht auf dem ElGamal–Verschlüsselungsverfahren (das wieder auf dem Diffie–Hellman–Verfahren beruht). Wir nehmen dazu an, dass Bob eine Nachricht c (die schon im chiffrierten Format vorliegt) an Alice schicken und signieren möchte. Ferner nehmen wir an, dass diese Nachricht c einen Klartext in dem von Bob benutzten ElGamal–Verfahren bildet.

Schlüsselerzeugung:

Alice erzeugt ein Schlüsselpaar $(k_{\text{pr},A}, k_{\text{pub},A}) = ((p_A, g_A, A), a)$ genauso wie beim ElGamal–Verfahren in Abschnitt 7.2.

Beispiel 8.5. Alice wählt die Primzahl $p_A = 2459$. Hierfür ist $p_A - 1 = 2458 = 2 \cdot 1229$ die Primfaktorzerlegung, dh. $p_A - 1$ hat einen großen Primfaktor. Ferner wählt sie die Zahl $g_A = 3$ aus, die in $E(\mathbb{F}_{p_A})$ die Ordnung $q = 1229$ hat. Ferner wählt sie zufällig die Zahl $a = 729$ und berechnet

$$A = g_A^a = 3^{729} = 2270 \pmod{p_A}$$

Alice publiziert ihren öffentlichen Schlüssel $k_{\text{pub},A} = (2459, 3, 2270)$. Den privaten Schlüssel $a = 729$ behält sie für sich.

Signatur durch Alice:

Die zu signierende Nachricht c liegt als Klartext $c \in \{0, \dots, p_A - 1\}$ vor. Alice signiert die Nachricht c nun wie folgt:

1. Alice wählt zufällig eine Zahl $z \in \{2, \dots, p_A - 2\}$, die teilerfremd zu $p - 1$ ist.
2. Alice berechnet die zu z in $\mathbb{Z}/(p_A - 1)\mathbb{Z}$ inverse Zahl z^{-1} .
3. Alice berechnet
$$\begin{aligned} r &= g_A^z \pmod{p_A} \\ s &= (c - a \cdot r) \cdot z^{-1} \pmod{p_A - 1} \end{aligned}$$
4. Alice setzt $\text{sig}_A(c, z) = (r, s)$ und schickt $(c, (r, s))$ an Bob.

Beispiel 8.6. Wir nehmen jetzt an, dass Bob ein RSA–Verfahren aufgesetzt hat, und zwar mit dem öffentlichen Schlüssel $k_{\text{pub},B} = (17, 2279)$ und dem privaten Schlüssel $d_B = 257$.

Alice will den Klartext $t = 300$ sicher und signiert an Bob schicken. Dazu benutzt sie zunächst den öffentlichen Schlüssel $(e_B, N_B) = (17, 2279)$ von Bob und berechnet

$$c = t^{e_B} = 300^{17} = 1504 \pmod{2279}$$

Nun erzeugt Sie dazu eine Signatur wie folgt

1. Sie wählt zufällig die Zahl $z = 1457$ (die teilerfremd zu $p_A - 1 = 2458$ ist).

2. Sie erhält aus dem euklidischen Algorithmus die Beziehung

$$1 = 131 \cdot 2458 - 221 \cdot 1457$$

und daraus $z^{-1} = -221 = 2237 \pmod{p_A - 1}$.

3. Sie berechnet

$$\begin{aligned} r &= g_A^z &= 3^{1457} &= 2071 \pmod{p_A} \\ s &= (c - a \cdot r) \cdot z^{-1} &= (1504 - 729 \cdot 2071) \cdot 2237 &= 2349 \pmod{p_A - 1} \end{aligned}$$

4. Alice setzt $\text{sig}_A(c, z) = (r, s) = (2071, 2349)$.

Die Daten $(c, \text{sig}_A(c, z)) = (1504, (2071, 2349))$ schickt sie jetzt an Bob.

Verifikation durch Bob:

Bob hat jetzt ein Paar $(c, (r, s))$ erhalten. Mithilfe des öffentlichen Schlüssels (p_a, g_A, A) von Alice überprüft er, ob es sich bei (r, s) um die Signatur $\text{sig}_A(c, z)$ von Alice handelt indem er vorgeht wie folgt:

1. Bob berechnet den Wert $x = A^r \cdot r^s \pmod{p_A}$.
2. Bob berechnet $y = g_A^c \pmod{p_A}$
3. Bob akzeptiert die Signatur, falls $x = y$, andernfalls lehnt er sie ab.

Bemerkung 8.2. Das Verfahren funktioniert korrekt. Dazu beachten wir zunächst, dass jedes Element in $E(\mathbb{F}_{p_A})$ eine Ordnung hat, die ein Teiler von $p_A - 1$ ist. Da

$$s = (c - a \cdot r) \cdot z^{-1} \pmod{p_A - 1}$$

gilt

$$r^s = r^{(c-a \cdot r) \cdot z^{-1}}$$

und damit gilt (modulo p_A)

$$\begin{aligned} x &= A^r \cdot r^s &= A^r \cdot r^{(c-a \cdot r) \cdot z^{-1}} \\ &&= (g_A^a)^r \cdot (g_A^z) (c - a \cdot r) \cdot z^{-1} \\ &&= g_A^{a \cdot r + z \cdot (c-a \cdot r) \cdot z^{-1}} \\ &&= g_A^{a \cdot r + c - a \cdot r} \\ &&= g_a^c = y \end{aligned}$$

Ist die Nachricht also korrekt signiert, so ist $x = y$ und Bob entscheidet richtig.

Es ist möglich, dass $x = y$ gilt, wenn (r, s) keine korrekte Signatur von c ist, die Wahrscheinlichkeit hierfür ist aber extrem gering und kann vernachlässigt werden.

Beispiel 8.7. In Beispiel 8.6 hat Bob die Daten $(1504, (2071, 2349))$ empfangen und will jetzt überprüfen, ob die Nachricht tatsächlich von Alice signiert wurde. Dazu benutzt er den öffentlichen Schlüssel $(p_A, g_A, A) = (2459, 3, 2270)$ von Alice und berechnet

1. den Wert $x = A^r \cdot r^s = 2270^{2071} \cdot 2071^{2349} = 2284 \cdot 2021 = 421$
2. den Wert $y = g_A^c = 3^{1504} = 421$.

Er überprüft, dass tatsächlich $x = y$ ist und akzeptiert daher die Nachricht als von Alice signiert.

Nun benutzt er seinen privaten Schlüssel $d_B = 257$ und berechnet

$$t = c^{d_B} = 1504^{257} = 300 \pmod{2279}$$

Damit hat er den Klartext zurückbekommen und kann sich sicher sein, dass diese Nachricht auch von Alice stammt. Darüberhinaus kann Alice nicht abstreiten, diesen Text geschickt zu haben, denn nur sie kann diese Signatur erzeugen.

Bemerkung 8.3. Es ist essentiell für das Signaturverfahren, dass Alice für jede Signatur eine neue Zufallszahl z verwendet. Bei mehrfacher Verwendung derselben Zufallszahl kann nämlich Catherine das Signaturschema angreifen, und zwar wie folgt:

Catherine fängt zwei Datensätze $(c_1, (r_1, s_1))$ und $(c_2, (r_2, s_2))$ ab, die mit derselben Zufallszahl z erzeugt wurden. Dann ist

$$r_1 = g_A^z = r_2 =: r$$

(An dem Übereinstimmen der r -Werte erkennt Catherine auch, dass Alice die Zahl z wiederverwendet hat). Damit hat Alice nun zwei Gleichungen

$$\begin{aligned} s_1 &= (c_1 - a \cdot r) \cdot z^{-1} \pmod{p_A - 1} \\ s_2 &= (c_2 - a \cdot r) \cdot z^{-1} \pmod{p_A - 1} \end{aligned} \tag{8.1}$$

Daraus erhält sie

$$s_1 - s_2 = (c_1 - c_2) \cdot z^{-1} \pmod{p_A - 1} \tag{8.2}$$

Falls also $s_1 - s_2$ invertierbar module $p_A - 1$ ist, kann sie daraus

$$z = \frac{c_1 - c_2}{s_1 - s_2} \pmod{p_A - 1}$$

ermitteln. Wenn Sie z kennt, kann sie aber aus den Gleichungen (8.1) den privaten Schlüssel a von Alice ermitteln, etwa als

$$a = \frac{c_1 - z \cdot s_1}{r} \pmod{p_A - 1}$$

Falls $s_1 - s_2$ nicht invertierbar modulo $p_A - 1$ ist, so hat Gleichung (8.2) eventuell mehr als eine Lösung. Catherine kann dann immer noch versuchen, durch Ausprobieren unter den Lösungen die richtige zu finden, oder sie kann ihr Glück mit einem anderen Paar von Datensätzen versuchen, bei denen die Signatur ebenfalls mit diesem z erzeugt wurde.

Beispiel 8.8. Alice hat in Beispiel 8.6 die Zufallszahl $z = 1457$ genutzt, um das Chiffrat $c_1 = 1504$ zu signieren und hat daraus die Signatur

$$\text{sig}_A(c_1, z) = (r_1, s_1) = (2071, 2349)$$

erzeugt. Angenommen, sie benutzt jetzt die gleiche Zahl $z = 1457$ um auch das Chiffrat $c_2 = 1235$ zu signieren. Dann berechnet sie

$$\begin{aligned} r_2 &= g_A^z &= 3^{1457} &= 2071 \mod p_A \\ s_2 &= (c - a \cdot r) \cdot z^{-1} &= (1235 - 729 \cdot 2071) \cdot 2237 &= 348 \mod p_A - 1 \end{aligned}$$

Catherine fängt die Daten ab und sieht, dass $r_1 = r_2 = 2071 =: r$. Daran erkennt sie, dass Alice dasselbe z wiederverwendet hat. Sie berechnet

$$s_1 - s_2 = 2349 - 348 = 2001 \mod p_A - 1$$

und sieht, dass diese Zahl teilerfremd zu $p_A - 1$ ist. Mit dem euklidischen Algorithmus berechnet sie

$$(s_1 - s_2)^{-1} = 1705 \mod p_A - 1$$

und ermittelt daraus

$$z = \frac{c_1 - c_2}{s_1 - s_2} = 269 \cdot 1705 = 1457 \mod p_A - 1$$

Ferner erkennt sie, dass auch $r = 2071$ teilerfremd zu $p_A - 1$ ist und berechnet (wieder mit Euklid)

$$r^{-1} = 235$$

Daraus erhält sie

$$a = \frac{c_1 - z \cdot s_1}{r} = (1504 - 1457 \cdot 2349) \cdot 235 = 729 \mod p_A - 1$$

und hat damit das Signaturschema von Alice geknackt.

Bemerkung 8.4. Wie auch die Schulbuch–RSA–Signatur ist diese Grundform der ElGamal–Signatur, die auch **Schulbuch–ElGamal–Signatur** nicht geschützt vor existentiellen Fälschungen.

Catherine, die den öffentlichen Schlüssel (p_A, g_A, A) von Alice kennt, kann dabei wie folgt vorgehen.

1. Catherine wählt zwei natürliche Zahlen m und n , wobei n teilerfremd zu $p_A - 1$ ist und sie berechnet das Inverse n^{-1} zu n modulo $p_A - 1$.
2. Catherine berechnet
 - $r = g_A^m \cdot A^n \pmod{p_A}$.
 - $s = -r \cdot n^{-1} \pmod{p_A - 1}$.
und setzt $\text{sig} = (r, s)$.
3. Catherine berechnet $c = n \cdot s \pmod{p_A - 1}$.
4. Catherine sendet $(c, (s, r))$ an Bob.

Bob empfängt die Daten und berechnet

$$x = A^r \cdot r^s \pmod{p_A}, \quad y = g_A^c \pmod{p_A}$$

und er wird die Signatur akzeptieren, denn modulo p_A gilt

$$\begin{aligned} x = A^r \cdot r^s &= (g_A^a)^r \cdot (g_A^n A^m)^s \\ &= g_A^{a \cdot r} \cdot g_A^{n \cdot s} \cdot A^{m \cdot s} \\ &= g_A^{a \cdot r} \cdot g_A^{n \cdot s} \cdot g_A^{a \cdot m \cdot s} \\ &= g_A^{a \cdot r} \cdot g_A^{n \cdot s} \cdot g_A^{a \cdot m \cdot (-r \cdot n^{-1})} \\ &= g_A^{a \cdot r} \cdot g_A^{n \cdot s} \cdot g_A^{-a \cdot r} \\ &= g_A^{n \cdot s} = y \end{aligned}$$

Allerdings ist Catherine (wie schon bei der RSA-Signatur) nicht in der Lage, den Inhalt der signierten Nachricht zu kontrollieren. Wie beim RSA-Verfahren kann auch hier der Angriff dadurch erschwert werden, dass nur Nachrichten einer gewissen Form und Struktur zugelassen werden.

Bemerkung 8.5. Neben der Gefahr der existentiellen Fälschung haben die Schulbuch-RSA-Signatur und die Schulbuch-ElGamal-Signatur noch den weiteren entscheidenden Nachteil, dass die Länge der Nachricht, die damit signiert werden kann, sehr stark begrenzt ist. Das könnte man zwar dadurch umgehen, dass der zu signierende Text in geeignet kurze Blöcke aufgeteilt und jeder Block einzeln signiert wird, aber dieser Ansatz hat sich als unpraktikabel erwiesen.

Die Texte selbst werden nämlich in der Regel symmetrisch verschlüsselt, die Signaturen dagegen mit asymmetrischen Verfahren erzeugt. Asymmetrische Verfahren sind aber

sehr viel teurer und rechenintensiver als symmetrische Verfahren, die Kosten für die Signatur (und ihre Verifikation) wären also um Vielfaches teurer als die Kosten für die eigentliche Nachricht. Außerdem hat diese Vorgehensweise den Nachteil, dass Catherine Abschnitte des Textes entfernen könnte oder ihr bekannte signierte Textpassagen einschieben könnte, ohne dass das bei der Signaturüberprüfung auffallen würde.

In der Praxis wird dieses Problem dadurch gelöst, dass man nicht die ursprüngliche Nachricht c signiert sondern aus der Nachricht erst eine Art Kurzzusammenfassung h (einen Hashwert von c) erzeugt und nur diesen Hashwert signiert. Bei geschickter Konstruktion dieses Hashwerts kann dadurch sichergestellt werden, dass tatsächlich diese Nachricht signiert wurde. Darüberhinaus können dadurch auch existentielle Fälschungen verhindert werden.

Mit der Konstruktion von Hashwerten und der Ableitung rechtssicherer digitaler Signaturen beschäftigt sich der nächste Abschnitt.

9. Hash–Funktionen

Das Konzept der Hash–Funktionen spielt eine wichtige Rolle in vielen Kryptographischen Anwendungen und Protokollen (wie etwa SSL oder IPsec zur Überprüfung der Integrität einer Nachricht und zur Authentifizierung des Absenders). Gründe für die Notwendigkeit von Hash–Funktionen ergeben sich unmittelbar schon aus den Ausführungen über digitale Signaturen z.B. mit elliptischen Kurven.

9.1. Probleme bei digitalen Signaturen

Die gesendete Nachricht m ist bei der digitalen Signatur etwa mithilfe der RSA–Signatur immer eine wesentliche Komponente, die in die Signatur mit eingeht. Da Signaturen aber mit asymmetrischen Verfahren erstellt werden, ist dadurch die Länge der Nachricht begrenzt, bei RSA–basierten Signaturen üblicherweise auf 1024 bis 3072 Bits (also 128 bis 384 Bytes). Der zu versendende Klartext (und damit auch der symmetrisch verschlüsselte Nachrichtentext) ist jedoch in aller Regel sehr viel länger.

Ein Ansatz, dieses Problem zu lösen, wäre natürlich eine Aufteilung der Nachricht in Blöcke geeigneter Länge, die der Blocklänge des asymmetrischen Signierverfahrens entspricht, und eine separate Signatur jedes einzelnen Blocks. Dieses Verfahren hat aber einige entscheidende Nachteile.

Sehr hoher Rechenaufwand:

Digitale Signaturen beruhen auf asymmetrischen Verfahren und asymmetrische Verschlüsselung ist sehr rechenintensiv und erfordert signifikant mehr Arbeit als die symmetrische Verschlüsselung eines entsprechend langen Textes. Die Verschlüsselung mit RSA (bei den üblichen Schlüssellängen) etwa erfordert ca. hundertmal so viel Rechenzeit wie eine Verschlüsselung mit AES. Die Berechnung der Signatur (auf der Senderseite) und die Verfizierung der Signatur (auf der Empfängerseite) würde also einen wesentlich höheren Aufwand verursachen als die Ver– bzw. Entschlüsselung der Nachricht selbst.

Hoher Datenüberhang:

Da die Signatur so lang ist wie die Nachricht selbst (in manchen Fällen sogar länger), vergrößert sich das Datenvolumen mindestens auf das Doppelte.

Hohes Sicherheitsrisiko:

Die Aufteilung der Nachricht in einzelne Blöcke führt zu Sicherheitsrisiken. Catherine kann zwar die einzelnen Blöcke selbst nicht manipulieren, sie kann aber z.B. einzelne

Blöcke entfernen, Blöcke vertauschen oder aus aktuellen und alten Blöcken neue Nachrichten zusammenstellen. Möglicherweise sind die so entstandenen Nachrichten sinnlos, aber die Signatur der Nachricht wäre auf jeden Fall korrekt, da die Signatur blockweise erfolgt. Falls Catherine im Besitz verschlüsselter und signierter Klartexte ist, kann sie eventuell sogar Nachrichten bewusst verfälschen und sinnvolle Texte erzeugen, in jedem Fall aber kann sie Verwirrung stiften und Chaos erzeugen.

Vorgehen zur Lösung des Problems:

In die Signatur geht nicht die ganze Nachricht m sondern nur eine Kurzform $h(m)$ der Nachricht ein, dh. zunächst wird aus der gesamten Nachricht m , die aus vielen Blöcken besteht, eine Kurzform $h(m)$ erzeugt, die nur einen Block umfasst, der dann auch auf einmal signiert werden kann.

9.2. Hash–Funktionen und ihre Eigenschaften

Die Verkürzung der Ausgangsnachricht wird üblicherweise mithilfe von Hash–Funktionen realisiert.

Definition 9.1. Eine **Hash–Funktion** ist eine Abbildung

$$h : \mathbb{F}_2^\bullet \longrightarrow \mathbb{F}_2^n, \quad m \longmapsto h(m)$$

die einen beliebig langen Bitstring m auf einen Bitstring $h(m)$ einer fixierten Länge n reduziert.

Eine Hash–Funktion verkürzt also eine gegebene Nachricht m (beliebiger Länge) zu einem Block $h(m)$ einer Länge, die dann in einem Schritt mit dem gewählten digitalen Signaturverfahren signiert werden kann. Wichtig hierfür ist natürlich, dass $h(m)$ trotzdem noch ein sinnvoller „Repräsentant“ der ursprünglichen Nachricht ist. Eine Minimalanforderung ist etwa, dass mit Hilfe von $h(m)$ erkannt werden kann, ob bei der Übertragung von m Fehler aufgetreten sind.

Ein klassisches Beispiel für diese Art von Absicherungen gegen fehlerhafte Übertragung sind Paritätsprüfzeichen. Dabei wird an einem Bitstring ein weiteres Bit angehängt, und zwar so, dass die Anzahl der 1–Einträge in dem String insgesamt gerade ist, dh. ist $B = (b_1, \dots, b_m)$ gegeben, so ist b_{m+1} so zu wählen, dass

$$\sum_{i=0}^{m+1} b_i = 0 \quad \text{in } \mathbb{F}_2$$

Beispiel 9.1. Der String $B = (1, 0, 0, 1, 0, 0, 1)$ wird nach Ergänzung um ein Paritätsbit zu

$$\tilde{B} = (1, 0, 0, 1, 0, 0, 1, 1)$$

Ein anderes Beispiel für ein Kontrollzeichen ist die (klassische) Buchkennzeichnung ISBN–10. Dabei wird jedes Buch durch eine neunstellige Ziffernfolge a_1, \dots, a_9 charakterisiert. Um Übertragungsfehler zu vermeiden, wird ein zehntes Zeichen a_{10} angefügt, das sich nach der Formel

$$a_{10} = \sum_{k=1}^9 k \cdot a_k \quad \text{mod } 11$$

berechnet (wobei $a_{10} = X$ notiert wird, falls $a_{10} = 10$).

Anhand des Prüfzeichens kann festgestellt werden, ob bei der Übermittlung der Daten an einer Stelle ein Fehler aufgetreten ist. Es kann auch erkannt werden, ob zwei Positionen vertauscht wurden.

Beispiel 9.2. Wir betrachten ein Buch, dessen neun Informationsstellen die Gestalt 3–86680–192 haben.

1. $s = 3 + 2 \cdot 8 + 3 \cdot 6 + 4 \cdot 6 + 5 \cdot 8 + 6 \cdot 0 + 7 \cdot 1 + 8 \cdot 9 + 9 \cdot 2 = 198$.
2. $s = 18 \cdot 11$, Division geht ohne Rest auf.
3. $a_{10} = 0$

Die ISBN-10 Nummer dieses Buches ist also 3–86680–192–0.

Beispiel 9.3. Wir betrachten ein Buch, dessen neun Informationsstellen die Gestalt 3–680–08783 haben.

1. $s = 3 + 2 \cdot 6 + 3 \cdot 8 + 4 \cdot 0 + 5 \cdot 0 + 6 \cdot 8 + 7 \cdot 7 + 8 \cdot 9 + 9 \cdot 3 = 227$.
2. $s = 20 \cdot 11 + 7$.
3. $a_{10} = 7$

Die ISBN-10 Nummer dieses Buches ist also 3–680–08783–7.

Mithilfe des ISBN–10–Codes kann ein Fehler bei der Übertragung erkannt werden, und außerdem kann erkannt werden, wenn zwei Positionen vertauscht wurden.

Für kryptographische Anforderungen ist diese Art des Vorgehens allerdings viel zu schwach. Hier muss davon ausgegangen werden, dass Catherine bewusst und vorsätzlich Fehler herbeiführen will. In obigen Beispielen könnte sie dabei sehr leicht den Nachrichtenteil so abändern, dass das Prüfzeichen wieder passt. Aus diesen Überlegungen ergeben sich einige sicherheitsrelevante Eigenschaften von Hash–Funktionen:

1. Urbildresistenz (oder Einwegeigenschaft).
 2. Schwache Kollisionsresistenz (oder zweite Urbildresistenz).
 3. Starke Kollisionsresistenz (oder Kollisionsresistenz).
1. **Urbildresistenz** bedeutet, dass eine Hash–Funktion eine Einwegfunktion ist, dh. dass es keinen (deterministischen oder probabilistischen) Algorithmus gibt, der in polynomialem Zeit (in der Anzahl N der Stellen des Arguments) mit nicht vernachlässigbarer Wahrscheinlichkeit zu einem gegebenen $h \in \mathbb{F}_2^n$ ein $m \in \mathbb{F}_2^N$ liefert mit $h(m) = h$.

Diese Eigenschaft ist wichtig für kryptographische Zwecke, damit eine Nachricht nicht aus Ihrer Signatur ermittelt werden kann:

Alice bestimmt $h = h(m)$ aus der Nachricht und übermittelt $(e_k(m), \text{sig}_A(h))$ an Bob, wobei sie $\text{sig}_A(h)$ mit einem digitalen Signaturverfahren (etwa DSA, das in Abschnitt 9.6 beschrieben wird) aus h berechnet hat.

Catherine kann über den öffentlichen Schlüssel von Alice die Signatur $\text{sig}_A(h)$ verifizieren und dadurch h bestimmen. Ist daher die Hash–Funktion nicht urbildresistent, so könnte Catherine aus h zumindest ein \tilde{m} bestimmen mit $h(\tilde{m}) = h$, und möglicherweise mit einigen Versuchen sogar m direkt.

2. **Schwache Kollisionsresistenz** bedeutet, dass es keinen Algorithmus gibt, der in polynomialem Laufzeit mit nicht vernachlässigbarer Wahrscheinlichkeit zu einer gegebenen Nachricht $m_1 \in \mathbb{F}_2^\bullet$ ein $m_2 \in \mathbb{F}_2^\bullet$ bestimmt mit $h(m_1) = h(m_2)$.

Das ist wichtig für die Signatur um die Integrität der gesendeten Nachricht sicherzustellen:

Alice bestimmt $h = h(m)$ aus der Nachricht und übermittelt $(e_k(m), \text{sig}_A(h))$ an Bob. Ist die Hash–Funktion, die sie verwendet hat, nicht schwach kollisionsresistent, so kann Alice ein m_2 finden mit $h(m_2) = h$ und im Nachhinein behaupten, m_2 signiert zu haben.

Ferner ist das Verfahren in diesem Fall anfällig für Substitutionsangriffe, falls in der Signatur der Hashwert der verschlüsselten Nachricht $e_1 = e_k(m)$ verwendet wird, also falls $h = h(e_1)$. Catherine könnte nämlich in diesem Fall $(e_1, \text{sig}_A(h))$ durch $(e_2, \text{sig}_A(h))$ für ein e_2 mit $h(e_2) = h$ ersetzen und Bob eine falsche aber korrekt signierte Nachricht unterschieben.

3. **Starke Kollisionsresistenz** bedeutet, dass es keinen Algorithmus gibt, der in polynomialer Laufzeit mit nicht vernachlässigbarer Wahrscheinlichkeit zwei Nachrichten $m_1, m_2 \in F_2^\bullet$ findet, für die $h(m_1) = h(m_2)$ gilt. Das ist eine sehr viel stärkere Anforderung als die schwache Kollisionsresistenz, denn in diesem Fall sind m_1 und m_2 beide frei wählbar).

Diese Anforderung ist relevant für die Kryptographie, um gewisse Arten von Substitutionsangriffen zu unterbinden:

Kann Catherine etwa Alice veranlassen, die Nachricht m_1 zu signieren, so kann sie Bob die Nachricht m_2 mit der (korrekten) Signatur von Alice unterschieben.

Auch Alice kann in diesem Fall hingehen und m_1 signieren (bzw. von Bob signieren lassen) und dann später behaupten, m_2 signiert zu haben (bzw. dass Bob den Vertrag m_2 unterschrieben habe).

Beachten Sie, dass die starke Kollisionsresistenz eine sehr viel stärkere Anforderung an eine Hash–Funktion ist als die schwache Kollisionsresistenz. Das lässt sich sehr einfach am sogenannten „Geburtstagsparadoxon“ veranschaulichen.

Beispiel 9.4. Befinden sich in einem Raum 23 zufällig ausgewählte Personen, so haben mit Wahrscheinlichkeit $p = 0.50$ mindestens zwei davon am gleichen Tag Geburtstag, bei 40 Personen steigt diese Wahrscheinlichkeit schon auf $p = 0.90$ (unter der Annahme, dass die Geburtstage gleichmäßig übers Jahr verteilt sind).

Wird dagegen eine Person ausgewählt, so sind mindestens noch 253 weitere Personen notwendig, um eine Wahrscheinlichkeit von mindestens 0.50 dafür zu bekommen, dass noch eine weitere Person an genau diesem Tag Geburtstag hat.

Übertragen auf Hash–Funktionen besagt das Geburtstagsparadoxon, dass sich bei einer Hash–Funktion

$$h : \mathbb{F}_2^\bullet \longrightarrow F_2^n$$

unter $t = \sqrt{2^n} = 2^{\frac{n}{2}}$ zufällig ausgewählten Nachrichten $m_1, \dots, m_t \in \mathbb{F}_2^\bullet$ mit einer Wahrscheinlichkeit von mindestens $p = 0.50$ zwei Nachrichten m_{τ_1}, m_{τ_2} mit

$$h(m_{\tau_1}) = h(m_{\tau_2})$$

befinden, dh. für kleine n können Kollisionen mit hoher Wahrscheinlichkeit mit einem brute-force-Angriff gefunden werden. Um dagegen zu einem gegebenen m_1 mit einer Wahrscheinlichkeit von mindestens $p = 0.50$ eine weitere Nachricht m_2 zu finden mit $h(m_2) = h(m_1)$ sind 2^{n-1} zufällig ausgewählte Nachrichten nötig.

Regel 9.1. Eine kollisionsresistente Hash–Funktion ist schwach kollisionsresistent, eine schwach–kollisionsresistente Hash–Funktion ist urbildresistant.

Beweis: Ist eine Hash–Funktion nicht schwach kollisionsresistent, so kann zu gegebenem $m_1 \in \mathbb{F}_2^\bullet$ in polynomialer Zeit mit nicht vernachlässigbarer Wahrscheinlichkeit ein m_2 gefunden werden mit

$$h(m_1) = h(m_2)$$

also ist h nicht kollisionsresistent.

Ist h nicht urbildresistant, so kann zu einem $m_1 \in F_2^\bullet$ mit $h(m_1) = h$ in polynomialer Zeit mit nicht vernachlässigbarer Wahrscheinlichkeit ein $\tilde{m} \in \mathbb{F}_2^\bullet$ gefunden werden mit $h(\tilde{m}) = h$. Da $h : \mathbb{F}_2^\bullet \rightarrow \mathbb{F}_2^n$ in der Regel zu jedem $h \in \mathbb{F}_2^n$ sehr viele Urbilder hat, ist mit hoher Wahrscheinlichkeit $\tilde{m} \neq m_1$. Also ist h nicht schwach kollisionsresistent.

Definition 9.2. Eine **kryptographische Hash–Funktion**

$$h : \mathbb{F}_2^\bullet \rightarrow \mathbb{F}_2^n$$

ist eine kollisionresistente Hash–Funktion.

Bemerkung 9.1. Aufgrund des Geburtstagsparadoxons muss n so groß sein, dass die Hashwerte von $2^{\frac{n}{2}}$ Nachrichten $m \in \mathbb{F}_2^\bullet$ nicht mehr in einer überschaubaren Zeit berechnet werden können. Andernfalls wäre die Hash–Funktion schon aufgrund eines brute–force–Angriffs nicht mehr kollisionsresistent.

Bemerkung 9.2. Außer bei digitalen Signaturen sind Hash–Funktionen auch für die Passwortkontrolle wesentlich: Gespeichert wird nämlich üblicherweise nicht das Passwort sondern der Hashwert des Passworts (bzw. eines Paddings davon). Zugang wird gewährt, wenn der Hashwert der Eingabe dem gespeicherten Wert entspricht.

Wird die Passwortdatei gestohlen, so kann sie aufgrund der (schwachen) Kollisionsresistenz nicht benutzt werden, um Zugang zum System zu bekommen.

9.3. Die Merkle–Damgård–Metakonstruktion

Die Merkle–Damgård–Metakonstruktion erzeugt aus Nachrichten von der Länge bis zu $2^{64} - 1$ Bits einen Hashwert einer vorgegebenen Länge n . Ausgangspunkt ist eine kollisionsresistente Abbildung

$$f : \mathbb{F}_2^{n+r} \rightarrow \mathbb{F}_2^n$$

Ein solches f heißt **Kompressionsfunktion** und das r heißt die **Kompressionsrate** von f . Dabei sollte r hinreichend groß sein (mindestens $r = 64$).

Beispiel 9.5. Wir betrachten \mathbb{F}_2^8 als Körper mit 256 Elementen (definiert durch die Relation $\alpha^8 = \alpha^4 + \alpha^3 + \alpha + 1$). Die Funktion

$$f : \mathbb{F}_2^{8+24} \longrightarrow \mathbb{F}_2^8$$

mit

$$f(A\|B\|C\|D) = A + \alpha \cdot B + \alpha^2 \cdot C + \alpha^3 \cdot D$$

(mit $A, B, C, D \in \mathbb{F}_2^8 = \mathbb{F}_{256}$) ist eine Kompressionsfunktion mit Kompressionsrate 24.

Hierfür gilt

$$f(0x\ 7ab3c02f) = 0x\ 01001001 = 0x\ 49$$

Die Merkle–Damgård–Metakonstruktion lässt sich wie folgt beschreiben:

Vorbereitung:

Fixiere einen Startwert $v \in \mathbb{F}_2^n$ (einmalig für die gesamte Metakonstruktion).

Padding:

Teile die gesamte Nachricht m in Blöcke der Länge r auf. Dabei erfolgt ein Padding der Nachricht für den Fall, dass l , die Länge von m , nicht durch r teilbar ist, auf eine der folgenden beiden Methoden

1. Methode 1 (verwendet etwa bei der MD–Familie):
 - a) Setze $t = l \bmod r$ (sodass also $1 \leq t \leq r - 1$).
 - b) Ergänze den letzten Block mit $10 \dots 0$ (mit $r - t - 1$ –mal der 0).
 - c) Schreibe l als Binärzahl b und ergänze einen weiteren Block der Länge r , der b (rechtsbündig geschrieben mit führenden Nullen) enthält.
2. Methode 2 (verwendet etwa bei der SHA–Familie, also bei SHA-1 oder SHA-2):
 - a) Setze $t = l \bmod r$ (sodass also $1 \leq t \leq r - 1$).
 - b) Falls $r - t < 65$, so fahre fort wie in Methode 1.
 - c) Falls $r - t \geq 65$, so schreibe l als Binärzahl b (rechtsbündig in eine 64–Bitblock) und ergänze den letzten Block durch $10 \dots 0b$ (mit $r - t - 65$ Nullen zwischen der 1 und b).

Als Ergebnis des Paddings erhalten wir in jedem Fall eine Erweiterung \tilde{m} von m von der Form

$$\tilde{m} = m_1 \| m_2 \| \dots \| m_k \quad \text{mit } m_i \in F_2^r$$

Berechnung des Hashwerts:

Der Hashwert berechnet sich nun wie folgt:

1. Setze $v_0 = v$ (aus dem allgemeinen Vorbereitungsschritt).
2. Für $i = 1, \dots, k$ setze

$$v_i = f(v_{i-1} \| m_i)$$

3. Definiere $h(m) = v_k$.

Beispiel 9.6. Wir betrachten \mathbb{F}_2^8 als Körper mit 256 Elementen (definiert durch die Relation $\alpha^8 = \alpha^4 + \alpha^3 + \alpha + 1$). Für die Kompressionsfunktion

$$f : \mathbb{F}_2^{8+24} \longrightarrow \mathbb{F}_2^8$$

mit

$$f(A \| B \| C \| D) = A + \alpha \cdot B + \alpha^2 \cdot C + \alpha^3 \cdot D$$

(mit $A, B, C, D \in \mathbb{F}_2^8 = \mathbb{F}_{256}$) aus Beispiel 9.5 und

$$v_0 = 0x 7a = 0x\ 01111010$$

berechnet sich der Hashwert der Nachricht

$$m = 0x b3c02fb1952d$$

wie folgt:

Ein Padding ist nicht notwendig.

Die Zerlegung von m ist

$$m = m_1 \| m_2$$

mit

$$m_1 = 0x b3c02f, \quad m_2 = 0x b1952d$$

Damit gilt

$$v_1 = f(v_0 \| m_1) = f(0x 7ab3c02f) = 0x 01001001 = 0x 49$$

und

$$v_2 = f(v_1 \| m_2) = f(0x 49b1952d) = 0x 00100001 = 0x 21$$

Damit gilt also

$$h = 0x 00100001 = 0x 21$$

9.4. Secure Hash Algorithm 1 (SHA-1)

Der **Secure Hash Algorithm 1** SHA-1 ist eine von der NSA entwickelte kryptographische Hash–Funktion, die immer noch in vielen Sicherheitsanwendungen und Protokollen verwendet wird (TLS, SSL, SSH, IPsec, …), die aber vom NIST seit 2011 nicht mehr empfohlen wird. Eine erste Kollision, bestehend aus zwei unterschiedlichen aber validen PDF–Dokumenten, die den gleichen Hashwert liefern, wurde 2017 gefunden. Daher sollte SHA-1 auf jeden Fall nicht mehr in sicherheitskritischen Bereichen verwendet werden. Seine Weiterentwicklung SHA-2, die aktuell noch als sicher gilt, beruht auf ähnlichen Techniken und Konstruktionen, ist aber deutlich komplexer. Daher beschränken wir uns hier auf SHA-1, um die grundlegenden Prinzipien zu erklären.

Zur Erklärung von SHA-1 werden neben dem binären $+$ (XOR) noch die folgenden binären Operationen benötigt:

Die AND–Operation \wedge :

\wedge	0	1
0	0	0
1	0	1

Die OR–Operation \vee :

\vee	0	1
0	0	1
1	1	1

Die NOT–Operation \neg die gegeben ist durch

$$\neg 0 = 1, \quad \neg 1 = 0$$

Bemerkung 9.3. Für $a, b \in \mathbb{F}_2$ gilt

$$a \wedge b = a \cdot b, \quad \neg a = 1 - a$$

und

$$a \vee b = 1 - \neg a \cdot \neg b = 1 - (1 - a) \cdot (1 - b)$$

Bemerkung 9.4. Die Operationen \wedge , \vee und \neg werden auch komponentenweise auf binäre n –Tupel angewendet.

Beispiel 9.7.

- $(010111) \wedge (111000) = (010000)$.

- $(010111) \vee (111000) = (111111)$.
- $\neg(010111) = (101000)$.

Definition 9.3. Für ein binäres n -Tupel $b = (b_0, b_1, \dots, b_{n-1})$ ist

$$\text{lrot}(b, t) = (b_t, b_{t+1}, \dots, b_{n-1}, b_0, \dots, b_{t-1})$$

die **Linksrotation** von b um t Stellen.

Der Secure-Hash-Algorithm SHA-1 liefert eine Hash-Funktion

$$\text{SHA-1} : \mathbb{F}_2^\bullet \longrightarrow \mathbb{F}_2^{160}$$

wobei $\bullet \leq 2^{64} - 1$, und beruht auf der Merkle–Damgård Metakonstruktion. Die Kompressionsfunktion

$$f : \mathbb{F}_2^{160+512} \longrightarrow \mathbb{F}_2^{160}$$

hat eine Kompressionsrate von 512 und funktioniert wie folgt:

1. Vorbereitung:

Setze

$$v_0 = H_0^{(0)} \| H_0^{(1)} \| H_0^{(2)} \| H_0^{(3)} \| H_0^{(4)}$$

wobei die $H_0^{(j)} \in \mathbb{F}_2^{32}$ wie folgt (hexadezimal) gegeben sind:

$$\begin{aligned} H_0^{(0)} &= 0x 67452301 \\ H_0^{(1)} &= 0x efcdab89 \\ H_0^{(2)} &= 0x 98badcfe \\ H_0^{(3)} &= 0x 10325476 \\ H_0^{(4)} &= 0x c3d2e1f0 \end{aligned}$$

2. Aufbereitung der Nachricht:

Nach Padding und Aufteilung der Nachricht

$$\tilde{m} = m_1 \| m_2 \| \dots \| m_k$$

mit $m_i \in \mathbb{F}_2^{512}$ wird der i -te Block geschrieben als

$$m_i = m_i^{(0)} \| m_i^{(1)} \| \dots \| m_i^{(15)}$$

mit $m_i^{(j)} \in \mathbb{F}_2^{32}$.

Die Merkle–Damgård–Metakonstruktion liefert aus Schritt $i - 1$ ein Element

$$v_{i-1} = H_{i-1}^{(0)} \| H_{i-1}^{(1)} \| H_{i-1}^{(2)} \| H_{i-1}^{(3)} \| H_{i-1}^{(4)}$$

mit $H_{i-1}^{(j)} \in \mathbb{F}_2^{32}$. Dann setzen wir für Schritt i :

$$\begin{array}{lll} A & = & H_{i-1}^{(0)} \\ B & = & H_{i-1}^{(1)} \\ C & = & H_{i-1}^{(2)} \\ D & = & H_{i-1}^{(3)} \\ E & = & H_{i-1}^{(4)} \end{array} \quad \begin{array}{lll} H_0 & = & H_{i-1}^{(0)} \\ H_0 & = & H_{i-1}^{(1)} \\ H_0 & = & H_{i-1}^{(2)} \\ H_0 & = & H_{i-1}^{(3)} \\ H_0 & = & H_{i-1}^{(4)} \end{array}$$

3. Verarbeitung in Schritt i :

Im Schritt i wird der i -te Block m_i wie folgt in 80 Runden verarbeitet:

1. Schreibe

$$m_i = m_i^{(0)} \| m_i^{(1)} \| \dots \| m_i^{(15)}$$

mit $m_i^{(j)} \in \mathbb{F}_2^{32}$.

2. Für $j = 0, \dots, 15$ setze

$$W[j] = m_i^{(j)}$$

und für $j = 16, \dots, 79$ setze

$$\begin{aligned} \text{TEMP} &= W[j-3] + W[j-8] + W[j-14] + W[j-16] \\ W[j] &= \text{lrot}(\text{TEMP}, 1) \end{aligned}$$

3. Für $j = 0, \dots, 19$ setze

$$\begin{aligned} f[j](B, C, D) &= (B \wedge C) \vee ((\neg B) \wedge D) \\ K[j] &= 0x 5a827999 \end{aligned}$$

für $j = 20, \dots, 39$ setze

$$\begin{aligned} f[j](B, C, D) &= B + C + D \\ K[j] &= 0x 6ed9eby1 \end{aligned}$$

für $j = 40, \dots, 59$ setze

$$\begin{aligned} f[j](B, C, D) &= (B \wedge C) \vee (B \wedge D) \vee (C \wedge D) \\ K[j] &= 0x 8f1bbcdcc \end{aligned}$$

und für $j = 60, \dots, 79$ setze

$$\begin{aligned} f[j](B, C, D) &= B + C + D \\ K[j] &= 0x ca62c1d6 \end{aligned}$$

4. Für $j = 0, \dots, 79$ definiere

$$\begin{aligned}\text{TEMP} &= \text{lrot}(A, 5) + f[j](B, C, D) + E + K[j] + W[j] \\ E &= D \\ D &= C \\ C &= \text{lrot}(B, 30) \\ B &= A \\ A &= \text{TEMP}\end{aligned}$$

5. Setze

$$\begin{aligned}H_i^{(0)} &= H_{i-1}^{(0)} + A \\ H_i^{(1)} &= H_{i-1}^{(1)} + B \\ H_i^{(2)} &= H_{i-1}^{(2)} + C \\ H_i^{(3)} &= H_{i-1}^{(3)} + D \\ H_i^{(4)} &= H_{i-1}^{(4)} + E\end{aligned}$$

und

$$v_i = H_i^{(0)} \| H_i^{(1)} \| H_i^{(2)} \| H_i^{(3)} \| H_i^{(4)}$$

Beispiel 9.8. Wir betrachten die Nachricht

$$m = \text{Das Pferd frisst keinen Gurkensalat}$$

(gespeichert im ASCII-Code). Dann gilt (hexadezimal geschrieben)

$$\text{SHA-1}(m) = 0x 286674e6501808070b91fa35caecb77d45cd7e3a$$

Betrachten wir dagegen

$$\tilde{m} = \text{Das Pferd frisst feinen Gurkensalat}$$

so gilt (hexadezimal geschrieben)

$$\text{SHA-1}(\tilde{m}) = 0x 9ff28d104ecfefddff97647ed3a649d11f556869$$

Die Nachrichten haben sich also nur an einer Stelle geändert, die Hashwerte dagegen an fast allen Stellen.

9.5. Die Schwamm-Konstruktion

Eine weitere Möglichkeit, Hash-Funktionen zu konstruieren, ist die sogenannte **Schwamm-Methode**, auf der etwa der Secure-Hash-Algorithm 3 SHA-3 beruht. SHA-3 ist eine Alternative zu SHA-2 und gilt aktuell als sicher.

Die Schwammkonstruktion besteht aus drei Komponenten:

1. Einem internen Zustandsvektor $S = \mathbb{F}_2^b$.
2. Einer Transformationsfunktion $f : \mathbb{F}_2^b \rightarrow \mathbb{F}_2^b$, wobei b aufgeteilt wird als $b = r + c$ mit einer **Bitrate** r und einer **Kapazität** c . Entsprechend wird f auch geschrieben als
$$f : \mathbb{F}_2^r \times \mathbb{F}_2^c \rightarrow \mathbb{F}_2^r \times \mathbb{F}_2^c$$
und ein Zustand S als $S = R \| C$ mit $R \in \mathbb{F}_2^r$ und $C \in \mathbb{F}_2^c$.
3. Einem Padding–Prozess P , der eine Nachricht m so ergänzt, dass ihre Bitlänge ein Vielfaches von r ist.

Beispiel 9.9. Wir betrachten \mathbb{F}_2^8 als Körper mit 256 Elementen (definiert durch die Relation $\alpha^8 = \alpha^4 + \alpha^3 + \alpha + 1$). Die Funktion

$$f : \mathbb{F}_2^{8+16} \rightarrow \mathbb{F}_2^{8+16}$$

mit

$$f(A \| B \| C) = (A + B + C) \| (A + \alpha \cdot B + \alpha^2 \cdot C) \| (\alpha^5 \cdot A + \alpha^4 \cdot B + \alpha^3 \cdot C)$$

(mit $A, B, C \in \mathbb{F}_2^8 = \mathbb{F}_{256}$) ist eine Transformationsfunktion mit Bitrate 8 und Kapazität 16.

Hierfür gilt

$$f(0x\ 7ab32f) = 0x\ 11100110 \| 10111011 \| 01111111 = 0x\ d6 \| bb \| 7f$$

Mit diesen Daten kann eine Hash–Funktion

$$h : \mathbb{F}_2^\bullet \rightarrow \mathbb{F}_2^n, \quad m \mapsto h(m)$$

beliebiger Länge n wie folgt konstruiert werden:

1. Initialisiere den internen Zustandsvektor

$$S = (0, 0, \dots, 0) \in \mathbb{F}_2^b$$

2. Benutze den Paddingprozess P um aus der Nachricht m einen Bitstring \tilde{m} zu erzeugen, dessen Länge ein Vielfaches von r ist und schreibe

$$\tilde{m} = m_1 \| m_2 \| \dots \| m_k$$

wobei $k = \lfloor \frac{\text{length}(m)}{r} \rfloor + 1$ oder $k = \lfloor \frac{\text{length}(m)}{r} \rfloor + 2$, je nach Art des Paddings und des Paddingprozesses.

Das $10 * 01$ -Padding, das bei SHA-3 verwendet wird, erfolgt einheitlich durch eine 1, gefolgt von hinreichend vielen 0-Stellen und am Schluss wieder einer 1. Dabei müssen die beiden 1-Stellen immer vorkommen, dh. wenn die Länge L von m bereits durch r teilbar ist, so wird ein gesamter Block der Länge r , bestehend aus einer 1, $r - 2$ mal der 0 und dann noch einer 1 angehängt, und wenn $L + 1$ durch r teilbar ist, dann wird ein Block der Länge $r + 1$ bestehend aus einer 1, $r - 1$ mal der 0 und dann noch einer 1 angehängt. Ist $L + 2$ durch r teilbar, so erfolgt das padding durch 11, dh. in diesem Fall treten im padding keine 0-Stellen auf.

3. Aufsaug–Phase oder absorbing phase:

Für $i = 1, \dots, k$

schreibe $S = R \| C$.

setze $S = f(R + m_i \| C)$.

4. Auspress–Phase oder squeezing phase:

Setze $l = \lceil \frac{n}{r} \rceil$ (wobei n die gewünschte output-Länge bezeichnet).

schreibe $S = R \| C$.

setze $H_1 = R$.

Für $i = 2, \dots, l$

setze $S = f(S)$.

schreibe $S = R \| C$.

setze $H_i = R$.

Schließlich wird H_l so abgeschnitten, dass H_l die Bitlänge $n - (l - 1) \cdot r$ hat.

5. Setze

$$h(m) = H_1 \| H_2 \| \dots \| H_l$$

Beispiel 9.10. Wir betrachten \mathbb{F}_2^8 als Körper mit 256 Elementen (definiert durch die Relation $\alpha^8 = \alpha^4 + \alpha^3 + \alpha + 1$) und die Transformationsfunktion

$$f : \mathbb{F}_2^{8+16} \longrightarrow \mathbb{F}_2^{8+16}$$

mit

$$f(A \| B \| C) = (A + B + C) \| (A + \alpha \cdot B + \alpha^2 \cdot C) \| (\alpha^5 \cdot A + \alpha^4 \cdot B + \alpha^3 \cdot C)$$

(mit $A, B, C \in \mathbb{F}_2^8 = \mathbb{F}_{256}$) aus Beispiel 9.9.

Mithilfe dieser Transformation soll nun der Hashwert der Nachricht

$$m = 0x\ 7ab3c0b1052d$$

der Länge 16 bestimmt werden. Ein Padding ist nicht erforderlich, und wir starten mit

$$m_1 = 0x\ 7a, \quad m_2 = 0x\ b3, \quad m_3 = 0x\ c0, \quad m_4 = 0x\ b1, \quad m_5 = 0x\ 05, \quad m_6 = 0x\ 2d$$

Zur Initialisierung setzen wir

$$S_0 = (0, 0, \dots, 0) \in \mathbb{F}_2^{24}$$

und schreiben

$$S_0 = S_{0,1} \| S_{0,2} \| S_{0,3}$$

mit $S_{0,i} \in \mathbb{F}_2^8$.

Die Aufsaugphase läuft nun ab wie folgt:

$i = 1$:

$$S_1 = S_{1,1} \| S_{1,2} \| S_{1,3} = f(S_{0,1} + m_1 \| S_{0,2} \| S_{0,3}) = 0x\ 7a \| 7a \| d9$$

$i = 2$:

$$S_2 = S_{2,1} \| S_{2,2} \| S_{2,3} = f(S_{1,1} + m_2 \| S_{1,2} \| S_{1,3}) = 0x\ 6a \| 74 \| 3b$$

$i = 3$:

$$S_3 = S_{3,1} \| S_{3,2} \| S_{3,3} = f(S_{2,1} + m_3 \| S_{2,2} \| S_{2,3}) = 0x\ f5 \| be \| 68$$

$i = 4$:

$$S_4 = S_{4,1} \| S_{4,2} \| S_{4,3} = f(S_{3,1} + m_4 \| S_{3,2} \| S_{3,3}) = 0x\ 92 \| 98 \| 20$$

$i = 5$:

$$S_5 = S_{5,1} \| S_{5,2} \| S_{5,3} = f(S_{4,1} + m_5 \| S_{4,2} \| S_{4,3}) = 0x\ bf \| ac \| b8$$

$i = 6$:

$$S_6 = S_{6,1} \| S_{6,2} \| S_{6,3} = f(S_{5,1} + m_6 \| S_{5,2} \| S_{5,3}) = 0x\ 86 \| 07 \| 44$$

Nach der Aufsaugphase hat der Zustandsvektor also den Wert

$$S = 0x\ 86 \| 07 \| 44$$

Für die Auspressphase berechnen wir

$$l = \lceil \frac{16}{8} \rceil = \frac{16}{8} = 2$$

Die Auspressphase läuft nun ab wie folgt:

$i = 1$:

$$S = 0x\ 86\|07\|44 = R\|C$$

wobei

$$H_1 = R = 0x\ 86$$

$i = 2$:

$$S = f(S) = 0x\ c5\|83\|0d = R\|C$$

wobei $R = 0x\ c5$. Ein Abschneiden ist hier nicht notwendig, und wir erhalten

$$H_2 = R = 0x\ c5$$

Als Hashwert ermitteln wir

$$h = H_1\|H_2 = 0x\ 86\|c5$$

Das SHA-3–Verfahren benutzt die Schwammmethode mit der Transformationsfunktion

$$f = \text{Keccak–f}[1600] : \mathbb{F}_2^b \longrightarrow \mathbb{F}_2^b$$

wobei $b = 5 \cdot 5 \cdot 2^6 = 1600$ (Breite der Transformation).

Der Zustandsvektor S wird dabei als dreidimensionaler Array der Dimension $5 \times 5 \times 64$ geschrieben, und zwar wie folgt:

ist $S = (S[0], S[1], \dots, S[1599])$, so setze

$$S[m, n, o] = S[(m + 5n) \cdot 64 + o] \quad (0 \leq m, n \leq 4, 0 \leq o \leq 63)$$

Die **Bahn** eines Zustands S zum Index (m, n) ist der Vektor

$$(S[m, n, 0], S[m, n, 1], \dots, S[m, n, 63]) \in \mathbb{F}_2^{64}$$

wofür wir kurz $S[m, n]$ schreiben.

Die **Scheibe** des Zustands S zum Index o ist die 5×5 –Matrix

$$\begin{pmatrix} S[0, 0, o] & S[0, 1, o] & \dots & S[0, 4, o] \\ S[1, 0, o] & S[1, 1, o] & \dots & S[1, 4, o] \\ \vdots & & & \vdots \\ S[4, 0, o] & S[4, 1, o] & \dots & S[4, 4, o] \end{pmatrix}$$

Eine **Zeile** des Zustands S zum Index n und o ist der Vektor

$$(S[0, n, o], S[1, n, o], S[2, n, o], S[3, n, o], S[4, n, o]) \in \mathbb{F}_2^5$$

und eine **Spalte** des Zustands S zum Index m und o ist der Vektor

$$(S[m, 0, o], S[m, 1, o], S[m, 2, o], S[m, 3, o], S[m, 4, o]) \in \mathbb{F}_2^5$$

Die Transformation Keccak-f[1600] besteht aus 24-Runden $i = 0, \dots, 23$, in denen jeweils die Transformation $g_i : \mathbb{F}_2^b \rightarrow \mathbb{F}_2^b$ ausgeführt wird, wobei g_i wiederum aufgebaut ist aus 5 Operationen,

$$g_i = \iota_i \circ \chi \circ \pi \circ \rho \circ \theta$$

(wobei nur ι_i von der aktuellen Runde i abhängt).

Definition 9.4. Für ein l -Tupel $b = (b_0, b_1, \dots, b_{l-1})$ bezeichnen wir mit

$$\text{rrot}(b, t) = (b_{l-t}, b_{l-t+1}, \dots, b_{l-1}, b_0, \dots, b_{l-t-1})$$

die **Rechtsrotation** von b um t Stellen.

Die Teilschritte von g_i in Runde i ($i = 0, \dots, 23$) sind nun wie folgt aufgebaut:

1. Die Operation θ :

Für $m = 0, \dots, 4$ setze

$$\begin{aligned} C[m] &= S[m, 0] + S[m, 1] + S[m, 2] + S[m, 3] + S[m, 4] \in \mathbb{F}_2^{64} \\ D[m] &= C[m - 1] + \text{rrot}(C[m + 1], 1) \end{aligned} \in \mathbb{F}_2^{64}$$

(wobei alle Indizes mod 5 zu lesen sind) und für $n = 0, \dots, 4$ setze

$$S[m, n] = S[m, n] + D[m]$$

Dieser Schritt kann elementweise so beschrieben werden:

$$S[m, n, o] = \S[m, n, o] + \sum_{\nu=0}^4 S[m - 1, \nu, o] + \sum_{\nu=0}^4 S[m + 1, \nu, o - 1]$$

2. Die Operation ρ :

Für $(m, n) \neq (0, 0)$ bestimme $\tau = \tau(m, n)$ so, dass

$$\binom{m}{n} = \begin{pmatrix} 3 & 2 \\ 1 & 0 \end{pmatrix}^\tau \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \mod 5$$

und setze

$$t(m, n) = \frac{(\tau + 1) \cdot (\tau + 2)}{2}$$

(was immer ganzzahlig ist). Ferner setze

$$t(0, 0) = 0$$

Für $m = 0, \dots, 4$, $n = 0, \dots, 4$ setze

$$S[m, n] = \text{rrot}(S[m, n], t(m, n))$$

oder elementweise

$$S[m, n, o] = S[m, n, o - t(m, n)]$$

für $m = 0, \dots, 4$, $n = 0, \dots, 4$ und $o = 0, \dots, 63$.

3. Die Operation π :

Für $m = 0, \dots, 4$, $n = 0, \dots, 4$ setze

$$S[m, n] = S[m + 3n, m]$$

oder elementweise

$$S[m, n, o] = S[m + 3n, m, o]$$

für $m = 0, \dots, 4$, $n = 0, \dots, 4$ und $o = 0, \dots, 63$.

4. Die Operation χ :

Für $m = 0, \dots, 4$, $n = 0, \dots, 4$ setze

$$S[m, n] = S[m, n] + (\neg S[m + 1, n] \wedge S[m + 2, n])$$

oder elementweise

$$S[m, n, o] = S[m, n, o] + (1 + S[m + 1, n, o]) \cdot S[m + 2, n, o]$$

für $m = 0, \dots, 4$, $n = 0, \dots, 4$ und $o = 0, \dots, 63$.

5. Die Opteration ι_i :

$$S[0, 0] = S[0, 0] + RC[i]$$

mit einer Rundenkonstante $RC[i]$.

Damit ist Runde i abgeschlossen. Die Indizes sind dabei immer mod 5 (für m und n) bzw. mod 64 (for o) zu lesen.

Bemerkung 9.5. Die Rundenkonstanten $RC[i]$ sind durch folgende Tabelle gegeben

i	$RC[i]$	i	$RC[i]$
0	0x 00000000000000000001	12	0x 000000008000808b
1	0x 0000000000008082	13	0x 80000000000000008b
2	0x 800000000000808a	14	0x 8000000000008089
3	0x 8000000080008000	15	0x 8000000000008003
4	0x 000000000000808b	16	0x 8000000000008002
5	0x 0000000080000001	17	0x 8000000000000080
6	0x 8000000080008081	18	0x 000000000000800a
7	0x 8000000000008009	19	0x 800000008000000a
8	0x 000000000000008a	20	0x 8000000080008081
9	0x 0000000000000088	21	0x 8000000000008080
10	0x 0000000080008009	22	0x 0000000080000001
11	0x 000000008000000a	23	0x 8000000080008008

Bemerkung 9.6. Standardinstanzen von SHA-3 nach NIST sind

Standard	Hashlänge	Bitrate r	Kapazität c
SHA-3-224	224	1152	448
SHA-3-256	256	1088	512
SHA-3-384	384	832	768
SHA-3-512	512	576	1024

Bemerkung 9.7. Die Verschiebungsparameter $t(m, n)$ sind durch die folgende Tabelle gegeben:

$n \backslash m$	0	1	2	3	4
0	0	1	62	28	27
1	36	44	6	55	20
2	3	10	43	25	39
3	41	45	15	21	8
4	18	2	61	56	8

Beispiel 9.11. Wir betrachten die Nachricht

$$m = \text{Das Pferd frisst keinen Gurkensalat}$$

(gespeichert im ASCII–Code). Dann gilt (hexadezimal geschrieben)

$$\text{SHA–3}(m) = 0x\ 97ca2b2e9d417f85669e82c026f2a51d2f81bafe0ada55beacf3a1c8$$

Betrachten wir dagegen

$$\widetilde{m} = \text{Das Pferd frisst feinen Gurkensalat}$$

so gilt (hexadezimal geschrieben)

$$\text{SHA–3}(\widetilde{m}) = 0x\ cb31c25920c22620e638c27238ccaaa9533cb0cf19ebdf77642b6c91$$

(wobei wir jeweils den SHA-3-224 –Standard benutzt haben). Auch hier erhalten wir also wieder sehr starke Unterschiede in den Hashwerten.

9.6. Der digitale Signaturalgorithmus DSA

Eine der Hauptanwendungen von Hash–Funktionen findet sich bei digitalen Signaturverfahren. Um eine gesamte, komplexe Nachricht mit einer Signatur zu versehen und zu unterzeichnen, wird zunächst aus der Nachricht ein Hashwert erzeugt, der eine Länge hat, die in einem Schritt signiert werden kann, und dann wird dieser Hashwert signiert. Aufgrund der Kollisionsresistenz ist die Signatur des Hashwertes als gleichwertig mit der Signatur der gesamten Nachricht zu betrachten.

Von NIST zur Verwendung im Digital Signature Standard DSS vorgeschlagen ist der Digital Signature Algorithm DSA, der die ElGamal–Signatur weiterentwickelt und auf dem (klassischen) Diffie–Hellman–Protokoll beruht.

Hierfür sind verschiedene Standards eingeführt worden. Wir betrachten den Standard (2048, 256), der einen Hashwert der Länge 256 Bit signiert. Dieses Verfahren gilt als voraussichtlich sicher bis 2030. Ab dann sollte der Standard (3072, 256) benutzt werden. Die verwendete Hash–Funktion muss mindestens die Ausgangslänge 256 Bit haben. Sind die Hashwerte länger, so werden sie nach 256 Stellen abgeschnitten. Dieser (gegebenenfalls trunkierte) Hashwert der Nachricht m wird im folgenden mit $\text{SHA}(m)$ bezeichnet.

Ein vergleichbares Signaturschema ECDSA, dass auf elliptischen Kurven beruht, wird im Abschnitt 14.4 vorgestellt.

Vorbereitung durch Alice:

Alice wählt für sich die folgenden Daten:

1. Eine Primzahl p mit $2^{2047} < p < 2^{2048}$ (also eine Primzahl der binären Länge 2048)

2. Eine Primzahl q mit $2^{255} < q < 2^{256}$, die $p - 1$ teilt (dh. die binäre Länge von q ist 256).
3. Ein Element $g \in \mathbb{F}_p^*$ der Ordnung q und die von g erzeugte Untergruppe $U = U_g = \langle g \rangle \subseteq \mathbb{F}_p^*$.

Bemerkung 9.8. Ein Element g der gewünschten Form kann Alice wie folgt finden:
Sie wählt zufällig ein $x \in \{2, \dots, p - 2\}$ aus und berechnet

$$g = x^{\frac{p-1}{q}}$$

Falls $g \neq 1$ (was mit hoher Wahrscheinlichkeit der Fall sein wird), so ist g ein Element der gewünschten Form, andernfalls wiederholt sie diesen Schritt mit einem anderen x .

Die Parameter (p, q, g) sind die allgemeinen Parameter des Signaturschemas von Alice und üblicherweise systemweit bekannt.

Schlüsselerzeugung:

Alice erzeugt ein Schlüsselpaar $(k_{\text{pr},A}, k_{\text{pub},A})$ wie folgt:

1. Alice wählt zufällig eine Zahl $u \in \{2, \dots, q - 1\}$.
2. Alice berechnet $v = g^u$ in U_g .
3. Der private Schlüssel von Alice ist $k_{\text{pr},A} = (q, p, g, u)$.
4. Der öffentliche Schlüssel von Alice ist $k_{\text{pub},A} = (q, p, g, v)$.
5. Alice veröffentlicht $k_{\text{pub},A}$.

Bemerkung 9.9. Aufgrund des diskreten Logarithmus-Problems ist es praktisch nicht möglich, aus p, q, g und v auf u zu schließen.

Signatur durch Alice:

Alice will eine Nachricht m bzw. ihren Hashwert $\text{SHA}(m)$, die sie an Bob schickt, signieren, damit Bob sicher sein kann, dass sie von ihr ist. Wir gehen hier davon aus, dass, mit $t = 256 = \lfloor \log_2(q) \rfloor + 1$, der Hashwert $\text{SHA}(m)$ bereits als binäres t -Tupel $m = (b_1, \dots, b_t) \in \mathbb{F}_2^t$ vorliegt. Dann geht Alice vor wie folgt.

1. Alice wählt zufällig ein $z \in \{2, \dots, q - 1\}$.
2. Alice berechnet $n = (g^z \bmod p) \bmod q$.

3. Falls $n = 0$, so geht Alice zurück zu Schritt (1).

4. Alice berechnet $z^{-1} \in \mathbb{F}_q$ und setzt

$$s = z^{-1} \cdot (\text{SHA}(m) + n \cdot u) \mod q$$

(dh. sie betrachtet alle Zahlen als Restklassen in \mathbb{F}_q).

5. Falls $s = 0$, so geht Alice zurück zu Schritt (1).

6. Alice signiert ihre Nachricht mit $\text{sig}_A(m) = (n, s)$ und schickt $c = (m, \text{sig}_A(m))$ an Bob.

Bemerkung 9.10. Die Zahl m kann größer sein als r , sie hat aber die gleiche binäre Größenordnung wie r .

Verifikation durch Bob:

Bob empfängt die Nachricht $c = (m, (n, s))$ und überprüft, ob es sich bei (n, s) um die Signatur $\text{sig}_A(m)$ von Alice handelt wie folgt:

1. Bob überprüft ob $1 \leq n \leq q - 1$ und $1 \leq s \leq q - 1$. Ist das nicht der Fall, so lehnt er die Signatur ab.
2. Bob berechnet $s^{-1} \in \mathbb{F}_q$ und setzt

$$w_1 = \text{SHA}(m) \cdot s^{-1} \mod q, \quad w_2 = n \cdot s^{-1} \mod q$$

und

$$\tilde{n} = (g^{w_1} \cdot v^{w_2} \mod p) \mod q$$

(wobei m die Nachricht ist und v aus $k_{\text{pub}, A}$).

3. Ist $\tilde{n} = 0$, so lehnt Bob die Signatur ab.
4. Bob akzeptiert die Signatur, falls $n = \tilde{n} \mod q$, andernfalls lehnt er sie ab.

Hilfssatz 9.2. Hat Alice die Nachricht korrekt mit $\text{sig}_A(m) = (n, s)$ signiert, so gilt

$$n = \tilde{n} \mod q$$

Beweis: Es ist $s = z^{-1} \cdot (\text{SHA}(m) + n \cdot u)$ (in \mathbb{F}_q). Damit gilt

$$\begin{aligned} z &= s^{-1} \cdot (\text{SHA}(m) + n \cdot u) \\ &= s^{-1} \cdot \text{SHA}(m) + s^{-1} \cdot n \cdot u \\ &= \text{SHA}(m) \cdot s^{-1} + n \cdot s^{-1} \cdot u \\ &= w_1 + w_2 \cdot u \end{aligned}$$

also

$$\begin{aligned} g^z &= g^{w_1} \cdot g^{n \cdot s^{-1} \cdot u} \\ &= g^{\text{SHA}(m) \cdot s^{-1}} \cdot v^{n \cdot s^{-1}} \\ &= \tilde{n} \end{aligned}$$

Daraus folgt

$$n = \tilde{n} \mod q$$

Bemerkung 9.11. Wie bei der ElGamal-Signatur ist es notwendig, dass Alice die zufällig gewählte Zahl z geheim hält und dass dieses z nur ein einziges Mal verwendet wird.

Beispiel 9.12. Alice wählt die Primzahl $p = 17509$ und den Primteiler $q = 1459$ von $p - 1 = 17508$ mit $\frac{p-1}{q} = 12$.

Ferner wählt sie zufällig $x = 8713 \in \{2, \dots, p - 2\}$ und berechnet

$$g = x^{12} = 4208 \mod p$$

Da $g \neq 1$, ist g ein Element der Ordnung q , und Alice wählt g .

Alice wählt $u = 960$ und berechnet

$$v = g^u = 7199 \mod p$$

1. Der private Schlüssel von Alice ist $k_{\text{pr},A} = (1459, 17509, 4208, 960)$.
2. Der öffentliche Schlüssel von Alice ist $k_{\text{pub},A} = (1459, 17509, 4208, 7199)$.
3. Alice veröffentlicht $k_{\text{pub},A} = (1459, 17509, 4208, 7199)$.

Alice will $\text{SHA}(m) = 1111$ ($\in \{1, \dots, q - 1\}$) signieren und geht dazu vor wie folgt:
Sie wählt zufällig $z = 773 \in \{2, \dots, q - 1\}$ und berechnet (mit Euklid)

$$z^{-1} = 1023$$

Alice berechnet

$$n' = g^z = 6275 \bmod p$$

und

$$n = n' = 439 \bmod q$$

Ferner setzt Alice

$$s = z^{-1} \cdot (\text{SHA}(m) + n \cdot u) = z^{-1} \cdot 35 = 789 \bmod q$$

und signiert ihre Nachricht mit

$$\text{sig}_A(m) = (439, 789)$$

Bob will die Signatur $\text{sig}_A(m) = (439, 789)$ der Nachricht M mit $\text{SHA}(m) = 1111$ überprüfen und geht vor wie folgt:

Bob berechnet (mit Euklid) $s^{-1} = 564 \pmod{q}$ und setzt

$$w_1 = \text{SHA}(m) \cdot s^{-1} = 693, w_2 = n \cdot s^{-1} = 1025 \pmod{q}$$

Bob berechnet

$$v^{w_2} \bmod p = 15975, g^{w_1} = 9070 \bmod p$$

und (als Zwischenergebnis)

$$\text{zwi} = g^{w_1} \cdot v^{w_2} = 6275 \bmod p$$

Bob berechnet

$$\tilde{n} = \text{zwi} = 439 \bmod q$$

und akzeptiert die Signatur, da $\tilde{n} = n$ in \mathbb{F}_q .

Bemerkung 9.12. Mit dem probabilistischen Signaturschema **RSA–PSS** (RSA Probabilistic Signature Scheme) gibt es ein in den **Public–Key Cryptography Standards PKCS** spezifiziertes auf der RSA–Signatur beruhendes probabilistisches Signaturverfahren, das (unter der RSA–Annahme und der Annahme, dass die verwendete Hashfunktion wie ein Zufallsorakel funktioniert) sicher gegen chosen–message–Angriffe ist.

10. Message Authentication Codes

Ziel eines **Message Authentication Codes MAC** ist es, durch die Berechnung einer geeigneten Prüfsumme oder Zusatzinformation sicherzustellen, dass eine übertragene Nachricht nicht manipuliert oder verändert wurde. Im Gegensatz zur digitalen Signatur geht es hier nicht um die Identität des Senders sondern nur um die Integrität einer (mit einem symmetrischen Verfahren verschlüsselten) Nachricht.

Formal gesehen besteht ein MAC–System aus drei Komponenten:

1. Einem Schlüsselerzeugungsverfahren, das einen symmetrischen Schlüssel k erzeugt und sicher an Alice und Bob verteilt.
2. Einem Prüfsummenverfahren **MAC**, das zu einem gegebenen Schlüssel k und einer gegebenen Nachricht m eine Prüfsumme

$$c = \mathbf{MAC}(k, m)$$

erzeugt.

3. Einem Verifikationsprozess **VER**, der für einen beliebigen Schlüssel einen Datenstrom (e, c) als authentisch akzeptiert, wenn es eine Nachricht m gibt mit $e = e_k(m)$ und $c = \mathbf{MAC}(k, m)$ und ablehnt, wenn das nicht der Fall ist, dh.

$$\mathbf{VER}(k, (e, c)) = 1 \iff \exists m \text{ mit } e_k(m) = e \text{ und } \mathbf{MAC}(k, m) = c$$

Eine MAC–System heißt **kryptographisch sicher**, wenn Catherine, selbst wenn sie Zugang zu einem Orakel hat, das den jeweils benutzten Schlüssel kennt und Catherine zu beliebigen, von ihr vorgegebenen Nachrichten m' das Paar $(e_k(m'), \mathbf{MAC}(k, m'))$ liefert (chosen–plaintext–Angriff), nicht in der Lage ist, eine existentielle Fälschung, also ein neues Paar (e, c) zu erzeugen, das mit nicht vernachlässigbarer Wahrscheinlichkeit den Verifikationsprozess erfolgreich übersteht und vom Empfänger akzeptiert wird.

Ein MAC–System hat die **Symmetrieeigenschaft**, wenn das Prüfsummenverfahren **MAC** Prüfsummen für symmetrisch verschlüsselte Nachrichten erzeugt, wobei Alice und Bob beide über den symmetrischen Schlüssel verfügen, und wenn der Verifikationsprozess für diese symmetrische verschlüsselten Daten korrekt funktioniert.

In der Praxis werden an ein MAC–System zur Kommunikation von Alice mit Bob die folgenden Anforderungen gestellt:

1. Es muss kryptographisch sicher sein.

2. Es muss die Symmetrieeigenschaft haben.
3. Es beachtet die Nachrichten- und die Prüfsummenlänge, dh. **MAC** macht aus einer Nachricht beliebiger Länge eine Prüfsumme fester Länge.
4. Es garantiert die Nachrichtenintegrität, dh. Bob kann sich mit an Sicherheit grenzender Wahrscheinlichkeit darauf verlassen, dass die Nachricht nicht manipuliert wurde, wenn das Ergebnis der Verifikation positiv ist.
5. Es garantiert die Nachrichtenauthentizität, dh. Bob kann sich mit an Sicherheit grenzender Wahrscheinlichkeit darauf verlassen, dass die Nachricht tatsächlich von Alice kommt.

Typischerweise werden MAC–Systeme eingesetzt, wenn Alice und Bob sicherstellen möchten, dass eine Veränderung der Nachricht c während der Übertragung erkannt wird.

1. Alice berechnet $c = \text{MAC}(k, m)$ und $e = e_k(m)$.
2. Alice schickt (e, c) an Bob.
3. Bob empfängt (e, c) und überprüft, ob Nachricht und Prüfsumme zusammenpassen, dh. ob $\text{MAC}(k, d_k(e)) = c$. Ist das der Fall, so akzeptiert er die Nachricht als integer und authentisch.

Jede bösartige oder zufällige Veränderung der Nachricht wird von Bob erkannt, da die Verifikation in diesem Fall fehlschlagen wird.

Der Schlüssel k ist ein gemeinsames Geheimnis von Alice und Bob (und nur diesen beiden). Daher kann Bob sicher sein, dass die Nachricht von Alice kommt, wenn die Verifikation ein positives Ergebnis liefert, denn niemand der k nicht kennt, kann c erzeugen.

Ein MAC–System kann jedoch nicht dazu benutzt werden, um die Urheberschaft einer Nachricht zu beweisen. Da sowohl Alice als auch Bob im Besitz des symmetrischen Schlüssels k sind, kann Bob mit $\text{MAC}(k, m)$ nicht nachweisen, dass die Nachricht m tatsächlich von Alice kommt, da er das Paar $(e_k(m), \text{MAC}(k, m))$ auch selbst erzeugt haben könnte. Dadurch unterscheiden sich MAC–Systeme ganz wesentlich von digitalen Signaturen.

Bemerkung 10.1. MAC–Systeme können auch eingesetzt werden, wenn die Nachricht selbst überhaupt nicht verschlüsselt wird.

Das ist der Fall, wenn es keine Notwendigkeit gibt, den Inhalt des gesendeten Dokumentes geheim zu halten, wenn es aber essentiell ist, dass das Dokument nicht verändert

wird. In diesem Fall werden also die Daten $(m, \text{MAC}(k, m))$ übertragen. Die Anforderungen an ein MAC–System sind daher hier besonders hoch und ein MAC–System muss insbesondere Angriffe auf diesen Typ abwehren können.

10.1. MAC–Systeme, die auf Hash–Funktionen basieren

Message Authentication Codes haben gewisse formale Ähnlichkeiten mit Hash–Funktionen, da auch hier Prüfsummen erzeugt werden, sie gehen aber in Ihrer Anwendung noch weit darüber hinaus. Da Hash–Funktionen in der Regel allgemein bekannt sind, kann Catherine selbst zu beliebigen Nachrichten selbst Hashs erzeugen, wodurch das Prinzip der Nachrichtenintegrität verletzt wird.

Trotzdem können kryptographisch sichere Hash–Funktionen h benutzt werden, um MAC–Systeme aufzusetzen. Die einfachsten Verfahren sind

$$\text{MAC}_{SP}(k, m) = h(k \| m) \quad (\text{secret prefix})$$

und

$$\text{MAC}_{SS}(k, m) = h(m \| k) \quad (\text{secret suffix})$$

Da Catherine den Schlüssel k nicht kennt, kann sie diese Werte nicht selbst erzeugen. Eine (zufällige) Veränderung der Nachricht führt mit an Sicherheit grenzender Wahrscheinlichkeit zu einem anderen Hash–Wert, sodass Nachricht und MAC nicht mehr zusammenpassen. Da außerdem (außer Bob) Catherine die einzige ist, die den geheimen Schlüssel k kennt, kann nur sie diesen Hash erzeugen, und daher kann Bob davon ausgehen, dass die Nachricht in der Tat von Alice kommt.

Allerdings haben beide Verfahren Schwächen und werden daher in der Praxis nicht verwendet.

Schwachstellen des Secret–Prefix–MAC

Die Schwäche des Secret–Prefix–MAC liegt in der Bauart der gängigen Hash–Funktionen (vergleiche Abschnitt 9). Diese sind üblicherweise so aufgebaut, dass die Nachricht m in Blöcke m_i geeigneter Länge zerlegt wird (eventuell mit Padding),

$$m = m_1 \| m_2 \| \cdots \| m_{t-1} \| m_t$$

Der Hashwert wird dann durch einen iterativen Prozess berechnet, also

$$h(m_1 \| m_2 \| \cdots \| m_{t-1} \| m_t) = h(h(m_1 \| m_2 \| \cdots \| m_{t-1}) \| m_t)$$

Diese Eigenschaft kann Catherine wie folgt ausnutzen:

Sie fängt eine Nachricht (m, c) von Alice an Bob ab und weiß, dass

$$c = \mathbf{MAC}_{SP}(k, m) = h(k \| m) = h(k \| m_1 \| \cdots \| m_t)$$

Sie wählt nun weitere Nachrichtenblöcke m_{t+1}, \dots, m_r aus und setzt

$$\begin{aligned}\tilde{m} &= m_1 \| \cdots \| m_t \| m_{t+1} \| \cdots \| m_r \\ \tilde{c} &= h(c \| m_{t+1} \| \cdots \| m_r)\end{aligned}$$

und schickt (\tilde{m}, \tilde{c}) an Bob.

Bob empfängt (\tilde{m}, \tilde{c}) und berechnet

$$\begin{aligned}\mathbf{MAC}_{SP}(k, \tilde{m}) &= h(k \| \tilde{m}) \\ &= h(k \| m \| m_{t+1} \| \cdots \| m_r) \\ &= h(h(k \| m) \| m_{t+1} \| \cdots \| m_r) \\ &= h(c \| m_{t+1} \| \cdots \| m_r)) \\ &= \tilde{c}\end{aligned}$$

Damit akzeptiert er die gefälschte Nachricht als authentisch und integer.

Catherine kann also beliebige Ergänzungen an dem Dokument vornehmen, etwa Zusätze und Ergänzungen in Verträge einfügen und dadurch die Nachricht massiv verändern.

Schwachstellen des Secret–Suffix–MAC

Ein Secret–Suffix–MAC–System wird unsicher, wenn es Catherine gelingt, Kolisionen (oder gar zweite Urbilder) für die Hash–funktion h zu finden (wie das bei SHA-1 teilweise schon der Fall ist). Falls Catherine eine Nachricht (m, c) von Alice an Bob abfängt und in der Lage ist, m so zu einer Nachricht \tilde{m} abzuändern, dass $h(m) = h(\tilde{m})$ (was bei SHA-1 teilweise schon der Fall ist), so leitet sie (\tilde{m}, c) an Bob weiter.

Bob empfängt (\tilde{m}, c) und berechnet

$$\mathbf{MAC}_{SS}(k, \tilde{m}) = h(\tilde{m} \| k) = h((h\tilde{m}) \| k) = h(h(m) \| k) = h(m \| k) = c$$

und akzeptiert die gefälschte Nachricht als authentisch und integer.

Falls es also Catherine gelingt m zu einem sinnvollen Text \tilde{m} anzuändern, etwa einer Variation des Textes die sin in einigen entscheidenden Details wie etwa den Beträgen vom Original unterscheidet, so kann sie Bob etwa ein falsches Angebot oder einen fehlerhaften Vertrag unterschieben.

10.2. Hash–Based Message Authentication Code HMAC

Bei diesem sehr gebräuchlichen Verfahren gehen wir davon aus dass die verwendete Hash–Funktion h nach dem Merkle–Damgård–Verfahren aufgebaut ist, und das sowohl

die Kompressionsrate r als auch die Hashlänge n der zugrundeliegenden Kompressionsfunktion

$$f : \mathbb{F}_2^{n+r} \longrightarrow \mathbb{F}_2^n$$

Vielfache von 8 sind (also in Byte gemessen werden können).

Wir betrachten ferner zwei Strings ipad (**inner padding**) und opad (**outer padding**), die (hexadezimal) wie folgt beschrieben werden können:

$$\begin{aligned} \text{ipad} &= 0x36 \quad \frac{r}{8}-\text{mal wiederholt} \\ \text{opad} &= 0x5C \quad \frac{r}{8}-\text{mal wiederholt} \end{aligned}$$

dh. ipad und opad sind Strings der Bitlänge r und der Form

$$\begin{aligned} \text{ipad} &= 0x0011011000110110\dots \\ \text{opad} &= 0x0101110001011100\dots \end{aligned}$$

Ferner wird k mit Nullen (nach rechts) aufgefüllt, bis es ebenfalls die Länge r hat oder durch $h(k)$ ersetzt, wenn es länger als r ist.

Definition 10.1. Für einen Schlüssel k und eine Nachricht m setzen wir

$$\mathbf{HMAC}(k, m) = h((k + \text{opad}) \| h((k + \text{ipad}) \| m))$$

Bemerkung 10.2. Es gilt

$$\mathbf{HMAC}(k, m) = f((k + \text{opad}) \| h((k + \text{ipad}) \| m))$$

Bemerkung 10.3. Ist h eine kryptographische Hash-Funktion, so ist **HMAC** kryptographisch sicher.

Ferner ist **HMAC** (durch die spezielle Struktur und das zweifache Einfüßen des Schlüssels k) weniger anfällig gegen Kollisionen als die unterliegende Hash-Funktion h selbst. Daher kann sogar SHA-1 für **HMAC** verwendet werden.

Umgekehrt kann gezeigt werden, dass **HMAC** nur gebrochen werden kann, wenn auch die zugrundeliegende Hash-Funktion gebrochen werden kann.

Bemerkung 10.4. Das Verifikationsverfahren für **HMAC** ist offensichtlich:

Bob hat das Paar (e, c) empfangen. Er ist im Besitz von k und kann sich aus dem Paar (e, c) (und erst recht aus (m, c)) daher $m = d_k(e)$ beschaffen. Also kann er

$$\mathbf{HMAC}(k, m) = h((k + \text{opad}) \| h((k + \text{ipad}) \| m))$$

berechnen und wird die Daten genau dann als integer und authentisch akzeptieren, falls

$$\mathbf{HMAC}(k, m) = c$$

Bemerkung 10.5. Falls der Schlüssel k länger als die Kompressionsrate r sein, so wird k in der Definition durch $h(k)$ ersetzt.

Bemerkung 10.6. Das **HMAC**-System ist weit verbreitet und wird in den IPsec-, SSH- und TLS-Protokollen sowie für JSON Web Tokens verwendet.

10.3. MAC mit Blockchiffren: CBC-MAC

Diese MAC-System beruht auf dem Cipher-Block-Chaining-Mode CBC der Blockchiffrierung , wie er in Abschnitt 2.2.2 behandelt wurde.

Dem Verfahren zugrunde liegt ein Blockchiffrierungsverfahren mit einer Blocklänge r . Eine Nachricht m wird daher in Blöcke der Länge r zerlegt (eventuell mit Padding),

$$m = m_1 \| m_2 \| \cdots \| m_{t-1} \| m_t, \quad m_i = \mathbb{F}_2^r$$

Ferner wird ein Initialisierungsvektor $\tilde{c}_0 \in \mathbb{F}_2^r$ festgelegt (z.B. $\tilde{c}_0 = (0, 0, \dots, 0)$) festgesetzt und ein Schlüssel \tilde{k} ausgewählt.

Definition 10.2. Für $i = 1, \dots, t$ setze

$$\tilde{c}_i = e_{\tilde{k}}(m_i + \tilde{c}_{i-1})$$

und

$$\text{CBC-MAC}(\tilde{k}, m) = \tilde{c}_t$$

Bemerkung 10.7. Das Verifikationsverfahren für CBC-MAC ist offensichtlich: Bob empfängt die Daten (e, c) . Da Bob den Schlüssel k für die Chiffrierung kennt, kann er $m = d_k(e)$ ermitteln. Da er auch den Schlüssel \tilde{k} (und den Initialisierungsvektor \tilde{c}_0) kennt, kann er mit m und \tilde{k} die Schritte des MAC-Erzeugungsverfahrens durchführen, also für $i = 1, \dots, t$ die Daten

$$\tilde{c}_i = e_{\tilde{k}}(m_i + \tilde{c}_{i-1})$$

und

$$\text{CBC-MAC}(\tilde{k}, m) = \tilde{c}_t$$

berechnen. Er wird die Daten genau dann als integer und authentisch akzeptieren, falls

$$\text{CBC-MAC}(\tilde{k}, m) = c$$

Bemerkung 10.8. Wird für die Verschlüsselung der Nachricht selbst ebenfalls der CBC-Mode verwendet, so ist es für die Sicherheit von CBC-MAC wesentlich, dass (bei Verwendung desselben Initialisierungsvektors) für die Verschlüsselung der eigentlichen

Nachricht ein anderer Schlüssel k verwendet wird als der für die Erzeugung von **CBC–MAC** verwendete Schlüssel \tilde{k} .

Wird nämlich in beiden Fällen derselbe Schlüssel k verwendet und wird die Nachricht m zu

$$e = e_1 \| e_2 \| \cdots \| e_{t-1} \| e_t$$

verschlüsselt, so folgt aus $\tilde{k} = k$, dass wir für alle $i = 1, \dots, t$ die Gleichheit

$$\tilde{c}_i = e_i$$

erhalten und damit gilt

$$c = \text{CBC–MAC}(k, m) = \tilde{e}_t = e_t$$

Fängt Catherine nun die Daten (e, c) ab und ersetzt sie e durch

$$e' = e'_1 \| e'_2 \| \cdots \| e'_{t-1} \| e_t$$

so führt die Dechiffrierung durch Bob zwar zu einen komplett anderen Klartext

$$m' = m'_1 \| m'_2 \| \cdots \| m'_{t-1} \| m'_t$$

(auch der letzte Block ändert sich dabei), die erneute Chiffrierung dieser Daten führt aber wieder zu

$$e' = e'_1 \| e'_2 \| \cdots \| e'_{t-1} \| e_t$$

und zwar egal ob Bob das zur Chiffratserzeugung durchführt oder zur MAC–Verifikation. Er erhält also

$$\text{CBC–MAC}(k, m) = e_t = c$$

und akzeptiert die Daten als integer und authentisch.

Solange Catherine also nur den letzten Datenblock beibehält, kann Sie die Nachricht davor nach belieben variieren und Bob wird die Nachricht als authentisch und integer akzeptieren.

10.4. Galois/Counter–Mode Message Authentication Code GMAC

Dieses MAC–System nutzt die Struktur endlicher Körper aus. Der Text wird in Blöcke der Länge 128 Bit aufgeteilt und jeder Block wird via

$$\mathbb{F}_2^{128} \cong \mathbb{F}_{2^{128}}$$

(wobei der Körper $\mathbb{F}_{2^{128}}$ durch die Relation $\alpha^{128} = \alpha^7 + \alpha^2 + \alpha + 1$ definiert wird) als Element der Körpers $\mathbb{F}_{2^{128}}$ aufgefasst. Es wird vorausgesetzt, dass es sich bei dem verwendeten Verschlüsselungsverfahren ebenfalls um ein 128–Bit Verschlüsselungsverfahren handelt (z.B. AES–128).

Verarbeitet werden dabei sowohl ein Block a von Daten, die nur authentiziert (aber nicht verschlüsselt) werden sollen (z.B. Protokolldaten) und verschlüsselte Daten e und wir schreiben

$$a = a_1 \| a_2 \| \dots \| a_m, \quad e = e_1 \| e_2 \| \dots \| e_n$$

wobei der letzte Block a_m bzw. e_n jeweils rechts mit Nullen aufgefüllt wird. Ferner bezeichnen wir mit $l(a)$ bzw. $l(e)$ die Länge von a bzw. e (in der 64–Bit–Darstellung) und wir setzen

$$s_j = \begin{cases} a_j & \text{für } j = 1, \dots, m \\ e_{j-m} & \text{für } j = m+1, \dots, m+n \\ l(a) \| l(e) & \text{für } j = m+n+1 \end{cases}$$

Ferner setzen wir

$$c_0 = 0 \quad (\text{in } \mathbb{F}_{2^{128}}), \quad h_0 = e_k(0) \quad (\text{mit } 0 \in \mathbb{F}_{2^{128}} = \mathbb{F}_2^{128})$$

Definition 10.3. Für $i = 1 \dots, m+n+1$ setze

$$c_i = \sum_{j=1}^i s_j \cdot h_0^{i-j+1}$$

(wobei Multiplikation und Addition in $\mathbb{F}_{2^{128}}$ durchzuführen sind) und

$$\mathbf{GMAC}(k, a, e) = c_{n+m+1}$$

Bemerkung 10.9. Die c_i lassen sich rekursiv nach der Formel

$$c_i = (c_{i-1} + s_i) \cdot h_0 \quad (i = 1, \dots, n+m+1)$$

berechnen. Es ist nämlich

$$\begin{aligned} c_i &= \sum_{j=1}^i s_j \cdot h_0^{i-j+1} \\ &= \sum_{j=1}^{i-1} s_j \cdot h_0^{i-j+1} + s_i \cdot h_0^{i-i+1} \\ &= \left(\sum_{j=1}^{i-1} s_j \cdot h_0^{i-1-j+1} \right) \cdot h_0 + s_i \cdot h_0 \\ &= c_{i-1} \cdot h_0 + s_i \cdot h_0 \\ &= (c_{i-1} + s_i) \cdot h_0 \end{aligned}$$

Bemerkung 10.10. Die **GMAC**–Methode ist nachweisbar sehr sicher, wenn es mit einer sicheren Blockverschlüsselung verwendet wird.

Bemerkung 10.11. Die **GMAC**–Methode kann gut parallelisiert und (trotz der Körperoperationen) sehr schnell implementiert werden.

Bemerkung 10.12. Die **GMAC**–Methode wird in verschiedenen IEEE–Standards verwendet, im IETF IPsec–Standard, in SSH, TLS 1.2 und TLS 1.3.

11. Elliptische Kurven

Asymmetrische kryptographische Verfahren, die in komplexen und umfangreichen digitalen Netzwerken zunehmend wichtig werden, beruhen in der Regel auf komplexen mathematischen Fragestellungen und Problemen. Oft geht es dabei um das Rechnen mit ganzen Zahlen und die Probleme stammen aus der Zahlentheorie (wie etwa beim Faktorisierungsproblem oder bei der Frage des diskreten Logarithmus). Es gibt jedoch auch in der Geometrie Objekte und Konstruktionen, die sich sehr gut für asymmetrische Kryptosysteme und public key–Verfahren eignen.

Eine Ansatz, der sehr intensiv genutzt wird und gerade in den letzten Jahren verstärkt Anwendung gefunden hat, ist die Konstruktion von public key–Verfahren mithilfe elliptischer Kurven. Dieser erweist sich vor allem vom Aufwand her als sehr viel effizienter als etwa RSA oder Diffie–Hellman.

In diesem Abschnitt betrachten wir einen beliebigen Körper k , z.B. $k = \mathbb{R}$, $k = \mathbb{C}$, $k = F_p$ für eine Primzahl p oder $k = F_q$ für eine Primzahlpotenz $q = p^e$. Bei den Anwendungen werden wir hier oft den Fall $q = 2^e$ haben, zunächst wollen wir aber voraussetzen, dass $\text{char}(k) = p > 3$ ist.

Denn Fall $p = 2$ werden wir später behandeln, der Fall $p = 3$ spielt in der Kryptographie keine Rolle und wird daher hier nicht betrachtet.

11.1. Definition elliptischer Kurven

Das Konzept allgemeiner ebener Kurven wurde in Anhang H erläutert. In diesem Abschnitt werden nun sehr spezielle Kurven untersucht.

Definition 11.1. Eine **affine elliptische Kurve** E über einem Körper k der Charakteristik $\text{char}(k) = 0$ oder $\text{char}(k) > 3$ ist eine Kurve, die durch ein Polynom $F(X, Y) \in k[X, Y]$ der Form

$$F(X, Y) = Y^2 - X^3 - aX - b$$

mit $a, b \in k$, für die $4a^3 + 27b^2 \neq 0$ gilt, gegeben ist.

Die Größe

$$\Delta = -16 \cdot (4a^3 + 27b^2)$$

heißt die **Diskriminante** von E und die Größe

$$j = -1728 \cdot \frac{4a^3}{\Delta}$$

heißt die j –**Invariante** von E .

Bemerkung 11.1. Die Punkte einer elliptischen Kurve, die durch ein Polynom der Form $F(X, Y) = Y^2 - X^3 - aX - b$ gegeben ist, sind die Lösungen der Gleichung

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

Insbesondere kann also ein Tupel $(r, s) \in k^2$ nur dann ein Punkt der elliptischen Kurve sein, wenn $r^3 + ar + b$ ein Quadrat in k ist (und dann ist notwendig s eine Wurzel daraus).

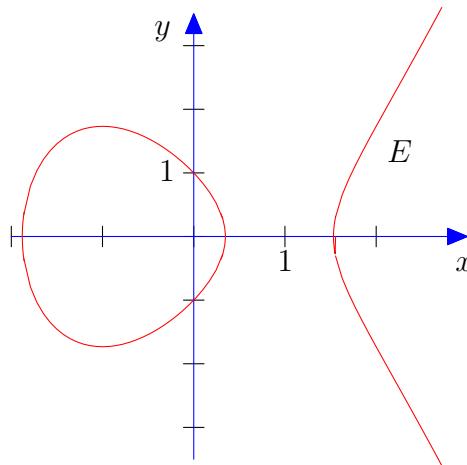
Bemerkung 11.2. Eine elliptische Kurve E ist immer symmetrisch zur x -Achse, d.h. mit (r, s) ist auch $(r, -s)$ in E .

Mit $s^2 = r^3 + ar + b$ gilt nämlich immer auch $(-s)^2 = r^3 + ar + b$.

Beispiel 11.1. Das Polynom $F(X, Y) = Y^2 - X^3 + 3X - 1 \in \mathbb{R}[X, Y]$ ist ein Polynom dieser Form (mit $a = -3$ und $b = 1$) und erfüllt

$$4 \cdot (-3)^3 + 27 \cdot 1^2 = -81 \neq 0$$

Also definiert $F(X, Y)$ eine affine elliptische Kurve. Diese hat etwa folgende Gestalt:

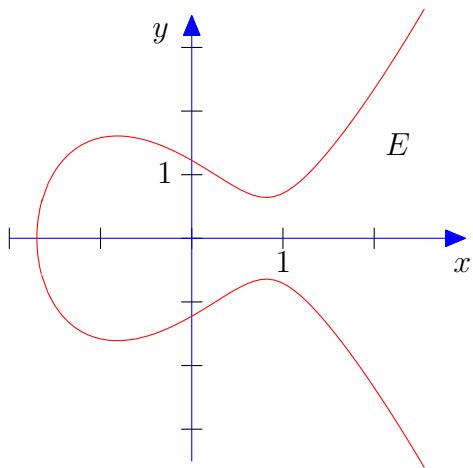


Diese Gestalt mit einer „Küste mit Insel“ ist typisch und tritt sehr häufig auf bei elliptischen Kurven.

Beispiel 11.2. Das Polynom $F(X, Y) = Y^2 - X^3 + 2X - \frac{3}{2} \in \mathbb{R}[X, Y]$ ist ein Polynom dieser Form (mit $a = -2$ und $b = \frac{3}{2}$) und erfüllt

$$4 \cdot (-2)^3 + 27 \cdot \left(\frac{3}{2}\right)^2 = 28.75 \neq 0$$

Also definiert $F(X, Y)$ eine affine elliptische Kurve. Diese hat etwa folgende Gestalt:

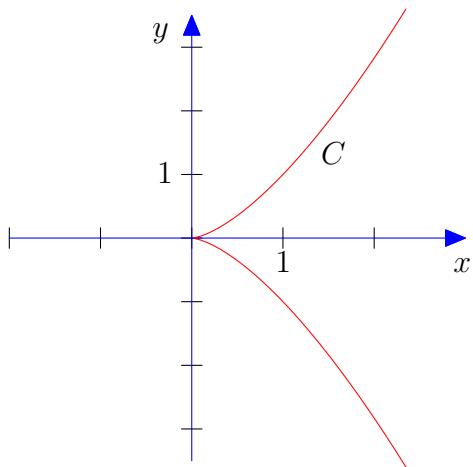


Diese Gestalt mit einem „Kleiderbügel“ ist ebenfalls typisch und neben der „Küste mit Insel“ die einzige andere mögliche Form einer elliptischen Kurven (über \mathbb{R}).

Beispiel 11.3. Das Polynom $F(X, Y) = Y^2 - X^3 \in \mathbb{R}[X, Y]$ hat eine gewisse Ähnlichkeit mit einem Polynom, das eine elliptische Kurve definiert (mit $a = 0, b = 0$). Hierfür gilt jedoch

$$4 \cdot 0^3 + 27 \cdot 0^2 = 0$$

und damit handelt es sich nicht um eine elliptische Kurve. Die durch $F(X, Y)$ definierte ebene Kurve hat die Gestalt

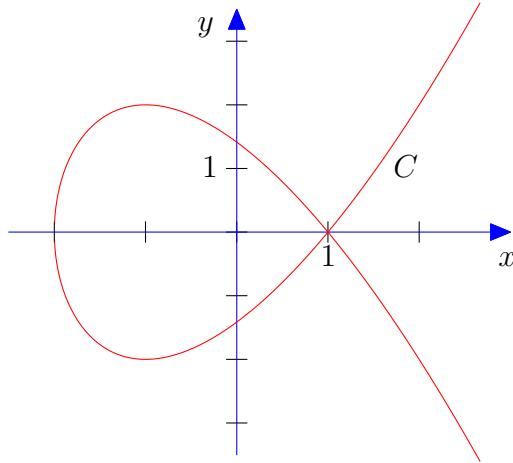


Das Bild unterscheidet sich von dem Bild einer elliptischen Kurve durch die Spitze im Punkt $(0, 0)$. Elliptische Kurven haben keine Spitzen.

Beispiel 11.4. Das Polynom $F(X, Y) = Y^2 - X^3 + 3X - 2 \in \mathbb{R}[X, Y]$ hat ebenfalls eine gewisse Ähnlichkeit mit einem Polynom, das eine elliptische Kurve definiert (mit $a = -3, b = 2$). Hierfür gilt jedoch

$$4 \cdot (-3)^3 + 27 \cdot 2^2 = 0$$

und damit handelt es sich nicht um eine elliptische Kurve. Die durch $F(X, Y)$ definierte ebene Kurve hat die Gestalt



Das Bild unterscheidet sich von dem Bild einer elliptischen Kurve durch die Kreuzung im Punkt $(1, 0)$. Elliptische Kurven haben keine Kreuzungen.

Beispiel 11.5. Wir betrachten das Polynom

$$F(X, Y) = Y^2 - X^3 + 3X - 3 = Y^2 + 12X^3 + 3X + 10 \in \mathbb{F}_{13}[X, Y]$$

Mit $a = -3 = 10$ und $b = 3 = 9$ gilt hierfür

$$4 \cdot 10^3 + 27 \cdot 9^2 = 6 \neq 0$$

und daher definiert $F(X, Y)$ eine elliptische Kurve über \mathbb{F}_{13} , und zwar

$$\begin{aligned} E = & \{(0, 4), (0, 9), (1, 1), (1, 12), (4, 4), (4, 9), (5, 3), (5, 10), \\ & (7, 0), (8, 6), (8, 7), (9, 4), (9, 9), (11, 1), (11, 12)\} \end{aligned}$$

Beispiel 11.6. Wir betrachten das Polynom

$$F(X, Y) = Y^2 - X^3 + 3X - 3 = Y^2 + 4X^3 + 3X + 2 \in \mathbb{F}_5[X, Y]$$

Mit $a = -3 = 2$ und $b = 3$ gilt hierfür

$$4 \cdot 2^3 + 27 \cdot 3^2 = 4 \neq 0$$

und daher definiert $F(X, Y)$ eine elliptische Kurve E über \mathbb{F}_5 . Zur Berechnung der Punkte von E betrachten wir diese als Lösungen der Gleichung

$$y^2 = x^3 - 3x + 3$$

Die rechte Seite muss also ein Quadrat sein. Das sind in \mathbb{F}_5 genau die Elemente 0, 1 und 4, wobei

$$0 = 0^2, \quad 1 = 1^2 = 4^2, \quad 4 = 2^2 = 3^2$$

Setzen wir $f(x) = x^3 - 3x + 3$, so gilt hierfür

$$f(0) = 3, f(1) = 1, f(2) = 0, f(3) = 1, f(4) = 0$$

Damit kommen als x -Komponente 1, 2, 3 und 4 in Frage, und y muss so gewählt sein, dass $y^2 = f(x)$. Deshalb gilt

$$E = \{(1, 1), (1, 4), (2, 0), (3, 1), (3, 4), (4, 0)\}$$

Beispiel 11.7. Wir betrachten das Polynom

$$F(X, Y) = Y^2 - X^3 + 3X - 3 = Y^2 + 6X^3 + 3X + 4 \in \mathbb{F}_7[X, Y]$$

Mit $a = -3 = 4$ und $b = 3$ gilt hierfür

$$4 \cdot 2^3 + 27 \cdot 3^2 = 6 \neq 0$$

und daher definiert $F(X, Y)$ eine elliptische Kurve E über \mathbb{F}_7 . Zur Berechnung der Punkte von E betrachten wir diese als Lösungen der Gleichung

$$y^2 = x^3 - 3x + 3$$

Die rechte Seite muss also ein Quadrat sein. Das sind in \mathbb{F}_7 genau die Elemente 0, 1, 2 und 4, wobei

$$0 = 0^2, \quad 1 = 1^2 = 6^2, \quad 2 = 3^2 = 4^3, \quad 4 = 2^2 = 5^2$$

Setzen wir $f(x) = x^3 - 3x + 3$, so gilt hierfür

$$f(0) = 3, f(1) = 1, f(2) = 5, f(3) = 0, f(4) = 6, f(5) = 1, f(6) = 5$$

Damit kommen als x -Komponente 1, 3 und 5 in Frage, und y muss so gewählt sein, dass $y^2 = f(x)$. Deshalb gilt

$$E = \{(1, 1), (1, 6), (3, 0), (5, 1), (5, 6)\}$$

Die Beispiele 11.5, 11.6 und 11.7 zeigen, dass für der Berechnung der Punkte auf elliptischen Kurven das Ziehen von Quadratwurzeln (aus Quadraten) über endlichen Körpern sehr wichtig ist. Da in der Kryptographie mit sehr großen Körpern (mit ca. 2^{300} bis 2^{500} Elementen) gearbeitet wird, scheidet die Methode des Ausprobierens aus und algorithmische Ansätze sind notwendig. Diese wurden bereits in Anhang B ausführlich behandelt.

Damit lassen sich die Punkte auf einer elliptischen Kurve E über einem Körper \mathbb{F}_p effizient berechnen. Ist diese Kurve gegeben durch $F(X, Y) = Y^2 - X^3 - aX - b$ in $\mathbb{F}_p[X, Y]$, so betrachte zunächst

$$f(x) = x^3 + ax + b$$

und gehe vor wie folgt:

1. Für $r \in F_p$ überprüfe, ob $f(r)$ ein Quadrat ist, dh. überprüfe, ob $f(r)^{\frac{p-1}{2}} = 1$ (durch iteriertes Quadrieren).
2. Ist das der Fall, so bestimme eine Quadratwurzel s aus $f(r)$.
3. Die Punkte (r, s) und $(r, -s)$ sind die einzigen Punkte auf E mit x -Koordinate r .

Beispiel 11.8. Wir betrachten die elliptische Kurve E (bzw. \bar{E}) über dem Körper \mathbb{F}_{19} , gegeben durch das Polynom

$$F(X, Y) = Y^2 - X^3 - 8 \cdot X - 4 \quad \in \mathbb{F}_{19}[X, Y]$$

und setzen $f(x) = x^3 + 8 \cdot x + 4$. Hier ist also $a = 8$ und $b = 4$ und damit

$$4 \cdot a^3 + 27 \cdot b^2 = 4 \cdot 8^3 + 8 \cdot 4^2 = 10 \neq 0 \quad \text{in } \mathbb{F}_{19}$$

und deshalb handelt es sich tatsächlich um eine elliptische Kurve.

Da $\frac{p-1}{2} = 9$, ist eine Zahl $z \in \mathbb{F}_{19}$ genau dann ein Quadrat, wenn $z^9 = 1$, und da hier $p+1 = 20$ durch 4 teilbar ist, kann aus einem Quadrat q eine Wurzel durch Bilden von $w = q^{\frac{p+1}{4}} = q^5$ gezogen werden.

Wir gehen nun alle Elemente \mathbb{F}_{19} durch:

Für $r = 0$ ist $f(0) = 4$. Das ist offensichtlich ein Quadrat (schon in \mathbb{Z}), es gilt aber natürlich auch $f(0)^9 = 4^9 = 1$ (in \mathbb{F}_{19}). Eine Wurzel aus $f(0)$ ist

$$s = f(0)^5 = 4^5 = 17$$

und damit sind $P_1 = (0, 17)$ und $P_2 = (0, -17) = (0, 2)$ zwei Punkte auf E .

Für $r = 1$ ist $f(1) = 13$. Hierfür gilt $f(1)^9 = 13^9 = 18 \neq 1$ und damit gibt es keine Punkt auf E mit x -Komponente 1.

Für $r = 2$ ist $f(2) = 9$. Das ist offensichtlich ein Quadrat (schon in \mathbb{Z}), es gilt aber natürlich auch $f(2)^9 = 9^9 = 1$. Eine Wurzel aus $f(2)$ ist

$$s = f(2)^5 = 2^5 = 16$$

Damit sind $P_3 = (2, 16)$ und $P_4 = (2, -16) = (2, 3)$ zwei weitere Punkte auf E .

Für $r = 3$ ist $f(3) = 17$. Hierfür gilt $f(3)^9 = 17^9 = 1$, und damit ist $f(3)$ ein Quadrat in \mathbb{F}_{19} (was hier nicht offensichtlich ist), und eine Wurzel aus $f(3)$ ist

$$s = f(3)^5 = 17^5 = 6$$

Damit sind $P_5 = (3, 6)$ und $P_6 = (3, -6) = (3, 13)$ zwei weitere Punkte auf E .

Für $r = 4$ ist $f(4) = 5$. Hierfür gilt $f(4)^9 = 5^9 = 1$, und damit ist $f(4)$ ein Quadrat in \mathbb{F}_{19} (was ebenfalls nicht offensichtlich ist), und eine Wurzel aus $f(4)$ ist

$$s = f(4)^5 = 5^5 = 9$$

Damit sind $P_7 = (4, 9)$ und $P_8 = (4, -9) = (4, 10)$ zwei weitere Punkte auf E .

Für $r = 5$ ist $f(5) = 17$. Wir wissen bereits (aus dem Fall $r = 3$), dass $f(5)$ ein Quadrat in \mathbb{F}_{19} ist und Wurzel $s = 6$ eine Wurzel daraus. Damit sind $P_9 = (5, 6)$ und $P_{10} = (5, -6) = (5, 13)$ zwei weitere Punkte auf E .

Für $r = 6$ ist $f(6) = 2$. Hierfür gilt $f(6)^9 = 2^9 = 18 \neq 1$ und damit gibt es keine Punkt auf E mit x -Komponente 6.

Für $r = 7$ ist $f(7) = 4$. Das haben wir im Fall $r = 0$ schon erhalten und die Wurzel $s = 17$ aus $f(7)$ gefunden. Damit sind $P_{11} = (7, 17)$ und $P_{12} = (7, -17) = (7, 2)$ zwei weitere Punkte auf E .

Für $r = 8$ ist $f(8) = 10$. Hierfür gilt $f(8)^9 = 10^9 = 18 \neq 1$ und damit gibt es keine Punkt auf E mit x -Komponente 8.

Für $r = 9$ ist $f(9) = 7$. Hierfür gilt $f(9)^9 = 7^9 = 1$, und damit ist $f(9)$ ein Quadrat in \mathbb{F}_{19} , und eine Wurzel aus $f(9)$ ist

$$s = f(9)^5 = 7^5 = 11$$

Damit sind $P_{13} = (9, 11)$ und $P_{14} = (9, -11) = (9, 8)$ zwei weitere Punkte auf E .

Für $r = 10$ ist $f(10) = 1$. Das ist offensichtlich ein Quadrat (schon in \mathbb{Z}), und hier ist auch offensichtlich, dass $f(10)^9 = 1^9 = 1$. Eine Wurzel aus $f(10)$ ist

$$s = f(1)^5 = 1^5 = 1$$

Damit sind $P_{15} = (10, 1)$ und $P_{16} = (10, -1) = (10, 18)$ weitere Punkte auf E .

Für $r = 11$ ist $f(11) = 17$. Wir haben schon im Fall $r = 3$ gesehen, dass 17 ein Quadrat in \mathbb{F}_{19} ist und $s = 6$ eine Wurzel daraus, und damit sind $P_{17} = (11, 6)$ und $P_{18} = (11, -6) = (11, 13)$ zwei weitere Punkte auf E .

Für $r = 12$ ist $f(12) = 4$. Wir haben schon im Fall $r = 0$ gesehen, dass 4 ein Quadrat in \mathbb{F}_{19} ist und $s = 17$ eine Wurzel daraus, und damit sind $P_{19} = (12, 17)$ und $P_{20} = (12, -17) = (12, 2)$ zwei weitere Punkte auf E .

Für $r = 13$ ist $f(13) = 6$. Hierfür gilt $f(13^9) = 6^9 = 1$, und damit ist $f(13)$ ein Quadrat in \mathbb{F}_{19} , und eine Wurzel aus $f(13)$ ist

$$s = f(13)^5 = 6^5 = 5$$

Damit sind $P_{21} = (13, 5)$ und $P_{22} = (13, -5) = (13, 14)$ weitere Punkte auf E .

Für $r = 14$ ist $f(14) = 10$. Hierfür gilt $f(14^9) = 10^9 = 18 \neq 1$ (siehe auch $r = 8$), und damit gibt es keine Punkt auf E mit x -Komponente 14.

Für $r = 15$ ist $f(15) = 3$. Hierfür gilt $f(15^9) = 3^9 = 18 \neq 1$ und damit gibt es keine Punkt auf E mit x -Komponente 15.

Für $r = 16$ ist $f(16) = 10$. Hierfür gilt $f(16^9) = 10^9 = 18 \neq 1$ (siehe auch $r = 8$), und damit gibt es keine Punkt auf E mit x -Komponente 16.

Für $r = 17$ ist $f(17) = 18$. Hierfür gilt $f(17^9) = 18^9 = 18 \neq 1$, und damit gibt es keine Punkt auf E mit x -Komponente 17.

Für $r = 18$ ist $f(18) = 14$. Hierfür gilt $f(18^9) = 14^9 = 18 \neq 1$ und damit gibt es keine Punkt auf E mit x -Komponente 18.

Insgesamt erhalten wir also mit dem unendlich fernen Punkt ∞ :

$$\begin{aligned} \overline{E} = & \{(0, 17), (0, 2), (2, 16), (2, 3), (3, 6), (3, 13), (4, 9), (4, 10), \\ & (5, 6), (5, 13), (7, 17), (7, 2), (9, 11), (9, 8), (10, 1), (10, 18), \\ & (11, 6), (11, 13), (12, 17), (12, 2), (13, 5), (13, 14), \infty\} \end{aligned}$$

Bemerkung 11.3. Jede einzelne Rechnung in Beispiel 11.8 ist relativ einfach und einsichtig, insgesamt wird das Verfahren aber natürlich aufwendig, wenn p groß wird. Ein Verfahren, dass die Ermittlung der Punkte auf einer elliptischen Kurve signifikant abkürzt, ist nicht bekannt.

Bemerkung 11.4. Wir werden noch (als Folgerung aus der Hasse–Weil–Schranke 12.3) sehen, dass in sehr grober Näherung etwa für die Hälfte aller für jedes Elemente $r \in \mathbb{F}_q$ ein Punkt $P = (r, s) \in E$ existiert. Das machte es relativ leicht, durch eine Zufallssuche einen Punkt auf einer elliptischen Kurve zu finden. Schwierig ist es aber, alle Punkte auf der Kurve zu finden.

11.2. Elliptische Kurven und Geraden

Nun wollen wir die Struktur elliptischer Kurven genauer untersuchen:

Satz 11.1. Wir betrachten eine affine elliptische Kurve E über k wobei $k = \mathbb{F}_p$, mit $p > 3$ oder $k = \mathbb{R}$ oder $k = \mathbb{C}$, die gegeben ist durch

$$F(X, Y) = Y^2 - X^3 - aX - b \in k[X, Y]$$

Ferner betrachten wir zwei Punkte $P_1 = (r_1, s_1)$ und $P_2 = (r_2, s_2)$ auf E , die Gerade L durch die Punkte P_1 und P_2 , und wir setzen $m = \frac{s_2 - s_1}{r_2 - r_1}$ (die Steigung der Geraden L).

a) Ist $r_2 \neq r_1$ und gilt sowohl $\frac{3r_1^2 + a}{2s_1} \neq m$ als auch $\frac{3r_2^2 + a}{2s_2} \neq m$, so schneidet die Gerade L die Kurve E in genau einem weiteren Punkt $P_3 = (r_3, s_3)$ und die Koordinaten dieses Punktes sind gegeben durch

$$r_3 = m^2 - r_1 - r_2, \quad s_3 = m \cdot (r_3 - r_1) + s_1 = m^3 - 2m \cdot r_1 - m \cdot r_2 + s_1$$

b) Ist $r_2 \neq r_1$ und gilt $\frac{3r_1^2 + a}{2s_1} = m$, so schneidet die Gerade L die Kurve E nicht mehr, aber P_1 ist ein doppelter Schnittpunkt (L ist eine Tangente an E in P_1).

c) Ist $r_2 \neq r_1$ und gilt $\frac{3r_2^2 + a}{2s_2} = m$, so schneidet die Gerade L die Kurve E nicht mehr, aber P_2 ist ein doppelter Schnittpunkt (L ist eine Tangente an E in P_2).

d) Ist $r_2 = r_1$ (und damit $s_2 = -s_1$), so schneidet die Gerade L die Kurve E nicht mehr.

Beweis: Der Nachweis ist eine etwas umfangreiche technische Übung im Rechnen mit Gleichungen:

a) Die Gerade L durch P_1 und P_2 hat die Steigung $m = \frac{s_2 - s_1}{r_2 - r_1}$ und ist gegeben durch

$$l(x) = m \cdot (x - r_1) + s_1$$

Die Punkte (x, y) die sowohl auf L als auch auf E liegen, sind also genau die Punkte, die die Gleichungen

$$\begin{aligned} y &= m \cdot (x - r_1) + s_1 \\ y^2 &= x^3 + ax + b \end{aligned}$$

erfüllen, also die Punkte $(x, m \cdot (x - r_1) + s_1)$, für die gilt

$$(m \cdot (x - r_1) + s_1)^2 = x^3 + ax + b$$

Multiplizieren wir diese Gleichung aus, so erhalten wir

$$m^2 \cdot (x - r_1)^2 + 2m \cdot (x - r_1) \cdot s_1 + s_1^2 = x^3 + ax + b$$

also

$$m^2 \cdot x^2 - 2m^2 \cdot x \cdot r_1 + m^2 \cdot r_1^2 + 2m \cdot s_1 \cdot x - 2m \cdot r_1 \cdot s_1 + s_1^2 = x^3 + ax + b$$

bzw.

$$x^3 - m^2 x^2 + (2m^2 \cdot r_1 - 2m \cdot s_1 + a) \cdot x + b - r_1 \cdot m^2 + 2m \cdot r_1 \cdot s_1 - s_1^2 = 0$$

Damit muss also x eine Nullstelle des Polynoms

$$f(X) = X^3 - m^2 X^2 + (2m^2 \cdot r_1 - 2m \cdot s_1 + a) \cdot X + b - r_1 \cdot m^2 + 2m \cdot r_1 \cdot s_1 - s_1^2$$

sein.

Wir wissen bereits, dass (r_1, s_1) und (r_2, s_2) auf L und E liegen, dass also r_1 und r_2 Nullstellen von $f(X)$ sind. Damit schreibt sich $f(X)$ nach Polynomdivision als

$$f(X) = (X - r_1) \cdot (X - r_2) \cdot g(X)$$

wobei $\deg(g(X)) = 3 - 2 = 1$, also $g(X)$ linear. Da $f(X)$ normiert ist, ist auch $g(X)$ normiert, d.h. $g(X) = X - r_3$ und damit

$$f(X) = (X - r_1) \cdot (X - r_2) \cdot (X - r_3)$$

Multiplizieren wir nun das aus, so erhalten wir

$$f(X) = X^3 - (r_1 + r_2 + r_3) \cdot X^2 - (r_1 r_2 + r_1 r_3 + r_2 r_3) \cdot X - r_1 r_2 r_3$$

und durch Koeffizientenvergleich (bei X^2) erhalten wir

$$-m^2 = -r_1 - r_2 - r_3$$

also

$$r_3 = m^2 - r_1 - r_2$$

Daher ist der Punkt $P_3 = (r_3, s_3)$ mit

$$r_3 = m^2 - r_1 - r_2, \quad s_3 = m \cdot (r_3 - r_1) + s_1 = m^3 - 2mr_1 - mr_2 + s_1$$

sowohl auf E als auch auf L .

- b) Wir beschränken uns hier auf den Fall $k = \mathbb{R}$. Der erforderliche Formalismus lässt sich auf beliebige Körper übertragen, das würde aber den Rahmen dieser Veranstaltung sprengen.

Wir gehen dabei vor wie in Teil a) und betrachten das Polynom

$$f(X) = X^3 - m^2 X^2 + (2m^2 \cdot r_1 - 2m \cdot s_1 + a) \cdot X + b - r_1 \cdot m^2 + 2m \cdot r_1 \cdot s_1 - s_1^2$$

Wie wir dort gesehen haben, sind die x -Komponenten der Punkte P von $E \cap L$ Nullstellen dieses Polynoms. Für die Ableitung dieses Polynoms gilt

$$f'(X) = 3X^2 - 2m^2 X + 2m^2 \cdot r_1 - 2m \cdot s_1 + a$$

Da $m = \frac{3r_1^2 + a}{2s_1}$ folgt

$$\begin{aligned} f'(r_1) &= 3r_1^2 - 2m^2 \cdot r_1 + 2m^2 \cdot r_1 - 2m \cdot s_1 + a \\ &= 3r_1^2 - 2m \cdot s_1 + a \\ &= 3r_1^2 - 2 \cdot \frac{3r_1^2 + a}{2s_1} \cdot s_1 + a \\ &= 3r_1^2 - (3r_1^2 + a) + a \\ &= 0 \end{aligned}$$

wir haben also in diesem Fall

$$f(r_1) = 0, \quad f'(r_1) = 0$$

und damit ist r_1 eine doppelte Nullstelle von $f(X)$. Da r_2 eine weitere (davon verschiedene) Nullstelle von $f(X)$ ist, muss gelten

$$f(X) = (X - r_1)^2 \cdot (X - r_2)$$

Insbesondere kann also L keine weiteren Schnittpunkte mit E haben, und der Schnittpunkt bei P_1 ist ein doppelter.

- c) Geht genauso wie b).
- d) Ist $r_2 = r_1$ und $s_2 = -s_1$, so ist die Gerade durch P_1 und P_2 parallel zur y -Achse und gegeben durch die Gleichung $x = r_1$. Eingesetzt in die Gleichung der elliptischen Kurve ergibt sich, dass die Punkte $P = (x, y) \in L \cap E$ die folgenden Gleichungen erfüllen müssen:

$$\begin{aligned} x &= r_1 \\ y^2 &= x^3 + ax + b \end{aligned}$$

und daraus ergibt sich, dass

$$y^2 = r_1^3 + a \cdot r_1 + b$$

gelten muss. Da die Gleichung $y^2 = r_1^3 + a \cdot r_1 + b$ aber schon die beiden Lösungen $y_1 = s_1$ und $y_2 = s_2$ hat, kann es keine weiteren Lösungen mehr geben.

Satz 11.2. Wir betrachten eine affine elliptische Kurve E über k wobei $k = \mathbb{F}_p$, mit $p > 3$ oder $k = \mathbb{R}$ oder $k = \mathbb{C}$, die gegeben ist durch

$$F(X, Y) = Y^2 - X^3 - aX - b \in k[X, Y]$$

Ferner betrachten wir einen Punkt $P = (r, s)$ auf E .

1. Ist $s = 0$, so ist die Gerade T , gegeben durch $x = r$ eine Tangente an E in P , und T schneidet E nur im Punkt P und in keinem weiteren Punkt mehr.
2. Ist $s \neq 0$, so ist die Gerade T , gegeben durch die Gleichung

$$y = m \cdot (x - r) + s \quad \text{wobei } m = \frac{3r^2 + a}{2s}$$

eine Tangente an E in P , und T schneidet E in noch genau einem weiteren Punkt $Q = (u, v)$, wobei

$$u = m^2 - 2 \cdot r, \quad v = m \cdot (u - r) + s = m^3 - 3mr + s$$

Beweis:

1. Dass T tangential an E in P ist, folgt aus dem Satz über implizite Funktionen (der im dritten Semester behandelt werden wird). Ferner liegt ein Punkt (x, y) auf der durch $X = r$ definierten Parallelen zur y -Achse und auf E , wenn

$$\begin{aligned} x &= r \\ y^2 &= x^3 + ax + b \end{aligned} \tag{11.1}$$

also wenn

$$y^2 = r^3 + ar + b$$

Da aber $y = 0$ eine Lösung ist, muss $r^3 + ar + b = 0$ gelten, und damit ist $y = 0$ die einzige Lösung für diese Gleichungen (11.1), und daher ist P der einzige Punkt auf T und E .

2. Auch hier ist mit $m = \frac{3r^2+a}{2s}$ die Gerade $T : y = m \cdot (x - r) + s$ tangential an E in P nach dem Satz über implizite Funktionen.

Ein Punkt (x, y) liegt auf T und auf E , wenn

$$\begin{aligned} x &= m \cdot (x - r) + s \\ y^2 &= x^3 + ax + b \end{aligned} \tag{11.2}$$

also wenn

$$(m \cdot (x - r) + s)^2 = x^3 + ax + b$$

Auflösen und Zusammenfassen der Gleichungen (11.2) liefert (wie im Beweis des vorhergehenden Satzes), dass

$$x^3 - m^2 x^2 + (2m^2 r + 2ms + a) \cdot x + b - r^2 m^2 + 2mrs - s^2 = 0$$

dh. x ist eine Nullstelle des Polynoms

$$f(X) = X^3 - m^2 X^2 + (2m^2 r + 2ms + a) \cdot X + b - r^2 m^2 + 2mrs - s^2 \in k[X]$$

Nach Voraussetzung ist aber schon $f(r) = 0$ und T tangential an E , dh. r ist eine doppelte Nullstelle von $f(X)$, und $f(X)$ schreibt sich als

$$f(X) = (X - r)^2 \cdot (X - u)$$

Ausmultiplizieren liefert

$$f(X) = X^3 - (2r + u) \cdot X^2 + (r^2 + 2ru) \cdot X - r^2 u$$

und durch Koeffizientenvergleich erhalten wir

$$-m^2 = -2r + u$$

woraus $u = m^2 - 2r$ folgt, und damit ist $Q = (u, v)$ mit

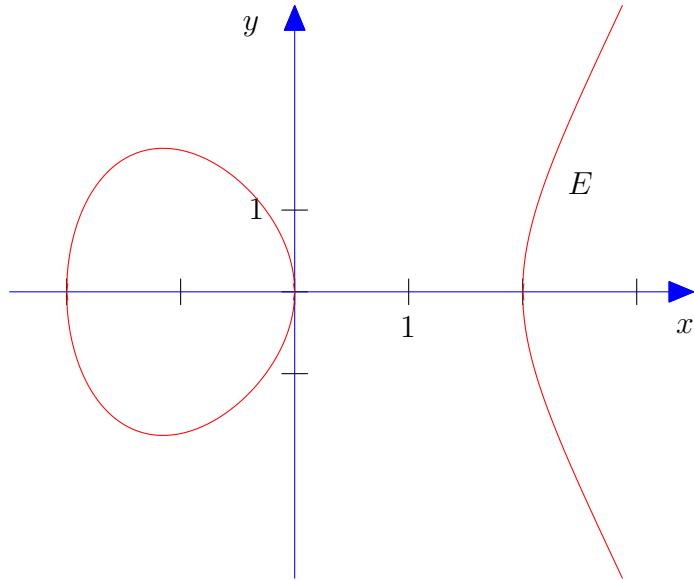
$$v = m \cdot (u - r) + s = m^3 - 3mr + s$$

der einzige weitere Punkt auf E und T .

Beispiel 11.9. Wir betrachten die elliptische Kurve E über \mathbb{R} , gegeben durch

$$F(X, Y) = Y^2 - X^3 + 4X \in \mathbb{R}[X, Y]$$

(mit $a = -4$ und $b = 0$), also die Kurve



Die beiden Punkte $P_1 = \left(-\frac{3}{2}, \sqrt{\frac{21}{8}}\right)$ und $P_2 = \left(-\frac{1}{2}, \sqrt{\frac{15}{8}}\right)$ liegen auf E , und die Gerade L durch P_1 und P_2 schneidet E noch in einem weiteren Punkt $P_3 = (r_3, s_3)$, wobei sich die Koordinaten wie folgt berechnen:

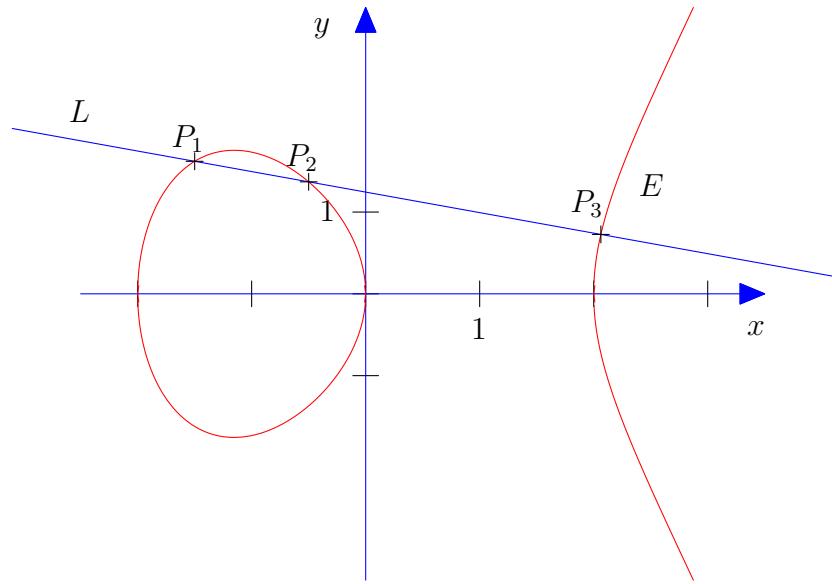
Die Steigung von L ist $m = \sqrt{\frac{15}{8}} - \sqrt{\frac{21}{8}}$. Damit ist

$$r_3 = m^2 - r_1 - r_2 = \frac{15}{8} - 2 \cdot \sqrt{\frac{15 \cdot 21}{8 \cdot 8}} + \frac{21}{8} + 2 = \frac{31}{2} - \frac{3}{4} \cdot \sqrt{35} \approx 2.0629$$

und

$$s_3 = m^3 - 2mr_1 - mr_2 + s_1 \approx 0.7263$$

Die Graphik bestätigt diese Rechnungen:



Der Punkt $P = (-1, \sqrt{3})$ liegt auf E und die Tangente T an E in P hat die Steigung

$$m = \frac{3 \cdot (-1)^2 + (-4)}{2 \cdot \sqrt{3}} = -\frac{1}{2 \cdot \sqrt{3}}$$

Daher ist der weitere Schnittpunkt von T mit E gegeben durch $Q = (u, v)$ mit

$$u = m^2 - 2 \cdot (-1) = \frac{25}{12}$$

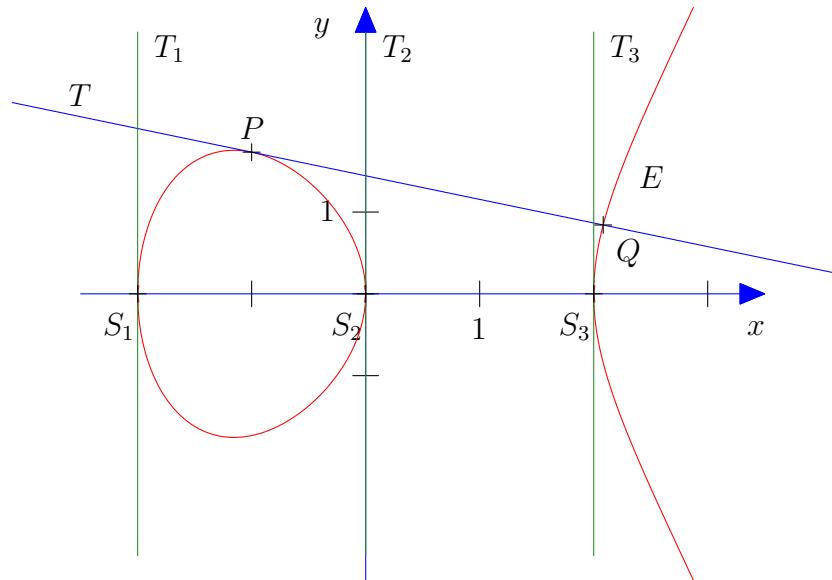
und

$$v = m^3 - 3m \cdot (-1) + \sqrt{3} = \frac{35}{72} \cdot \sqrt{3}$$

Die Punkte auf E mit y -Koordinate 0 sind die Lösungen von

$$x^3 - 4x = 0$$

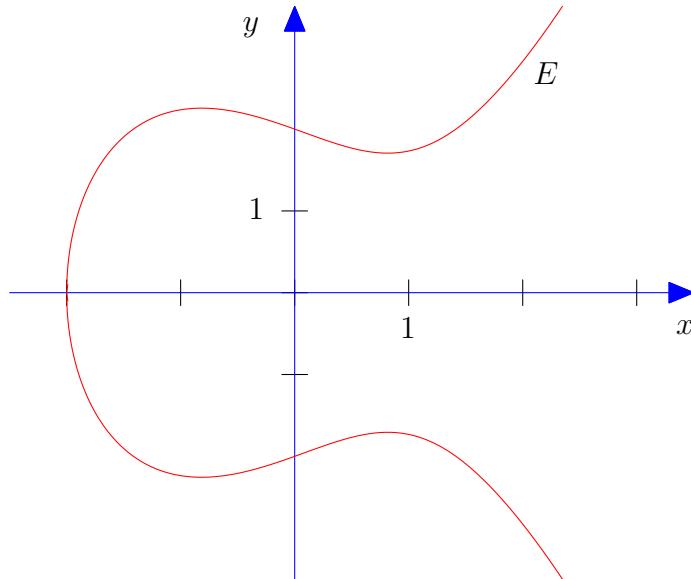
also die Punkte $(-2, 0)$, $(0, 0)$ und $(2, 0)$, und dort sind die Tangenten senkrecht. Auch das bestätigt die Graphik.



Beispiel 11.10. Wir betrachten die elliptische Kurve E über \mathbb{R} , gegeben durch

$$F(X, Y) = Y^2 - X^3 + 2X - 4 \in \mathbb{R}[X, Y]$$

(mit $a = -2$ und $b = 4$), also die Kurve



Dann liegen beide Punkte $P_1 = \left(-\frac{3}{2}, \sqrt{\frac{29}{8}}\right)$ und $P_2 = \left(-\frac{1}{2}, \sqrt{\frac{39}{8}}\right)$ auf E , und die Gerade L durch P_1 und P_2 schneidet E noch in einem weiteren Punkt $P_3 = (r_3, s_3)$. Die Gerade

L hat die Steigung

$$m = \sqrt{\frac{39}{8}} - \sqrt{\frac{29}{8}} \approx 0.3040$$

und die Koordinaten von P_3 ergeben sich (näherungsweise) als

$$r_3 \approx 2.0924, \quad s_3 \approx 2.9960$$

Ferner liegt auch der Punkt $P = (-1, \sqrt{5})$ auf E und die Tangente T an E in P hat die Steigung

$$m = \frac{3 \cdot (-1)^2 + (-2)}{2 \cdot \sqrt{5}} = \frac{1}{2 \cdot \sqrt{5}} \approx 0.2236$$

Diese Tangente T schneidet E noch in dem weiteren Punkt $Q = (u, v)$ mit

$$u = 2.0500, \quad v = 2.9181$$

Beispiel 11.11. Wir betrachten die elliptische Kurve E über \mathbb{F}_{13} , die durch das Polynom

$$F(X, Y) = Y^2 - X^3 + 3X - 3 = Y^2 + 12X^3 + 3X + 10 \in \mathbb{F}_{13}[X, Y]$$

definiert wird. Wir wissen schon, dass diese aus den Punkten

$$\begin{aligned} E = & \{(0, 4), (0, 9), (1, 1), (1, 12), (4, 4), (4, 9), (5, 3), (5, 10), \\ & (7, 0), (8, 6), (8, 7), (9, 4), (9, 9), (11, 1), (11, 12)\} \end{aligned}$$

besteht.

Insbesondere sind also die Punkte $P_1 = (1, 1)$ und $P_2 = (5, 3)$ auf E . Die Gerade L durch diese beiden Punkte hat die Steigung

$$m = \frac{3 - 1}{5 - 1} = \frac{2}{4} = \frac{1}{2} = 7$$

(denn $2 \cdot 7 = 1$ in \mathbb{F}_{13}), und damit hat der dritte Punkt $P_3 = (r_3, s_3)$, der sowohl auf E als auch auf L liegt, die Koordinaten

$$r_3 = m^2 - r_1 - r_2 = 7^2 - 1 - 3 = 10 - 1 - 3 = 4$$

und

$$s_3 = m^3 - 2mr_1 - mr_2 + s_1 = 7^3 - 2 \cdot 7 \cdot 1 - 7 \cdot 5 + 1 = 5 - 1 - 9 + 1 = 9$$

also $P_3 = (4, 9)$.

Ferner sind die Punkte $P_1 = (5, 3)$ und $P_2 = (8, 6)$ auf E . Die Gerade L durch diese beiden Punkte hat die Steigung

$$m = \frac{6 - 3}{8 - 5} = \frac{3}{3} = 1$$

und damit hat der dritte Punkt $P_3 = (r_3, s_3)$, der sowohl auf E als auch auf L liegt, die Koordinaten

$$r_3 = m^2 - r_1 - r_2 = 1^2 - 5 - 8 = 1$$

und

$$s_3 = m^3 - 2mr_1 - mr_2 + s_1 = 1^3 - 2 \cdot 1 \cdot 5 - 1 \cdot 8 + 3 = 12$$

Die Tangente T an E im Punkt $P = (1, 1)$ hat die Steigung

$$m = \frac{3 \cdot 1^2 + 10}{2 \cdot 1} = 0$$

und damit hat der zweite Punkt $Q = (u, v)$ auf E und T die Koordinaten

$$u = m^2 - 2r = -2 = 11, \quad v = m^3 - 3mr + s = 1$$

Beispiel 11.12. Wir betrachten die elliptische Kurve E über \mathbb{F}_{17} , die durch das Polynom

$$F(X, Y) = Y^2 - X^3 + 3X - 3 = Y^2 + 16X^3 + 3X + 14 \in \mathbb{F}_{17}[X, Y]$$

(mit $a = -3 = 14$ und $b = 3$) definiert wird. Dann sind die Punkte $P_1 = (3, 2)$ und $P_2 = (11, 3)$ auf E . Die Gerade L durch P_1 und P_2 hat die Steigung $m = 15$ und schneidet E in dem weiteren Punkt $P_3 = (7, 11)$:

Es ist $m = \frac{3-2}{11-3} = \frac{1}{8} = 15$ (in \mathbb{F}_{17}), und damit

$$r_3 = m^2 - r_1 - r_2 = 10^2 - 11 - 3 = 7$$

und

$$s_3 = m^3 - 2mr_1 - mr_2 + s_1 = 15^3 - 2 \cdot 15 \cdot 3 - 15 \cdot 11 + 2 = 11$$

Ferner ist der Punkt $P = (9, 5)$ auf E . Die Tangente an E in P hat die Steigung $m = 15$ und schneidet E in dem weiteren Punkt $Q = (14, 6)$.

Die Tangente hat die Steigung

$$m = \frac{3r^2 + a}{2s} = \frac{3 \cdot 9^2 + 14}{2 \cdot 5} = 12 \cdot (3 \cdot 9^2 + 14) = 7$$

und damit gilt

$$u = m^2 - 2 \cdot r = 7^2 - 2 \cdot 9 = 14$$

und

$$v = m^3 - 3mr + s = 7^3 - 3 \cdot 7 \cdot 9 + 5 = 6$$

12. Arithmetik elliptischer Kurven

Die Eigenschaften, dass die „meisten“ Geraden genau drei Schnittpunkte mit E haben, ist der entscheidende Punkt für die Kryptographie. Dafür müssen wir jedoch aus „die meisten“ noch „alle“ (in einem geeigneten Sinn) machen. Wir betrachten dazu wieder einen beliebigen Körper, wobei wir in diesem Abschnitt $\text{char}(k) = 0$ oder $\text{char}(k) = p > 3$ voraussetzen.

12.1. Projektive elliptische Kurven

Zunächst ist es notwendig, elliptische Kurven um einen weiteren Punkt zu ergänzen:

Definition 12.1. Ein **projektive elliptische Kurve** \bar{E} über einem Körper k besteht aus einer affinen Kurve E (gegeben durch ein Polynom $F(X, Y) = Y^2 - X^3 - aX - b$ wie oben) zusammen mit einem „unendlich fernen Punkt“ ∞ .

Bemerkung 12.1. Ist E gegeben durch $F(X, Y) = Y^2 - X^3 - aX - b$ so schreiben wir für einen Punkt P auf der zugehörigen projektiven elliptischen Kurve \bar{E} auch

- $P = [1 : r : s]$ falls $P = (r, s) \in E$ (also $(r, s) \in k^2$ mit $F(r, s) = 0$).
- $P = [0 : 0 : 1]$ falls $P = \infty$.

Beispiel 12.1. Die projektive elliptische Kurve \bar{E} über \mathbb{F}_{13} , die durch das Polynom

$$F(X, Y) = Y^2 - X^3 + 3X - 3 = Y^2 + 12X^3 + 3X + 10 \in \mathbb{F}_{13}[X, Y]$$

definiert wird, besteht aus den 16 Punkten

$$\begin{aligned} \bar{E} = & \{(0, 4), (0, 9), (1, 1), 1, 12), (4, 4), (4, 9), (5, 3), (5, 10), \\ & (7, 0), (8, 6), (8, 7), (9, 4), (9, 9), (11, 1), (11, 12), \infty\} \end{aligned}$$

Bemerkung 12.2. Der Punkt ∞ auf einer projektiven Kurve \bar{E} ist ein Schnittpunkt aller Geraden mit \bar{E} , die parallel zur y -Achse sind, die also E selbst nicht genügend oft schneiden.

Bemerkung 12.3. Sind P_1 und P_2 zwei Punkte auf einer elliptischen Kurve und ist die Gerade L durch P_1 und P_2 die Tangente an E in P_1 , so bezeichnen wir P_1 auch als den dritten Punkt auf der Geraden L und auf E , dh. in diesem Fall zählt P_1 doppelt und $P_3 = P_1$.

Mit dieser Konvention schneidet jetzt jede Gerade die Kurve \bar{E} in genau drei Punkten.

12.2. Addition auf elliptischen Kurven in der Charakteristik $p > 3$

Ist \overline{E} eine projektive Kurve über einem Körper k , gegeben durch ein Polynom der Form $F(X, Y) = Y^2 - X^3 - aX - b$, so definieren wir eine Operation

$$+ : \overline{E} \times \overline{E} \longrightarrow \overline{E}$$

wie folgt:

1. Es ist $\infty + \infty = \infty$.
2. Für $P = (r, s) \in E$ ist $P + \infty = \infty + P = P$.
3. Für $P_1 = (r_1, s_1), P_2 = (r_2, s_2) \in E$ mit $r_1 \neq r_2$ bezeichne $P_3 = (r_3, s_3)$ den dritten Punkt, der auf E und der Geraden L durch P_1 und P_2 liegt, so ist auch $R = (r_3, -s_3) \in E$ und $P_1 + P_2 = R$.
4. Für $P_1 = (r_1, s_1)$ und $P_2 = (r_2, s_2) \in E$ mit $s_1 = -s_2 (\neq 0)$ ist $P_1 + P_2 = \infty$.
5. Für $P = (r, 0)$ auf E ist $P + P = \infty$.
6. Für $P = (r, s) \in E$ mit $s \neq 0$ bezeichne $Q = (u, v)$ den Schnittpunkt der Tangente an E in P (als der Gerade mit Steigung $m = \frac{3r^2+a}{2s}$ durch P), so ist $R = (u, -v) \in E$ und $P + P = R$.

Beachten Sie dabei, dass wir ausgenutzt haben, dass mit einem Punkt $(r, s) \in E$ auch $(r, -s) \in E$ gilt (vergleiche Bemerkung 11.2).

Satz 12.1. *Mit dieser Operation $+$ wird $(\overline{E}, +)$ zu einer kommutativen Gruppe mit neutralem Element ∞ .*

Beweis: Einen Beweis dieser nicht-trivialen Aussage übersteigt den Rahmen dieser Vorlesung. Sie finden ihn z.B. im Buch SILVERMAN, J.: *The Arithmetic of Elliptic Curves*, in Kapitel III, Proposition 2.2.

Bemerkung 12.4. Die Operation $+$ auf \overline{E} ist gerade so gewählt, dass für eine Gerade L , die drei Schnittpunkte P_1, P_2 und P_3 mit E hat, gilt

$$P_1 + P_2 + P_3 = \infty$$

Folgerung 12.2. *Ist $P = (r, s) \in \overline{E}$, so ist $-P = (r, -s)$.*

Bemerkung 12.5. Wir schreiben kurz $2 \cdot P$ für $P + P$ und allgemeiner

$$n \cdot P = \underbrace{P + P + \cdots + P}_{n-\text{mal}}$$

Bemerkung 12.6. Bei der Berechnung der Summe $P_1 + P_2$ von zwei Punkten $P_1 = (r_1, s_1)$ und $P_2 = (r_2, s_2)$ mit $r_1 \neq r_2$ muss nicht speziell untersucht werden, ob die Steigung $m = \frac{s_2 - s_1}{r_2 - r_1}$ mit einer der Tangentensteigungen übereinstimmt oder nicht.

Betrachten wir dazu etwa die elliptische Kurve \bar{E} über \mathbb{F}_{13} , die durch das Polynom

$$F(X, Y) = Y^2 - X^3 + 3X - 3 = Y^2 + 12X^3 + 3X + 10 \in \mathbb{F}_{13}[X, Y]$$

definiert wird (siehe Beispiel 12.1), und die beiden Punkte $P_1 = (4, 4)$ und $P_2 = (8, 7)$, so gilt hierfür

1. Die Steigung der Geraden L durch die beiden Punkte ist

$$m = \frac{7 - 4}{8 - 4} = \frac{3}{4} = 10 \cdot 3 = 4$$

2. Die Koordinaten des dritten Schnittpunktes $Q = (u, v)$ von L und E sind

$$\begin{aligned} u &= m^2 - r_1 - r_2 = 4^2 - 4 - 8 = 4 \\ v &= m \cdot (u - r_1) + s_1 = 4 \cdot (4 - 4) + 4 = 4 \end{aligned}$$

3. Es gilt

$$P_1 + P_2 = -Q = (4, -4) = 4, 9)$$

Der „dritte“ Schnittpunkt Q ist also hier wieder der Punkt P_1 . Das liegt daran, dass die Tangentensteigung an den Punkt P_1 gegeben ist durch

$$\frac{3 \cdot r_1^2 + a}{2 \cdot s_1} = \frac{3 \cdot 4^2 + 10}{2 \cdot 4} = \frac{6}{8} = 5 \cdot 6 = 4 = m$$

also mit der Steigung der Geraden L übereinstimmt. Demgemäß ist P_1 doppelt zu zählen, und daher ist es auch korrekt, dass der dritte Schnittpunkt wieder P_1 ist. Die Berechnungsformel liefert das also automatisch.

Dadurch können wir uns Fallunterscheidungen sparen und erhalten für zwei Punkte $P_1 = (r_1, s_1)$ und $P_2 = (r_2, s_2)$ für die Berechnung von $P_1 + P_2$ die folgende vereinfachte Vorgehensweise:

1. Falls $r_1 \neq r_2$, so setze $m = \frac{s_2 - s_1}{r_2 - r_1}$ und

$$u = m^2 - r_1 - r_2, \quad v = m \cdot (u - r_1) + s_1$$

Dann ist

$$P_1 + P_2 = (u, -v)$$

2. Falls $r_1 = r_2$ und $s_1 \neq s_2$, so ist notwendig $s_2 = -s_1$ und

$$P_1 + P_2 = \infty$$

3. Falls $r_1 = r_2$ und $s_1 = s_2$ und $s_1 \neq 0$, so setze $m = \frac{3 \cdot r_1^2 + a}{2 \cdot s_1}$ und

$$u = m^2 - 2 \cdot r_1, \quad v = m \cdot (u - r_1) + s_1$$

Dann ist

$$P_1 + P_2 = 2 \cdot P_1 = (u, -v)$$

4. Falls $r_1 = r_2$ und $s_1 = s_2$ und $s_1 = 0$, so ist

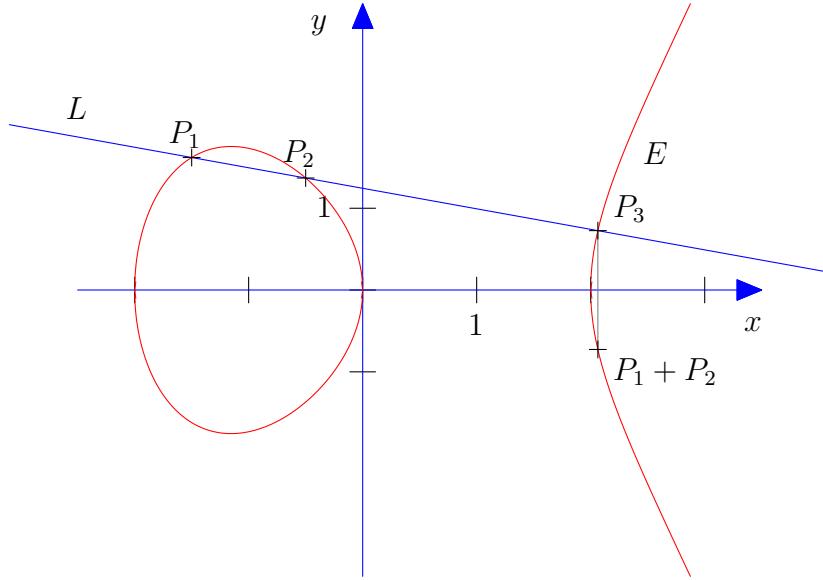
$$P_1 + P_2 = 2 \cdot P_1 = \infty$$

Beispiel 12.2. Wir betrachten wieder die elliptische Kurve E über \mathbb{R} aus Beispiel 11.9, gegeben durch

$$F(X, Y) = Y^2 - X^3 + 4X \in \mathbb{R}[X, Y]$$

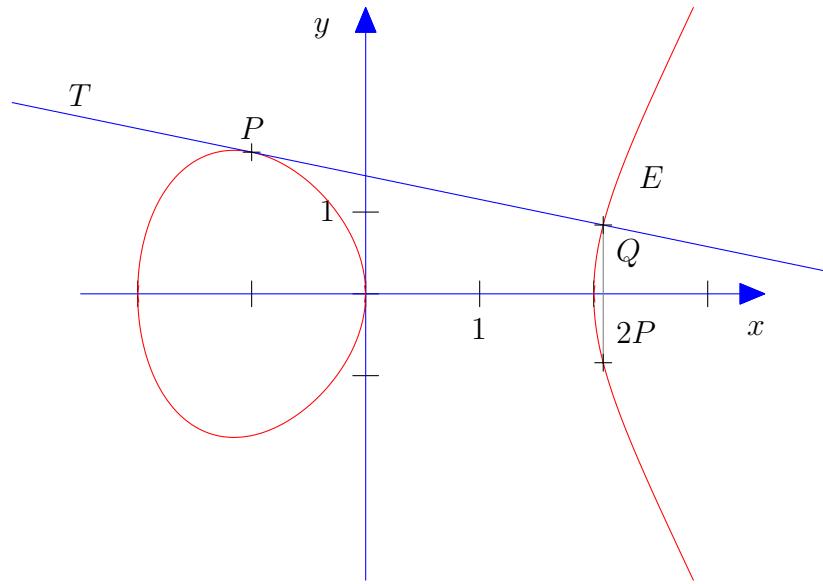
Die beiden Punkte $P_1 = \left(-\frac{3}{2}, \sqrt{\frac{21}{8}}\right)$ und $P_2 = \left(-\frac{1}{2}, \sqrt{\frac{15}{8}}\right)$ liegen auf E , und die Gerade L durch P_1 und P_2 schneidet E noch in einem weiteren Punkt $P_3 = (2.0629, 0.7263)$ (gerundet). Damit ist auch $R = (2.0629, -0.7263)$ auf E und es gilt

$$P_1 + P_2 = (2.0629, -0.7263)$$



Der Punkt $P = (-1, \sqrt{3})$ liegt auf E und die Tangente T an E in P schneidet E im Punkt $Q = (2.0833, 0.8420)$. Daher ist auch $R = (2.0833, -0.8420)$ auf E und

$$2 \cdot P = (2.0833, -0.8420)$$



Beispiel 12.3. Wir betrachten die elliptische Kurve \bar{E} über \mathbb{F}_{13} , die durch das Polynom

$$F(X, Y) = Y^2 - X^3 + 3X - 3 = Y^2 + 12X^3 + 3X + 10 \in \mathbb{F}_{13}[X, Y]$$

definiert wird. Dann sind die Punkte $P_1 = (1, 1)$ und $P_2 = (5, 3)$ auf E und die Gerade L durch diese beiden Punkte schneidet E im Punkt $P_3 = (4, 9)$ (vergleiche Beispiel 11.11), also ist

$$(1, 1) + (5, 3) = (4, -9) = (4, 4)$$

Analog gilt

$$(5, 3) + (8, 6) = (1, 1)$$

denn die Gerade L durch die Punkte $(5, 3)$ und $(8, 6)$ enthält auch noch den Punkt $(1, 12)$ von E , und

$$2 \cdot (1, 1) = (11, 12)$$

denn die Tangente T an den Punkt $(1, 1)$ von E geht auch durch $(11, 1) \in E$.

Beispiel 12.4. Wir betrachten die elliptische Kurve E über \mathbb{F}_{17} , die durch das Polynom

$$F(X, Y) = Y^2 - X^3 + 3X - 3 = Y^2 + 16X^3 + 3X + 14 \in \mathbb{F}_{17}[X, Y]$$

definiert wird. Dann sind die Punkte $P_1 = (3, 2)$ und $P_2 = (11, 3)$ auf E und

$$(3, 2) + (11, 3) = (7, 6)$$

Ferner ist der Punkt $P = (9, 5)$ auf E und

$$2 \cdot P = (14, 11)$$

Beispiel 12.5. Wir betrachten die elliptische Kurve E über \mathbb{F}_{19} , die durch das Polynom

$$F(X, Y) = Y^2 - X^3 - 10X - 3 = Y^2 + 18X^3 + 9X + 16 \in \mathbb{F}_{19}[X, Y]$$

definiert wird. Dann sind die Punkte $P_1 = (5, 11)$ und $P_2 = (5, 8)$ auf E und

$$(5, 11) + (5, 8) = \infty$$

Beispiel 12.6. Wir betrachten die elliptische Kurve E über \mathbb{F}_{19} , die durch das Polynom

$$F(X, Y) = Y^2 - X^3 - 7X - 13 = Y^2 + 18X^3 + 12X + 6 \in \mathbb{F}_{19}[X, Y]$$

definiert wird. Dann sind die Punkte $P_1 = (2, 15)$ und $P_2 = (6, 10)$ auf E und der dritte Punkt $P_3 = (r_3, s_3)$ auf E und der Geraden durch P_1 und P_2 berechnet sich wie folgt:

$$m = \frac{10 - 15}{6 - 2} = \frac{14}{4} = 13$$

und damit

$$r_3 = 13^2 - 2 - 6 = 9, \quad s_3 = 13 \cdot (9 - 2) + 15 = 11$$

also ist

$$(2, 15) + (6, 10) = (9, 8)$$

Außerdem ist der Punkt $P = (7, 5)$ auf E . Dann gilt

$$8 \cdot (7, 5) = (14, 10)$$

Das können wir wie folgt einsehen:

Die Steigung der Tangente T an E im Punkte P ist

$$m = \frac{3 \cdot 7^2 + 7}{2 \cdot 5} = \frac{2}{10} = 4$$

und damit hat der zweite Punkt Q auf T und E die Koordinaten $Q = (r_2, s_2)$ mit

$$r_2 = 4^2 - 2 \cdot 7 = 2, \quad s_2 = 4 \cdot (2 - 7) + 5 = 4$$

also ist $2 \cdot P = (2, 15)$. Die Tangente T_2 an E in $2 \cdot P$ hat die Steigung

$$m = \frac{3 \cdot 2^2 + 7}{2 \cdot 15} = \frac{0}{11} = 0$$

und damit hat der zweite Punkt Q auf T_2 und E die Koordinaten $Q = (r_4, s_4)$ mit

$$r_4 = 0^2 - 2 \cdot 2 = 15, \quad s_4 = 0 \cdot (15 - 2) + 15 = 15$$

also ist $4 \cdot P = 2 \cdot (2 \cdot P) = (15, 4)$. Die Tangente T_4 an E in $4 \cdot P$ hat die Steigung

$$m = \frac{3 \cdot 15^2 + 7}{2 \cdot 4} = \frac{17}{8} = 14$$

und damit hat der zweite Punkt Q auf T_4 und E die Koordinaten $Q = (r_8, s_8)$ mit

$$r_8 = 14^2 - 2 \cdot 15 = 14, \quad s_8 = 14 \cdot (14 - 15) + 4 = 9$$

also ist

$$8 \cdot P = 2 \cdot (2 \cdot (2 \cdot P)) = (14, 10)$$

Diese Gruppen spielen nun eine wichtige Rolle. Entscheidend ist dabei, dass sie eine kontrollierte Größe haben. Hierzu bezeichnen wir für eine endliche Menge M mit $|M|$ die Mächtigkeit von M , also die Anzahl der Elemente in M .

Satz 12.3 (Hasse–Weil–Schranke). *Ist $k = \mathbb{F}_q$ ein endlicher Körper und \bar{E} eine elliptische Kurve über k , so gilt*

$$q + 1 - 2 \cdot \sqrt{q} \leq |\bar{E}| \leq q + 1 + 2 \cdot \sqrt{q}$$

Beweis: Einen Beweis dieser sehr tiefen Aussage finden Sie z.B. auch in dem Buch SILVERMAN, J.: *The arithmetic of Elliptic Curves*, in Kapitel V, Theorem 1.1.

Bemerkung 12.7. Die Hasse–Weil–Schranke besagt, dass sich für große q die Anzahl der Elemente auf einer elliptischen Kurve etwa in der Größenordnung von q bewegt. Da mit einem Punkt $P = (r, s)$ auf E auch immer $Q = (r, -s)$ auf E liegt, da das die beiden einzigen Punkte mit x -Komponente r sind, und da es höchstens 3 Punkte auf E mit $s = 0$ gibt, bedeutet das, dass es etwa für die Hälfte aller $r \in \mathbb{F}_q$ eine Punkt P mit x -Komponente r auf E gibt.

Beispiel 12.7.

a) Eine elliptische Kurve $\overline{E}/\mathbb{F}_{13}$ erfüllt

$$7 \leq |\overline{E}| \leq 21$$

b) Eine elliptische Kurve $\overline{E}/\mathbb{F}_{167}$ erfüllt

$$143 \leq |\overline{E}| \leq 193$$

c) Eine elliptische Kurve $\overline{E}/\mathbb{F}_{13^3}$ erfüllt

$$2105 \leq |\overline{E}| \leq 2291$$

Definition 12.2. Für einen Punkt $P \in \overline{E}$ heißt

$$\text{ord}(P) = \min\{n > 0 \mid n \cdot P = \infty\}$$

die **Ordnung** von P .

Satz 12.4. Ist $k = \mathbb{F}_q$ ein endlicher Körper und \overline{E}/k eine elliptische Kurve, so gilt für die Gruppe $(\overline{E}, +)$ entweder

\overline{E} ist zyklisch, $\overline{E} = \mathbb{Z}_n$ für ein $n \in \mathbb{N}$.

oder

$\overline{E} = \mathbb{Z}_n \times \mathbb{Z}_{n \cdot t}$ für $n, t \in \mathbb{N}$, $n \geq 2$ und $n|(q-1)$.

Im zweiten Fall ist $n \cdot t = \max\{\text{ord}(P) \mid P \in \overline{E}\}$.

Beweis: Ein Beweis dieser Aussage folgt aus SILVERMAN, J.: *The arithmetic of Elliptic Curves*, Kapitel V, Corollary 6.4 und Corollary 8.1.

Beispiel 12.8. Wir betrachten wieder die elliptische Kurve \overline{E} über \mathbb{F}_{13} , die durch das Polynom

$$F(X, Y) = Y^2 - X^3 + 3X - 3 = Y^2 + 12X^3 + 3X + 10 \in \mathbb{F}_{13}[X, Y]$$

definiert wird. Dann hat \overline{E} genau 16 Punkte, wie wir schon gesehen haben. Es gilt dabei

$$\text{ord}((0, 4)) = 16$$

Damit hat \overline{E} ein Element der Ordnung 16 und ist daher notwendig zyklisch,

$$\overline{E} = \mathbb{Z}_{16}$$

Beispiel 12.9. Wir betrachten die elliptische \overline{E} über \mathbb{F}_{11} , die durch das Polynom

$$F(X, Y) = Y^2 - X^3 + X = Y^2 + 10X^3 + X \in \mathbb{F}_{11}[X, Y]$$

definiert wird. Diese hat die Gestalt

$$\overline{E} = \{(0, 0), (1, 0), (4, 4), (4, 7), (6, 1), (6, 10), (8, 3), (8, 8), (9, 4), (9, 7), (10, 0), \infty\}$$

Hierfür gilt:

1. Die Elemente $(0, 0)$, $(1, 0)$ und $(10, 0)$ haben die Ordnung 2.
2. Die Elemente $(4, 4)$ und $(4, 7)$ haben die Ordnung 3.
3. Die Elemente $(6, 1)$, $(6, 10)$, $(8, 3)$, $(8, 8)$, $(9, 4)$ und $(9, 7)$ haben die Ordnung 6.
4. Das Elemente ∞ hat die Ordnung 1.

Damit ist \overline{E} nicht zyklisch und

$$\overline{E} = \mathbb{Z}_2 \times \mathbb{Z}_6$$

Oft hilft ein einfaches Kriterium bei der Entscheidung ob eine elliptische Kurve eine zyklische Gruppe ist:

Satz 12.5. Ist $|E| = m$ und gibt es keine Primzahl r , für die sowohl $r|(q-1)$ als auch $r^2|m$ gilt, so ist E zyklisch,

$$\overline{E} = \mathbb{Z}_m$$

Beweis: Andernfalls müsste gelten

$$\overline{E} = \mathbb{Z}_n \times \mathbb{Z}_{n \cdot t}$$

für ein $n \geq 2$, also $m = n^2 \cdot t$. Dann ist aber jeder Primteiler r von n ein Teiler von $q-1$ und $r^2|m$, ein Widerspruch.

13. Elliptische Kurven in der Charakteristik 2

In der Praxis der Kryptographie spielen Körper der Charakteristik 2 eine besondere Rolle, und es werden sehr häufig auch elliptische Kurven E/\mathbb{F}_q bzw. \bar{E}/\mathbb{F}_q für $q = 2^l$ betrachtet, da sich die Arithmetik dieser Körper gut für Prozessoren eignet.

13.1. Beschreibung elliptischer Kurven in der Charakteristik 2

Elliptische Kurven in der Charakteristik 2 unterscheiden sich in ihrer Beschreibung deutlich von den Kurven über Körpern der Charakteristik 0 oder $p > 3$, die wir bis jetzt betrachtet haben. In ihrer Normalform werden die in der Kryptographie relevanten Kurven beschrieben durch ein Polynom der Form

$$F(X, Y) = Y^2 + XY + X^3 + aX^2 + b \in \mathbb{F}_q[X, Y]$$

mit $b \neq 0$ (wobei auch hier bei \bar{E} der Punkt ∞ hinzukommt).

Bemerkung 13.1. In der Sprache der Mathematik ist eine elliptische Kurve \bar{E} über einem Körper k eine glatte projektive Kurve vom Geschlecht $g = 1$. Diese kann (im Sinne der projektiven Geometrie) immer beschrieben werden durch ein geeignetes homogenes Polynom vom Grad 3. Falls $\text{char}(k) = 0$ oder $\text{char}(k) > 3$, so kann diese Gleichung soweit umgeformt und vereinfacht werden, dass \bar{E} aus dem Punkt ∞ und einem Teil $E \subseteq k^2$ besteht, der durch eine Polynom der Form

$$F(X, Y) = Y^2 - X^3 - aX - b \in [X, Y]$$

mit $4 \cdot a^3 + 27 \cdot b^2 \neq 0$ gegeben ist. In Fall $\text{char}(k) = 2$ ist das nicht möglich (hier fehlt die Möglichkeit der quadratischen Ergänzung). Hier kann die elliptische Kurve entweder durch den Punkt ∞ und einen Teil $E \subseteq k^2$, der durch ein Polynom

$$F(X, Y) = Y^2 + XY + X^3 + aX + b \in [X, Y]$$

mit $b \neq 0$ gegeben wird, beschrieben werden, oder durch den Punkt ∞ und einen Teil $E \subseteq k^2$, der durch ein Polynom

$$F(X, Y) = Y^2 + cY + X^3 + aX + b \in [X, Y]$$

mit $c \neq 0$ definiert ist. Der zweite Fall ist für die Kryptologie nicht relevant und wird daher hier nicht weiter betrachtet werden.

Ebenso nicht relevant in der Kryptographie sind elliptische Kurven über Körpern der Charakteristik 3.

Wir betrachten also jetzt einen Körper $k = \mathbb{F}_q$ mit $q = 2^l$ und eine elliptische Kurve über k , die beschrieben wird durch ein Polynom der Form

$$F(X, Y) = Y^2 + XY + X^3 + aX^2 + b \in \mathbb{F}_q[X, Y]$$

mit $b \neq 0$. Zur Bestimmung der Punkte $(r, s) \in E$ sind alle Lösungen $(x, y) \in \mathbb{F}_8^2$ der Gleichung

$$y^2 + xy + x^3 + ax^2 + b = 0 \quad (13.1)$$

zu finden, und dazu gehen wir vor wie folgt:

- Setze $f(x) = x^3 + ax^2 + b$.
- Fixiere die x -Komponente $x = r$, setzte r in Gleichung (13.1) ein und erhalte

$$y^2 + ry + f(r) = 0 \quad (13.2)$$

- Bestimme die Lösungen von Gleichung (13.2) wie in Anhang D skizziert.

Beispiel 13.1. Wir betrachten den Körper \mathbb{F}_8 , gegeben durch die Relation $\alpha^3 = \alpha + 1$ und die elliptische Kurve über \mathbb{F}_8 , gegeben durch

$$F(X, Y) = Y^2 + XY + X^3 + X^2 + \alpha \in \mathbb{F}_8[X, Y]$$

Für die Rechnungen ist es sinnvoll, die Elemente von $\mathbb{F}_8 \setminus \{0\}$ auch als Potenzen von α beschreiben zu können. Dabei gilt

$$\begin{aligned} \alpha^1 &= \alpha & \alpha^4 &= \alpha^2 + \alpha & \alpha^6 &= \alpha^2 + 1 \\ \alpha^2 &= \alpha^2 & \alpha^5 &= \alpha^2 + \alpha + 1 & \alpha^7 &= 1 \\ \alpha^3 &= \alpha + 1 & & & & \end{aligned}$$

Wir setzen ferner $f(x) = x^3 + x^2 + \alpha$, und haben für jedes $r \in \mathbb{F}_8$, zu untersuchen, ob die Gleichung

$$y^2 + ry + f(r) = 0 \quad (13.3)$$

in y Lösungen hat, und wenn ja, welche. Dazu benutzen wir die Techniken aus Anhang D, speziell die Spur D.1, die in diesem Fall definiert ist als

$$\text{Tr}(a) = \sum_{j=0}^{3-1} a^{2^j} = a + a^2 + a^4$$

Da $l = 3$ ungerade ist, kommt auch die Halbspur D.2 zum Einsatz, die hier beschrieben wird durch

$$\text{HTr}(a) = \sum_{j=0}^{\frac{3-1}{2}} a^{2^{2j}} = a + a^4$$

Wir gehen nun die Elemente von \mathbb{F}_8 einzeln durch:

1. $r = 0$. In diesem Fall wird Gleichung (13.3) zu

$$y^2 + \alpha = 0$$

Nach Folgerung D.3 ist

$$s = \alpha^{\frac{8}{2}} = \alpha^4 = \alpha^2 + \alpha$$

die eindeutige Lösung dieser Gleichung, und daher bekommen wir den Punkt

$$P_1 = (0, \alpha^2 + \alpha)$$

auf der elliptischen Kurve.

2. $r = \alpha$. In diesem Fall ist

$$f(\alpha) = \alpha^3 + \alpha^2 + \alpha = \alpha + 1 + \alpha^2 + \alpha = \alpha^2 + 1$$

also wird Gleichung (13.3) zu

$$y^2 + \alpha \cdot y + \alpha^2 + 1 = 0$$

Gemäß Hilfssatz D.6 betrachten wir hierfür zunächst die Gleichung

$$y^2 + y + \frac{\alpha^2 + 1}{\alpha^2} = 0$$

also (wegen $\frac{\alpha^2+1}{\alpha^2} = \frac{\alpha^6}{\alpha^2} = \alpha^4$)

$$y^2 + y + \alpha^4 = 0$$

Hierfür gilt

$$\text{Tr}(\alpha^4) = \alpha^4 + \alpha^8 + \alpha^{16} = \alpha^2 + \alpha + \alpha + \alpha^2 = 0$$

und damit hat diese Gleichung nach Regel D.9 Lösungen, und eine davon ist

$$n_1 = \text{HTr}(\alpha^4) = \alpha^4 + \alpha^{16} = \alpha^2 + \alpha + \alpha^2 = \alpha$$

und die zweite ist $n_2 = \alpha + 1$. Damit hat die Ausgangsgleichung nach Regel D.6 die beiden Lösungen

$$s_1 = \alpha \cdot n_1 = \alpha^2, \quad s_2 = \alpha \cdot n_2 = \alpha^2 + \alpha$$

und wir erhalten die beiden Punkte

$$P_2 = (\alpha, \alpha^2), \quad P_3 = (\alpha, \alpha^2 + \alpha)$$

auf E .

3. $r = \alpha^2$. In diesem Fall ist

$$f(\alpha^2) = \alpha^6 + \alpha^4 + \alpha = \alpha^2 + 1 + \alpha^2 + \alpha + \alpha = 1$$

also wird Gleichung (13.3) zu

$$y^2 + \alpha^2 \cdot y + 1 = 0$$

Die gemäß Hilfssatz D.6 zugehörige reduzierte Gleichung ist

$$y^2 + y + \frac{1}{\alpha^4} = 0$$

also (wegen $\frac{1}{\alpha^4} = \frac{\alpha^7}{\alpha^4} = \alpha^3$)

$$y^2 + y + \alpha^3 = 0$$

Hierfür gilt

$$\text{Tr}(\alpha^3) = \alpha^3 + \alpha^6 + \alpha^{12} = \alpha + 1 + \alpha^2 + 1 + \alpha^2 + \alpha + 1 = 1$$

und damit hat diese Gleichung nach Regel D.9 keine Lösungen. Es gibt also keinen Punkt auf E mit x -Komponenten α^2 .

4. $r = \alpha^3 = \alpha + 1$. In diesem Fall ist

$$f(\alpha^3) = \alpha^9 + \alpha^6 + \alpha = \alpha^2 + \alpha^2 + 1 + \alpha = \alpha + 1$$

also wird Gleichung (13.3) zu

$$y^2 + (\alpha + 1) \cdot y + \alpha + 1 = 0$$

Die gemäß Hilfssatz D.6 zugehörige reduzierte Gleichung ist

$$y^2 + y + \frac{\alpha^3}{\alpha^6} = 0$$

also zu

$$y^2 + y + \alpha^4 = 0$$

Hierfür gilt, wie wir schon im Fall $r = \alpha$ gesehen haben,

$$\text{Tr}(\alpha^4) = 0$$

und damit hat diese Gleichung nach Regel D.9 Lösungen, und eine davon ist

$$n_1 = \text{HTr}(\alpha^4) = \alpha$$

und die zweite ist $n_2 = \alpha + 1$. Damit hat die Ausgangsgleichung nach Regel D.6 die beiden Lösungen

$$s_1 = \alpha^3 \cdot n_1 = \alpha^2 + \alpha, \quad s_2 = \alpha^3 \cdot n_2 = \alpha^2 + 1$$

und wir erhalten die beiden Punkte

$$P_4 = (\alpha + 1, \alpha^2 + \alpha), \quad P_5 = (\alpha + 1, \alpha^2 + 1)$$

auf E .

5. $r = \alpha^4 = \alpha^2 + \alpha$. In diesem Fall ist

$$f(\alpha^4) = \alpha^{12} + \alpha^8 + \alpha = \alpha^2 + \alpha + 1 + \alpha + \alpha = \alpha^2 + \alpha + 1$$

also wird Gleichung (13.3) zu

$$y^2 + (\alpha + 1) \cdot y + \alpha^2 + \alpha + 1 = 0$$

Die gemäß Hilfssatz D.6 zugehörige reduzierte Gleichung ist

$$y^2 + y + \frac{\alpha^5}{\alpha^8} = 0$$

also zu

$$y^2 + y + \alpha^4 = 0$$

Hierfür gilt, wie wir schon im Fall $r = \alpha$ gesehen haben,

$$\text{Tr}(\alpha^4) = 0$$

und damit hat diese Gleichung nach Regel D.9 Lösungen, und eine davon ist

$$n_1 = \text{HTr}(\alpha^4) = \alpha$$

und die zweite ist $n_2 = \alpha + 1$. Damit hat die Ausgangsgleichung nach Regel D.6 die beiden Lösungen

$$s_1 = \alpha^4 \cdot n_1 = \alpha^2 + \alpha + 1, \quad s_2 = \alpha^4 \cdot n_2 = 1$$

und wir erhalten die beiden Punkte

$$P_6 = (\alpha^2 + \alpha, \alpha^2 + \alpha + 1), \quad P_7 = (\alpha^2 + \alpha, 1)$$

auf E .

6. $r = \alpha^5 = \alpha^2 + \alpha + 1$. In diesem Fall ist

$$f(\alpha^5) = \alpha^{15} + \alpha^{10} + \alpha = \alpha + \alpha + 1 + \alpha = \alpha + 1$$

also wird Gleichung (13.3) zu

$$y^2 + (\alpha^2 + \alpha + 1) \cdot y + \alpha + 1 = 0$$

Die gemäß Hilfssatz D.6 zugehörige reduzierte Gleichung ist

$$y^2 + y + \frac{\alpha^3}{\alpha^{10}} = 0$$

also (wegen $\frac{\alpha^3}{\alpha^{10}} = \frac{\alpha^3}{\alpha^3} = 1$)

$$y^2 + y + 1 = 0$$

Hierfür gilt

$$\text{Tr}(1) = 1 + 1^2 + 1^4 = 1$$

und damit hat diese Gleichung nach Regel D.9 keine Lösungen. Es gibt also keinen Punkt auf E mit x -Komponenten α^5 .

7. $r = \alpha^6 = \alpha^2 + 1$. In diesem Fall ist

$$f(\alpha^6) = \alpha^{18} + \alpha^{12} + \alpha = \alpha^2 + \alpha + \alpha^2 + \alpha + 1 + \alpha = \alpha + 1$$

also wird Gleichung (13.3) zu

$$y^2 + (\alpha^2 + 1) \cdot y + \alpha + 1 = 0$$

Die gemäß Hilfssatz D.6 zugehörige reduzierte Gleichung ist

$$y^2 + y + \frac{\alpha^3}{\alpha^{12}} = 0$$

also (wegen $\frac{\alpha^3}{\alpha^{10}} = \frac{\alpha^{10}}{\alpha^5} = \alpha^2 + \alpha + 1$)

$$y^2 + y + \alpha^2 + \alpha + 1 = 0$$

Hierfür gilt

$$\text{Tr}(\alpha^5) = \alpha^5 + \alpha^{10} + \alpha^{20} = \alpha^2 + \alpha + 1 + \alpha + 1 + \alpha^2 + 1 = 1$$

und damit hat diese Gleichung nach Regel D.9 keine Lösungen. Es gibt also keinen Punkt auf E mit x -Komponenten α^6 .

8. $r = \alpha^7 = 1$. In diesem Fall ist

$$f(1) = 1 + 1 + \alpha = \alpha$$

also wird Gleichung (13.3) zu

$$y^2 + y + \alpha = 0$$

und liegt schon in reduzierter Form vor. Hierfür gilt

$$\text{Tr}(\alpha) = \alpha + \alpha^2 + \alpha^4 = 0$$

und damit hat diese Gleichung nach Regel D.9 Lösungen, und eine davon ist

$$n_1 = \text{HTr}(\alpha) = \alpha + \alpha^4 = \alpha^2$$

und die zweite ist $n_2 = \alpha^2 + 1$. Wir erhalten also noch die beiden Punkte

$$P_8 = (1, \alpha^2), \quad P_9 = (1, \alpha^2 + 1)$$

auf E .

Insgesamt haben wir also 9 Punkte auf E gefunden.

Bemerkung 13.2. Die Berechnung der Punkte auf einer elliptischen Kurve in der Charakteristik 2 sieht relativ aufwendig und komplex aus, kann aber ziemlich gut implementiert werden. Für Körper mit vielen Elementen wird die Bestimmung aller Punkte dennoch aufwendig.

Bemerkung 13.3. Im Fall $\text{char}(k) = p > 3$ oder $\text{char}(k) = 0$ sind elliptische Kurven (in ihrer Normalform) immer achsensymmetrisch zur x -Achse, dh. mit $P = (r, s)$ ist auch $Q = (r, -s)$ auf E . Diese Notation ergibt im Fall der Charakteristik 2 keinen Sinn mehr, denn $-s = s$. Allerdings gilt hier:

Ist $P = (r, s)$ ein Punkt auf E , so ist aus $Q = (r, s + r)$ ein Punkt auf E , und P und Q sind die einzigen beiden Punkte auf E mit x -Komponente r .

Das kann direkt durch Einsetzen verifiziert werden, denn es gilt

$$\begin{aligned} F(r, s + r) &= (s + r)^2 + r \cdot (s + r) + r^3 + a \cdot r^2 + b \\ &= s^2 + r^2 + r \cdot s + r^2 + r^3 + a \cdot r^2 + b \\ &= s^2 + r \cdot s + r^3 + a \cdot r^2 + b \\ &= F(r, s) \\ &= 0 \end{aligned}$$

Falls $r = 0$, so ist $(0, s) = (0, s + 0)$ die einzige Lösung der Gleichung

$$y^2 + b = 0$$

denn nach Anhang D sind Quadratwurzeln in der Charakteristik 2 eindeutig. Falls $r \neq 0$, so sind s und $r + s$ zwei (voneinander verschiedene) Lösungen der quadratischen Gleichung

$$y^2 + r \cdot y + r^3 + a \cdot r^2 + b = 0$$

Daher kann es keine weitere Lösung dieser Gleichung geben, also auch keinen weiteren Punkt auf E mit x -Komponente r .

13.2. Addition auf elliptischen Kurven in der Charakteristik 2

Auch in diesem Fall gibt es eine Addition $+$ auf \overline{E} , die $(\overline{E}, +)$ zu einer abelschen Gruppe mit neutralem Element ∞ macht. Diese Addition beruht (wie schon in Abschnitt 12) ebenfalls auf der Tatsache, dass jede Gerade L die elliptische Kurve \overline{E} in genau drei Punkten schneidet (wobei Tangentenpunkte doppelt zählen) und ist so konstruiert, dass für die drei Schnittpunkte P_1 , P_2 und P_3 gilt

$$P_1 + P_2 + P_3 = \infty$$

Wie im Abschnitt 12 kann auch hier der dritte Schnittpunkt P_3 aus den beiden Punkten P_1 und P_2 explizit konstruiert werden.

Regel 13.1. *Wir betrachten eine elliptische Kurve E/\mathbb{F}_q mit $q = 2^l$, gegeben durch*

$$F(X, Y) = Y^2 + XY + X^3 + a \cdot X^2 + b \in \mathbb{F}_q[X, Y]$$

und zwei Punkte $P_1 = (r_1, s_1)$ und $P_2 = (r_2, s_2)$ auf E .

1. Ist $r_1 = r_2 = 0$, so ist $s_1 = s_2$, und die Gerade $L : x = 0$ schneidet die Kurve E in P_1 (doppelt) und in keinem weiteren Punkt.
2. Ist $r_1 = r_2 \neq 0$, so ist $s_2 = s_1 + r_1$ und die Gerade $L : x = r_1$ schneidet E in den beiden Punkten P_1 und P_2 (jeweils einfach) und in keinem weiteren Punkt.
3. Ist $r_1 \neq r_2$, ist $m = \frac{s_2 + s_1}{r_2 + r_1}$ die Steigung der Geraden L durch P_1 und P_2 , und ist

$$m \neq r_1 + \frac{s_1}{r_1}, \quad m \neq r_2 + \frac{s_2}{r_2}$$

so schneidet L die Kurve E in genau noch einem weiteren Punkt $Q = (u, v)$, wobei

$$u = m^2 + m + a, \quad v = m \cdot (u + r_1) + s_1$$

4. Ist $r_1 \neq r_2$, ist $m = \frac{s_2+s_1}{r_2+r_1}$ die Steigung der Geraden L durch P_1 und P_2 , und ist $m \neq r_1 + \frac{s_1}{r_1}$ so schneidet L die Kurve E im Punkt P_1 (doppelt), im Punkt P_2 (einfach) und in keinem weiteren Punkt.
5. Ist $r_1 \neq r_2$, ist $m = \frac{s_2+s_1}{r_2+r_1}$ die Steigung der Geraden L durch P_1 und P_2 , und ist $m \neq r_2 + \frac{s_2}{r_2}$ so schneidet L die Kurve E im Punkt P_1 (einfach), im Punkt P_2 (doppelt) und in keinem weiteren Punkt.

Diese Aussagen können (wie im Abschnitt 11) relativ elementar nachgerechnet werden (lediglich der Nachweis, dass manche Schnittpunkte doppelte Schnittpunkte sind, bereitet etwas Mühe). Wir wollen hier nur einige ausgewählte Punkte ansprechen:

1. Ist (r, s) ein Punkt auf E , so ist auch $(r, r + s)$ ein Punkt auf E , wie wir schon in Bemerkung 13.3 nachgerechnet haben.
2. Sind die beiden Punkte $P_1 = (r, s)$ und $P_2 = (r, r + s)$ mit $r \neq 0$ auf E , so schneidet die Gerade L durch P_1 und P_2 die Kurve E nicht mehr, wie wir auch in Bemerkung 13.3 überprüft haben.
3. Jede Zahl in \mathbb{F}_q ist ein Quadrat. Daher gibt es ein eindeutig bestimmtes $\beta \in \mathbb{F}_q$ mit $b = \beta^2$. Ferner liegt der Punkt $P = (0, \beta)$ auf E , denn

$$\beta^2 + 0 \cdot \beta + 0^3 + a \cdot 0^2 + b = b + b = 0$$

und die Gerade $x = 0$ schneidet E nur im Punkt P (tangential) und sonst nicht mehr (siehe auch hierzu Bemerkung 13.3).

Der Punkt ∞ auf \overline{E} erfüllt nun die Bedingung, dass alle Geraden der Form $L : x = r$, die E nur in zwei Punkten (einfach) oder (im Fall $r = 0$) in einem Punkt doppelt schneiden, auch noch den Punkt ∞ mit \overline{E} gemeinsam haben. Damit wird die Addition auf \overline{E} wieder so konstruiert, dass ∞ das neutrale Element für diese Operation ist, und dass für drei Punkte P_1 , P_2 und P_3 auf E , die auf einer Geraden liegen,

$$P_1 + P_2 + P_3 = \infty$$

also $P_1 + P_2 = -P_3$ gilt.

Da der dritte Punkt auf \overline{E} und der Geraden durch zwei Punkte $P_1 = (r, s)$ und $P_2 = (r, s + r)$ auf E nach Regel 13.1 der Punkt ∞ ist, bedeutet das

$$(r, s) + (r, s + r) + \infty = \infty$$

also, da ∞ das neutrale Element ist, $(r, s) + (r, s + r) = \infty$ und daher

$$(r, s + r) = -(r, s)$$

Damit ergibt sich durch Rechnungen sehr ähnlich zu denen, die wir für den Nachweis der Sätze 11.1 und 11.2 benutzt haben:

Regel 13.2. Für die Addition auf \overline{E} gilt:

1. Ist $P = (r, s)$, so ist $-P = (r, s + r)$.
2. Es ist $\infty + \infty = \infty$.
3. Für $P = (r, s) \in E$ ist $P + \infty = \infty + P = P$.
4. Sind $P_1 = (r_1, s_1), P_2 = (r_2, s_2) \in E$ mit $P_1 = -P_2$, so ist $P_1 + P_2 = \infty$.
5. Sind $P_1 = (r_1, s_1), P_2 = (r_2, s_2) \in E$ mit $r_1 \neq r_2$ (also auch $P_1 \neq P_2$), so ist

$$P_1 + P_2 = R = (u, v)$$

wobei mit $m = \frac{s_2+s_1}{r_2+r_1}$ gilt:

$$u = m^2 + m + a + r_1 + r_2, \quad v = m \cdot (r_1 + u) + u + s_1$$

6. Sind $P_1 = (r_1, s_1), P_2 = (r_2, s_2) \in E$ mit $r_1 = r_2$ aber $P_1 \neq -P_2$, so ist notwendig $P_1 = P_2$ und es gilt

$$2 \cdot P_1 = P_1 + P_2 = R = (u, v)$$

wobei mit $m = r_1 + \frac{s_1}{r_1}$ gilt:

$$u = m^2 + m + a, \quad v = m \cdot (r_1 + u) + u + s_1$$

7. Ist $P = (0, s) \in E$, so ist

$$2 \cdot P = P + P = \infty$$

Beispiel 13.2. Wir betrachten den Körper $k = \mathbb{F}_8$ mit acht Elementen, gegeben durch die Relation $\alpha^3 = \alpha + 1$ und die elliptische Kurve \overline{E} über \mathbb{F}_8 , gegeben durch

$$F(X, Y) = Y^2 + XY + X^3 + X^2 + \alpha \in \mathbb{F}_8[X, Y]$$

(vergleiche Beispiel 13.1). Diese Kurve besteht aus den folgenden Punkten

$$\begin{aligned} \overline{E} = & \{(0, \alpha^2 + \alpha), (\alpha, \alpha^2), (\alpha, \alpha^2 + \alpha), (\alpha + 1, \alpha^2 + \alpha), (\alpha + 1, \alpha^2 + 1), \\ & (\alpha^2 + \alpha, \alpha^2 + \alpha + 1), (\alpha^2 + \alpha, 1), (1, \alpha^2), (1, \alpha^2 + 1), \infty\} \end{aligned}$$

Um mit dieser Kurve besser zu rechnen, nutzen wir wieder aus, dass

$$\begin{aligned}\alpha^1 &= \alpha, & \alpha^2 &= \alpha^2, & \alpha^3 &= \alpha + 1, & \alpha^4 &= \alpha^2 + \alpha, \\ \alpha^5 &= \alpha^2 + \alpha + 1, & \alpha^6 &= \alpha^2 + 1, & \alpha^7 &= 1\end{aligned}$$

Damit erhalten wir:

Für $P = (\alpha^2 + \alpha, 1)$ gilt $-P = (\alpha^2 + \alpha, \alpha^2 + \alpha + 1)$.

Für $P = (0, \alpha^2 + \alpha)$ gilt $2 \cdot P = \infty$ nach Regel 13.2, 7.

Für $P_1 = (\alpha, \alpha^2 + \alpha)$ und $P_2 = (\alpha, \alpha^2)$ gilt $P_1 + P_2 = \infty$ nach Regel 13.2, 4.

Für $P_1 = (\alpha, \alpha^2)$ und $P_2 = (1, \alpha^2 + 1)$ ist

$$m = \frac{\alpha^2 + 1 + \alpha^2}{\alpha + 1} = \frac{1}{\alpha^3} = \alpha^4 = \alpha^2 + \alpha$$

und damit ist nach Regel 13.2, 5.

$$\begin{aligned}u &= (\alpha^4)^2 + \alpha^4 + 1 + \alpha + 1 &= \alpha^2 + \alpha \\ v &= \alpha^4 \cdot (\alpha^2 + \alpha + \alpha) + \alpha^2 + \alpha + \alpha^2 &= \alpha^2 + \alpha + 1\end{aligned}$$

also

$$P_1 + P_2 = (\alpha^2 + \alpha, \alpha^2 + \alpha + 1)$$

Für $P = (\alpha + 1, \alpha^2 + \alpha)$ hat die Tangente an E in P die Steigung

$$m = \alpha + 1 + \frac{\alpha^2 + \alpha}{\alpha + 1} = \alpha + 1 + \alpha = 1$$

und damit ist nach Regel 13.2, 6.

$$\begin{aligned}u &= 1^2 + 1 + 1 &= 1 \\ v &= 1 \cdot (1 + \alpha + 1) + 1 + \alpha^2 + \alpha &= \alpha^2 + 1\end{aligned}$$

also

$$2 \cdot P = (1, \alpha^2 + 1)$$

Beispiel 13.3. Wir betrachten den Körper $k = \mathbb{F}_8$ mit acht Elementen, gegeben durch die Relation $\alpha^3 = \alpha + 1$ und die elliptische Kurve \overline{E} über \mathbb{F}_8 , gegeben durch

$$F(X, Y) = Y^2 + XY + X^3 + \alpha X^2 + \alpha^2 \in \mathbb{F}_8[X, Y]$$

Diese Kurve besteht aus den folgenden Punkten

$$\begin{aligned}\overline{E} = \{ &(0, \alpha), (\alpha + 1, \alpha^2 + \alpha + 1), (\alpha + 1, \alpha^2), (\alpha^2 + \alpha, 1), \\ &(\alpha^2 + \alpha, \alpha^2 + \alpha + 1), (\alpha^2 + \alpha + 1, 0), (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1), \infty \}\end{aligned}$$

Um mit dieser Kurve besser zu rechnen, nutzen wir wieder aus, dass

$$\begin{aligned}\alpha^1 &= \alpha, & \alpha^2 &= \alpha^2, & \alpha^3 &= \alpha + 1, & \alpha^4 &= \alpha^2 + \alpha, \\ \alpha^5 &= \alpha^2 + \alpha + 1, & \alpha^6 &= \alpha^2 + 1, & \alpha^7 &= 1\end{aligned}$$

Wir wollen damit beispielhaft überprüfen, dass der Punkt $(\alpha^2 + \alpha, 1)$ tatsächlich auf der elliptischen Kurve liegt:

$$\begin{aligned}F(\alpha^2 + \alpha, 1) &= 1^2 + (\alpha^2 + \alpha) \cdot 1 + (\alpha^2 + \alpha)^3 + \alpha \cdot (\alpha^2 + \alpha)^2 + \alpha^2 \\ &= 1 + \alpha^2 + \alpha + (\alpha^4)^3 + \alpha \cdot (\alpha^4)^2 + \alpha^2 \\ &= 1 + \alpha^2 + \alpha + \alpha^{12} + \alpha \cdot \alpha^8 + \alpha^2 \\ &= 1 + \alpha^2 + \alpha + \alpha^5 + \alpha^2 + \alpha^2 \\ &= 1 + \alpha^2 + \alpha + \alpha^5 \\ &= 1 + \alpha^2 + \alpha + \alpha^2 + \alpha + 1 \\ &= 0\end{aligned}$$

Als nächstes bestimmen wir den dritten Schnittpunkt P_3 der Geraden L durch $P_1 = (\alpha + 1, \alpha^2)$ und $P_2 = (\alpha^2 + \alpha, 1)$ mit E . Dazu stellen wir zunächst die Geradengleichung von L auf:

Die Gerade L hat die Steigung

$$m = \frac{1 - \alpha^2}{\alpha^2 + \alpha - \alpha - 1} = \frac{\alpha^2 + 1}{\alpha^2 + 1} = 1$$

und ist damit gegeben durch

$$y = 1 \cdot (x - \alpha - 1) + \alpha^2 = x + \alpha^2 + \alpha + 1 = x + \alpha^5$$

Setzen wir das in die Gleichung für E ein, so erhalten wir

$$(x + \alpha^5)^2 + x \cdot (x + \alpha^5) + x^3 + \alpha \cdot x^2 + \alpha^2 = 0$$

also

$$x^2 + \alpha^3 + x^2 + \alpha^5 \cdot x + x^3 + \alpha \cdot x^2 + \alpha^2 = 0$$

und damit

$$x^3 + \alpha \cdot x^2 + \alpha^5 \cdot x + \alpha^5 = 0 \tag{13.4}$$

Wir wissen schon, dass $r_1 = \alpha + 1$ und $r_2 = \alpha^2 + \alpha$ Lösungen dieser Gleichung sind, und damit ist $(X - \alpha - 1) \cdot (X - \alpha^2 - \alpha)$ ein Teiler von

$$f(X) = X^3 + \alpha \cdot X^2 + \alpha^5 \cdot X + \alpha^5$$

Es ist

$$\begin{aligned}
 (X - \alpha - 1) \cdot (X - \alpha^2 - \alpha) &= (X + \alpha^3) \cdot (X + \alpha^4) \\
 &= X^2 + (\alpha^3 + \alpha^4) \cdot X + \alpha^7 \\
 &= X^2 + \alpha^6 \cdot X + 1
 \end{aligned}$$

und Polynomdivision liefert

$$(X^3 + \alpha \cdot X^2 + \alpha^5 \cdot X + \alpha^5) \div (X^2 + \alpha^6 \cdot X + 1) = X + \alpha^5$$

Damit ist also $r_3 = \alpha^5 = \alpha^2 + \alpha + 1$ die dritte Lösung von Gleichung 13.4. Die y -Komponente des zugehörigen Punktes ist gegeben durch

$$s_3 = m \cdot (r_3 + r_1) + s_1 = \alpha^2 + \alpha + 1 + \alpha + 1 + \alpha^2 = 0$$

dh. $P_3 = (\alpha^2 + \alpha + 1, 0)$. Damit ist also

$$P_1 + P_2 + P_3 = \infty$$

folglich

$$P_1 + P_2 = -P_3 = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1)$$

Wenn wir also, wie üblich, $r_2 \cdot \alpha^2 + r_1 \cdot \alpha + r_0$ mit $(r_2, r_1, r_0) \in \mathbb{F}_2^3$ identifizieren, so bedeutet das

$$((0, 1, 1), (1, 0, 0)) + ((1, 1, 0), (0, 0, 1)) = ((1, 1, 1), (1, 1, 1))$$

Wenden wir direkt die Regel 13.2 an, so erhalten wir mit

$$m = \frac{s_1 + s_2}{r_1 + r_2} = \frac{\alpha^2 + 1}{\alpha^2 + 1} = 1$$

für die beiden Komponenten (u, v) von $P_1 + P_2$:

$$u = m^2 + m + a + r_1 + r_2 = 1^2 + 1 + \alpha + \alpha + 1 + \alpha^2 + \alpha = \alpha^2 + \alpha + 1$$

und

$$v = m \cdot (r_1 + u) + u + s_1 = 1 \cdot (\alpha + 1 + \alpha^2 + \alpha + 1) + \alpha^2 + \alpha + 1 + \alpha^2 = \alpha^2 + \alpha + 1$$

also tatsächlich dasselbe Resultat.

Als nächstes berechnen wir für $P = (\alpha + 1, \alpha^2 + \alpha + 1)$ den Punkt $4 \cdot P$. Dazu ermitteln wir zunächst $Q = P + P$ mit Koordinaten $Q = (u, v)$. Hier greift Regel 13.2, 6.:

$$m = r_1 + \frac{s_1}{r_1} = \alpha + 1 + \frac{\alpha^2 + \alpha + 1}{\alpha + 1} = \alpha + 1 + \frac{\alpha^5}{\alpha^3} = \alpha + 1 + \alpha^2 = \alpha^5$$

also

$$u = m^2 + m + a = \alpha^{10} + \alpha^5 + \alpha = \alpha + 1 + \alpha^2 + \alpha + 1 + \alpha = \alpha^2 + \alpha$$

und

$$v = m \cdot (r_1 + u) + u + s_1 = \alpha^5 \cdot (\alpha + 1 + \alpha^2 + \alpha) + \alpha^2 + \alpha + \alpha^2 + \alpha + 1 = \alpha^2 + \alpha + 1$$

also $Q = (\alpha^2 + \alpha, \alpha^2 + \alpha + 1)$. Als nächstes berechnen wir $R = 2 \cdot Q$ mit Koordinaten $R = (t, w)$. Hier greift Regel 13.2, 6.:

$$m = u + \frac{v}{u} = \alpha^2 + \alpha + \frac{\alpha^5}{\alpha^4} = \alpha^2 + \alpha + \alpha = \alpha^2$$

also

$$t = m^2 + m + a = \alpha^4 + \alpha^2 + \alpha = \alpha^2 + \alpha + \alpha^2 + \alpha = 0$$

und

$$w = m \cdot (u + t) + t + u = \alpha^2 \cdot (\alpha^2 + \alpha) + \alpha^2 + \alpha + 1 = \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1 = \alpha$$

Also gilt $4 \cdot P = 2 \cdot Q = (0, \alpha)$, bzw.

$$4 \cdot ((0, 1, 1), (1, 1, 1)) = ((0, 0, 0), (0, 1, 0))$$

Beispiel 13.4. Wir betrachten wieder den Körper $k = \mathbb{F}_8$, gegeben durch die Relation $\alpha^3 = \alpha + 1$ und die elliptische Kurve \overline{E} über \mathbb{F}_8 , gegeben durch

$$F(X, Y) = Y^2 + XY + X^3 + \alpha^2 X^2 + \alpha \in \mathbb{F}_8[X, Y]$$

Diese Kurve besteht aus den folgenden Punkten

$$\begin{aligned} \overline{E} = & \{(0, \alpha^2 + \alpha), (\alpha^2, \alpha + 1), (\alpha^2, \alpha^2 + \alpha + 1), (\alpha^2 + 1, \alpha), \\ & (\alpha^2 + 1, \alpha^2 + \alpha + 1), (\alpha^2 + \alpha + 1, \alpha^2), (\alpha^2 + \alpha + 1, \alpha + 1), \infty\} \end{aligned}$$

Für $P_1 = (\alpha^2, \alpha + 1)$ und $P_2 = (\alpha^2 + 1, \alpha^2 + \alpha + 1)$ gilt

$$m = \frac{s_1 + s_2}{r_1 + r_2} = \frac{\alpha^2}{1} = \alpha^2$$

und damit gilt für $R = (u, v) = P_1 + P_2$:

$$u = \alpha^2 + \alpha + 1, \quad v = \alpha + 1$$

also

$$((1, 0, 0), (0, 1, 1)) + ((1, 0, 1), (1, 1, 1)) = ((1, 1, 1), (0, 1, 1))$$

Für $P = (\alpha^2 + \alpha + 1, \alpha^2)$ ($= P_1 = P_2$) gilt

$$m = r_1 + \frac{s_1}{r_1} = \alpha^2 + \alpha + 1 + \frac{\alpha^2}{\alpha^5} = 1$$

und damit gilt für $R = (u, v) = 2 \cdot P$:

$$u = \alpha^2, \quad v = \alpha + 1$$

also

$$2 \cdot ((1, 1, 1), (1, 0, 0)) = ((1, 0, 0), (0, 1, 1))$$

Beispiel 13.5. Wir betrachten den Körper $k = \mathbb{F}_{16}$, gegeben durch die Relation $\alpha^4 = \alpha + 1$ und die elliptische Kurve \overline{E} über \mathbb{F}_{16} , gegeben durch

$$F(X, Y) = Y^2 + XY + X^3 + \alpha^2 X^2 + \alpha^3 \in \mathbb{F}_{16}[X, Y]$$

Dann sind die Punkte $P_1 = (\alpha^3 + \alpha^2, \alpha^3 + \alpha)$ und $P_2 = (\alpha^2 + \alpha, \alpha^3 + \alpha + 1)$ auf \overline{E} und

$$P_1 + P_2 = (\alpha^3, \alpha^3 + \alpha)$$

Ferner ist $P = (\alpha^3 + \alpha^2 + \alpha, \alpha^2 + \alpha + 1)$ auf \overline{E} und

$$12 \cdot P = (\alpha, 1)$$

14. Elliptische Kryptosysteme

Elliptische Kurven \overline{E} über endlichen Körpern können genutzt werden, um (nach aktuellem Stand der Technik) sehr sichere Verschlüsselungsalgorithmen zu implementieren. Die Grundidee ist sehr ähnlich zu „klassischen“ asymmetrischen kryptographischen Verfahren wie ElGamal oder Diffie–Hellman. Elliptische Kurven \overline{E} über einem endlichen Körper \mathbb{F}_q der Charakteristik $\neq 3$ bilden nicht immer zyklische Gruppen, im allgemeinen gilt (im Fall $p > 3$)

$$\overline{E} = \mathbb{Z}_n \times \mathbb{Z}_{n \cdot t}$$

(vergleiche Kapitel 12 über elliptische Kurven), aber \overline{E} enthält zyklische Untergruppen, häufig sogar sehr große zyklische Gruppen.

Zum Aufbau eines Kryptosystems wähle eine zyklische Untergruppe $U \subseteq \overline{E}$, deren Ordnung eine Primzahl r ist, und ein Element $g \in U$, $g \neq \infty$.

Bemerkung 14.1. Das Element hat die Ordnung r und $U = \langle g \rangle$, dh. U wird von g erzeugt.

Nach dem Satz von Fermat ist nämlich $\text{ord}(g)$ ein Teiler der Gruppenordnung $|U|$, also ein Teiler von r . Da r eine Primzahl ist und $g \neq \infty$, dh. $\text{ord}(g) > 1$, ist notwendig $\text{ord}(g) = r$.

Das diskrete Logarithmusproblem ECDLP für elliptische Kurven:

Zu einem beliebigen Punkt $h \in U$ bestimme ein $t \in \mathbb{N}$ mit $h = t \cdot g$.

Das ECDLP gilt als schwer. Gegenwärtig ist kein algorithmischer Ansatz bekannt, der das gesuchte t allgemein schneller bestimmt als ein brute-force–Ansatz, also als vollständiges Ausprobieren. Dadurch unterscheidet sich das ECDLP vom DLP.

14.1. Vielfachenbildung durch iteriertes Verdoppeln

Für einen beliebigen Punkt $g \in \overline{E}$ ist die Berechnung von $h = t \cdot g$ aus g und t vergleichsweise einfach:

1. Schreibe $t = \sum_{i=0}^l \tau_i \cdot 2^i$ mit $\tau_i \in \{0, 1\}$ und mit $\tau_l = 1$ und setze $\tau = (\tau_l, \tau_{l-1}, \dots, \tau_0)$.
2. Setze $h_l = g$.
3. für $i = l - 1, \dots, 0$ (abwärts laufend) setze

$$\widetilde{h}_i = h_{i+1} + h_{i+1}$$

(Punktverdoppelung auf \overline{E} gemäß den Gruppengesetzen von \overline{E}).

Falls $\tau_i = 1$ setze

$$h_i = \widetilde{h}_i + g$$

andernfalls setze

$$h_i = \widetilde{h}_i$$

4. Setze $h = h_0$.

Hilfssatz 14.1. $h = t \cdot g$.

Beweis: Für $i = 0, \dots, l$ setze

$$t_i = \sum_{j=i}^l \tau_j \cdot 2^{j-i}$$

(sodass also $t_0 = t$). Durch Rückwärtsinduktion für i von $l, \dots, 0$ zeigen wir nun, dass

$$h_i = t_i \cdot g$$

(mit h_i aus der obigen Konstruktion)

Induktionsanfang $i = l$:

Es ist

$$t_l = \sum_{j=l}^l \tau_j \cdot 2^{j-l} = \tau_l = 1$$

und daher gilt

$$t_l \cdot g = 1 \cdot g = g = h_l$$

Induktionsschritt $i \rightarrow i - 1$ ($l \geq i \geq 1$):

Wir nehmen an, dass wir für $t_i = \sum_{j=i}^l \tau_j \cdot 2^{j-i}$ bereits wissen, dass

$$h_i = t_i \cdot g$$

(Induktionsvoraussetzung) und schließen daraus, dass für

$$\begin{aligned} \widetilde{t}_{i-1} &= 2 \cdot t_i \\ &= 2 \cdot \sum_{j=i}^l \tau_j \cdot 2^{j-i} \\ &= \sum_{j=i}^l \tau_j \cdot 2^{j-i+1} \\ &= \sum_{j=i}^l \tau_j \cdot 2^{j-(i-1)} \end{aligned}$$

gilt

$$\widetilde{h}_{i-1} := h_i + h_i = 2 \cdot h_i = 2 \cdot t_i \cdot g = \widetilde{t}_{i-1} \cdot g$$

Falls $\tau_{i-1} = 0$, so gilt

$$\widetilde{t_{i-1}} = \sum_{j=i}^l \tau_j \cdot 2^{j-(i-1)} = \sum_{j=i-1}^l \tau_j \cdot 2^{j-(i-1)} = t_{i-1}$$

und

$$\widetilde{h_{i-1}} = h_{i-1}$$

sodass also in diesem Fall folgt

$$h_{i-1} = t_{i-1} \cdot g$$

Falls $\tau_{i-1} = 1$, so gilt

$$\widetilde{t_{i-1}} + 1 = \sum_{j=i}^l \tau_j \cdot 2^{j-(i-1)} + 1 = \sum_{j=i-1}^l \tau_j \cdot 2^{j-(i-1)} = t_{i-1}$$

und

$$h_{i-1} = \widetilde{h_{i-1}} + g = \widetilde{t_{i-1}} \cdot g + g = (\widetilde{t_{i-1}} + 1) \cdot g = t_{i-1} \cdot g$$

und die Induktionsbehauptung ist auch in diesem Fall gezeigt.

Damit ist die Aussage bewiesen, und es gilt speziell

$$h = h_0 = t_0 \cdot g = t \cdot g$$

Beispiel 14.1. Wir betrachten die elliptische Kurve \overline{E} über \mathbb{F}_{17} , gegeben durch das Polynom

$$F(X, Y) = Y^2 - X^3 - 11X - 3 = Y^2 + 16X^3 + 6X + 14 \in \mathbb{F}_{17}[X, Y]$$

Dann ist der Punkt $g = (5, 8)$ auf \overline{E} , der die Ordnung 21 hat und die gesamte Gruppe \overline{E} erzeugt. Der Punkt $10 \cdot g$ berechnet sich wie folgt:

Zunächst ist $10 = 0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3$, sodass also

$$\tau = (1, 0, 1, 0)$$

Wir beginnen mit $l = 3$ und setzen

$$h_3 = (5, 8)$$

$l = 2$: Wir berechnen

$$\widetilde{h_2} = h_3 + h_3 = (5, 8) + (5, 8)$$

Es ist

$$m = \frac{3 \cdot 5^2 + 11}{2 \cdot 8} = \frac{1}{16} = 16$$

und damit ist der Schnittpunkt $Q = (u, v)$ der Tangente T an E in g gegeben durch

$$u = m^2 - 2 \cdot r = 16^2 - 2 \cdot 5 = 8$$

und

$$v = m \cdot (u - r_1) + s_1 = 16 \cdot (8 - 5) + 8 = 5$$

also ist

$$\tilde{h}_2 = (8, -5) = (8, 12)$$

Da $\tau_2 = 0$ ist also

$$h_2 = \tilde{h}_2 = (8, -5) = (8, 12)$$

$l = 1$: Wir berechnen

$$\tilde{h}_1 = h_2 + h_2 = (9, 10)$$

wobei hier

$$m = \frac{3 \cdot 8^2 + 11}{2 \cdot 12} = \frac{16}{7} = 12$$

Da $\tau_1 = 1$ gilt

$$h_1 = \tilde{h}_1 + g = (9, 10) + (5, 8) = (16, 12)$$

wobei hier

$$m = \frac{8 - 10}{5 - 9} = \frac{1}{2} = 9$$

$l = 0$: Wir berechnen

$$\tilde{h}_0 = h_1 + h_1 = (6, 8)$$

wobei hier

$$m = \frac{3 \cdot 16^2 + 11}{2 \cdot 12} = \frac{14}{7} = 2$$

Da $\tau_0 = 0$ gilt

$$h_0 = \tilde{h}_0 = (6, 8)$$

Damit ist

$$h = 10 \cdot g = (6, 8)$$

Beispiel 14.2. Wir betrachten die elliptische Kurve \overline{E} über \mathbb{F}_{19} , gegeben durch das Polynom

$$F(X, Y) = Y^2 - X^3 - 14X - 9 = Y^2 + 18X^3 + 5X + 10 \in \mathbb{F}_{19}[X, Y]$$

(also mit $a = 14$ und $b = 9$). Dann hat \overline{E} genau 23 Punkte (ist also von Primzahlordnung) und der Punkt $g = (8, 5)$ ist auf \overline{E} , notwendig also ein Erzeuger, da $g \neq \infty$.

Der Punkt $h = 5 \cdot g$ berechnet sich wie folgt:

Zunächst ist $5 = 1 \cdot 2^0 + 0 \cdot 2^2 + 1 \cdot 2^2$, sodass also

$$\tau = (1, 0, 1)$$

Wir beginnen mit $l = 2$ und setzen

$$h_2 = g = (8, 5)$$

$l = 1$: Wir ermitteln

$$m = \frac{3 \cdot 8^2 + 14}{2 \cdot 5} = \frac{16}{10} = 2 \cdot 16 = 13$$

und berechnen den Schnittpunkt $Q = (u, v)$ der Tangente T an E in h_2 durch

$$\begin{aligned} u &= 13^2 - 2 \cdot 8 &= 1 \\ v &= 13 \cdot (1 - 8) + 5 &= 9 \end{aligned}$$

und damit ist

$$\widetilde{h}_1 = h_2 + h_2 = (1, -9) = (1, 10)$$

Da $\tau_1 = 0$ gilt

$$h_1 = \widetilde{h}_1 = (1, 10)$$

$l = 0$: Wir ermitteln

$$m = \frac{3 \cdot 1^2 + 14}{2 \cdot 10} = 17$$

und berechnen den Schnittpunkt $Q = (u, v)$ der Tangente T an E in h_1 durch

$$\begin{aligned} u &= 17^2 - 2 \cdot 1 &= 2 \\ v &= 17 \cdot (2 - 1) + 10 &= 8 \end{aligned}$$

und damit

$$\widetilde{h}_0 = h_1 + h_1 = (2, -8) = (2, 11)$$

Da $\tau_0 = 1$, ist noch g zu addieren. Dabei ist

$$m = \frac{8 - 2}{5 - 11} = \frac{6}{-6} = -1 = 18$$

und daher gilt für den dritten Schnittpunkt $Q = (u, v)$ der Geraden L durch \widetilde{h}_0 und g mit E :

$$\begin{aligned} u &= 18^2 - 8 - 2 &= 10 \\ v &= 18 \cdot (10 - 2) + 11 &= 3 \end{aligned}$$

und damit

$$h_0 = \widetilde{h}_0 + g = (2, 11) + (8, 5) = (10, -3) = (10, 16)$$

Damit ist

$$h = 5 \cdot g = (10, 16)$$

Beispiel 14.3. Wir betrachten die elliptische Kurve \overline{E} über \mathbb{F}_8 (mit $\alpha^3 = \alpha + 1$), die gegeben ist durch

$$F(X, Y) = Y^2 + YX + X^3 + X^2 + \alpha \in \mathbb{F}_8[X, Y]$$

Die Kurve (und Gruppe) \overline{E} besteht aus 10 Elementen,

$$\begin{aligned} \overline{E} = & \{(0, \alpha^2 + \alpha), (1, \alpha^2), (1, \alpha^2 + 1), (\alpha, \alpha^2), \\ & (\alpha, \alpha^2 + \alpha), (\alpha + 1, \alpha^2 + \alpha), (\alpha + 1, \alpha^2 + 1), \\ & (\alpha^2 + \alpha, \alpha^2 + \alpha + 1), (\alpha^2 + \alpha, 1), \infty\} \end{aligned}$$

und $g = (\alpha^2 + \alpha, \alpha^2 + \alpha + 1)$ ist ein Element von \overline{E} (von der Ordnung 10).

Der Punkt $h = 9 \cdot g$ berechnet sich wie folgt:

Zunächst ist $9 = 1 \cdot 2^0 + 0 \cdot 2^2 + 0 \cdot 2^2 + 1 \cdot 2^3$, sodass also

$$\tau = (1, 0, 0, 1)$$

Wir beginnen mit $l = 3$ und setzen

$$h_3 = g = (\alpha^2 + \alpha, \alpha^2 + \alpha + 1)$$

$l = 2$: Wir berechnen

$$\widetilde{h}_2 = h_3 + h_3 = (\alpha^2 + \alpha, \alpha^2 + \alpha + 1) + (\alpha^2 + \alpha, \alpha^2 + \alpha + 1)$$

Es ist

$$m = r_1 + \frac{s_1}{r_1} = \alpha^2 + \alpha + \frac{\alpha^2 + \alpha + 1}{\alpha^2 + \alpha} = \alpha^2 + \alpha + \alpha = \alpha^2$$

und damit ist $h_3 + h_3 = (u, v)$ gegeben durch

$$u = m^2 + m + a = \alpha^4 + \alpha^2 + 1 = \alpha^2 + \alpha + \alpha^2 + 1 = \alpha + 1$$

und

$$v = m \cdot (u + r_1) + u + s_1 = (\alpha + 1) \cdot (\alpha^2 + 1) + \alpha^2 + 1 + \alpha^2 + \alpha + 1 = \alpha^3 \cdot \alpha^9 + \alpha = \alpha^2 + \alpha$$

also ist

$$\widetilde{h}_2 = (\alpha + 1, \alpha^2 + \alpha)$$

Da $\tau_2 = 0$ ist also

$$h_2 = \widetilde{h}_2 = (\alpha + 1, \alpha^2 + \alpha)$$

$l = 1$: Wir berechnen

$$\widetilde{h}_1 = h_2 + h_1 = (1, \alpha^2 + 1)$$

(wobei $m = 1$). Da $\tau_1 = 0$ gilt

$$h_1 = \widetilde{h}_1 = (1, \alpha^2 + 1)$$

$l = 0$: Wir berechnen

$$\widetilde{h}_0 = h_1 + h_0 = (\alpha + 1, \alpha^2 + 1)$$

(wobei hier $m = r_1 + \frac{s_1}{r_1} = 1 + \frac{\alpha^2 + 1}{1} = \alpha^2$). Da $\tau_0 = 1$ gilt

$$h_0 = \widetilde{h}_0 + g = (\alpha + 1, \alpha^2 + 1) + (\alpha^2 + \alpha, \alpha^2 + \alpha + 1) = (\alpha^2 + \alpha, 1)$$

$$(\text{mit } m = \frac{s_1+s_2}{r_1+r_2} = \frac{\alpha}{\alpha^2+1} = \alpha^2).$$

Damit ist

$$h = 9 \cdot g = (\alpha^2 + \alpha, 1)$$

Beispiel 14.4. Wir betrachten die elliptische Kurve \overline{E} über \mathbb{F}_8 (mit $\alpha^3 = \alpha + 1$), die gegeben ist durch

$$F(X, Y) = Y^2 + YX + X^3 + X^2 + 1 \in \mathbb{F}_8[X, Y]$$

Die Kurve (und Gruppe) \overline{E} besteht aus 14 Elementen,

$$\begin{aligned} \overline{E} = & \{(0, 1), (\alpha, \alpha^2 + \alpha + 1), (\alpha, \alpha^2 + 1), (\alpha + 1,), \\ & (\alpha + 1, \alpha + 1), (\alpha^2, \alpha + 1), (\alpha^2, \alpha^2 + \alpha + 1), (\alpha^2 + 1, 0), \\ & (\alpha^2 + 1, \alpha^2 + 1), (\alpha^2 + \alpha, \alpha + 1), (\alpha^2 + \alpha, \alpha + 1), \\ & (\alpha^2 + \alpha + 1, 0), (\alpha^2 + \alpha + 1), \infty\} \end{aligned}$$

und $g = (\alpha + 1, \alpha + 1)$ ist ein Element von \overline{E} von der Ordnung 7, erzeugt also eine Untergruppe $U \subseteq \overline{E}$ von der Primzahlordnung $r = 7$.

Der Punkt $h = 6 \cdot g$ berechnet sich wie folgt:

Zunächst ist $6 = 0 \cdot 2^0 + 1 \cdot 2^2 + 1 \cdot 2^2$, sodass also

$$\tau = (1, 1, 0)$$

Wir beginnen mit $l = 2$ und setzen

$$h_2 = g = (\alpha + 1, \alpha + 1)$$

$l = 1$: Wir berechnen

$$\widetilde{h}_1 = h_2 + h_2 = (\alpha + 1, \alpha + 1) + (\alpha + 1, \alpha + 1)$$

Es ist

$$m = r_1 + \frac{s_1}{r_1} = \alpha + 1 + \frac{\alpha + 1}{\alpha + 1} = \alpha$$

und damit ist $h_2 + h_2 = (u, v)$ gegeben durch

$$u = m^2 + m + a = \alpha^2 + \alpha + 1$$

und

$$v = m \cdot (u + r_1) + u + s_1 = \alpha \cdot \alpha^2 + \alpha^2 + \alpha + 1 + \alpha + 1 = \alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1$$

also ist

$$\widetilde{h}_1 = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1)$$

Da $\tau_1 = 1$ ist also

$$h_1 = \widetilde{h}_1 + g = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1) + (\alpha + 1, \alpha + 1) = (\alpha^2 + 1, 0)$$

(wobei hier $m = \frac{s_1 s_2}{r_1 + r_2} = \frac{\alpha^2}{\alpha^2} = 1$).

$l = 0$: Wir berechnen

$$\widetilde{h}_0 = h_1 + h_1 = (\alpha^2 + 1, 0) + (\alpha^2 + 1, 0) = (\alpha + 1, 0)$$

(wobei hier $m = r_1 + \frac{s_1}{r_1} = \alpha^2 + 1$). Da $\tau_0 = 0$ gilt

$$h_0 = \widetilde{h}_0 = (\alpha + 1, 0)$$

Damit ist

$$h = 6 \cdot g = (\alpha + 1, 0)$$

14.2. ECDH Diffie–Hellman–Schlüsselaustausch mit elliptischen Kurven

Elliptische Kurven \overline{E} über endlichen Körpern \mathbb{F}_q mit $q = p^l$ Elementen und zyklische Untergruppen $U \subseteq \overline{E}$ von Primzahlordnung r können zum geheimen Schlüsselaustausch benutzt werden. Die Grundidee dahinter ist ähnlich zur Idee hinter dem Diffie–Hellman–Schlüsselaustausch.

Vorbereitung:

Alice und Bob einigen sich auf die folgenden Daten:

1. Auf eine Primzahl $p \neq 3$.
2. Auf ein $l > 1$ falls $p = 2$ bzw. auf $l = 1$ falls $p > 3$ und auf $q = p^l$.
3. Auf eine elliptische Kurve \overline{E} über \mathbb{F}_q , gegeben durch

$$F(X, Y) = Y^2 - X^3 - aX - b \in \mathbb{F}_q[X, Y]$$

falls $p > 3$ bzw.

$$F(X, Y) = Y^2 + XY + X^3 + aX^2 + b \in \mathbb{F}_q[X, Y]$$

falls $p = 2$.

4. Auf einen Punkt $g = (u, v) \in E$ der Ordnung $r = \text{ord}(g)$, wobei r eine Primzahl ist und auf die von g erzeugte Untergruppe $U = U_g = \langle g \rangle \subseteq \overline{E}$.

Schlüsselaustausch:

Alice und Bob einigen sich auf einen Schlüssel wie folgt:

Alice geht vor wie folgt:

1. Alice wählt ihren privaten Schlüssel $k_{\text{pr}, A} = a \in \{2, \dots, r - 1\}$.
2. Alice berechnet ihren öffentlichen Schlüssel $k_{\text{pub}, A} = a \cdot g \in U$.
3. Alice schickt $k_{\text{pub}, A}$ an Bob.

Bob geht vor wie folgt:

1. Bob wählt seinen privaten Schlüssel $k_{\text{pr}, B} = b \in \{2, \dots, r - 1\}$.
2. Bob berechnet seinen öffentlichen Schlüssel $k_{\text{pub}, B} = b \cdot g \in U$.
3. Bob schickt $k_{\text{pub}, B}$ an Alice.

Alice und Bob berechnen den gemeinsamen Schlüssel wie folgt:

1. Alice berechnet $T_A = a \cdot k_{\text{pub}, B}$.
2. Bob berechnet $T_B = b \cdot k_{\text{pub}, A}$.

Hilfssatz 14.2. $T_A = T_B =: T_{A,B}$.

Beweis: Es gilt

$$\begin{aligned} T_A &= a \cdot k_{\text{pub},B} = a \cdot b \cdot g \\ &= b \cdot a \cdot g = b \cdot k_{\text{pub},A} \\ &= T_B \end{aligned}$$

Beispiel 14.5. Alice und Bob einigen sich auf die Primzahl $p = 13$ und auf die elliptische Kurve \overline{E} über \mathbb{F}_{13} , die durch das Polynom

$$F(X, Y) = Y^2 - X^3 - 7X - 11 \in \mathbb{F}_{13}[X, Y]$$

definiert wird. Diese besteht aus den Punkten

$$\overline{E} = \{(4, 5), (4, 8), (6, 3), (6, 10), (7, 0), (9, 6), (9, 7), (12, 4), (12, 9), \infty\}$$

und $|\overline{E}| = 10 = 2 \cdot 5$.

Für $g = (4, 5)$ gilt $\text{ord}(g) = 5$ und

$$\begin{aligned} U_g &= \{1 \cdot (4, 5), 2 \cdot (4, 5), 3 \cdot (4, 5), 4 \cdot (4, 5), 5 \cdot (4, 5)\} \\ &= \{(4, 5), (6, 10), (6, 3), (4, 8), \infty\} \end{aligned}$$

Alice und Bob wählen $g = (4, 5)$ aus.

Vorgehen von Alice:

1. Alice wählt als privaten Schlüssel $k_{\text{pr},A} = 2$.
2. Alice berechnet ihren öffentlichen Schlüssel $k_{\text{pub},A} = 2 \cdot (4, 5) = (6, 10)$.
3. Alice schickt $k_{\text{pub},A} = (6, 10)$ an Bob.

Vorgehen von Bob:

1. Bob wählt als privaten Schlüssel $k_{\text{pr},B} = 4$.
2. Bob berechnet seinen öffentlichen Schlüssel $k_{\text{pub},B} = 4 \cdot (4, 5) = (4, 8)$.
3. Bob schickt $k_{\text{pub},B} = (4, 8)$ an Alice.

Schlüsselfestlegung:

1. Alice berechnet $T_A = 2 \cdot k_{\text{pub},B} = 2 \cdot (4, 8) = (6, 3)$.
2. Bob berechnet $T_B = 4 \cdot k_{\text{pub},A} = 4 \cdot (6, 10) = (6, 3)$.

Alice und Bob haben sich also auf das gemeinsame Geheimnis $T_{A,B} = (6, 3)$ geeinigt.

Beispiel 14.6. Die elliptische Kurve \bar{E} in Beispiel 14.5 besteht aus 10 Elementen, und sie ist in der Tat zyklisch von der Ordnung 10 und $g = (9, 7)$ ist ein Element der Ordnung 10, also ein Erzeuger von \bar{E} . Die Wahl dieses Elements g kann aber zu Problemen führen, wie das folgende Beispiel zeigt:

Alice wählt als privaten Schlüssel $k_{\text{pr},A} = 5$ und berechnet

$$k_{\text{pub},A} = 5 \cdot (9, 7) = (7, 0)$$

Bob wählt als privaten Schlüssel $k_{\text{pr},B} = 2$ und berechnet

$$k_{\text{pub},B} = 2 \cdot (9, 7) = (4, 8)$$

Alice ermittelt dann

$$T_A = 5 \cdot (4, 8) = \infty$$

und Bob berechnet

$$T_B = 2 \cdot (7, 0) = \infty$$

Das gemeinsame Geheimnis von Alice und Bob ist also ∞ . Das ist aber als gemeinsames Geheimnis (etwa als neuer Schlüssel) nicht brauchbar. Um solche Probleme gleich von Anfang an auszuschließen, wird g so gewählt, dass die Ordnung von g eine Primzahl ist.

Beispiel 14.7. Wir betrachten die elliptische Kurve \bar{E} über \mathbb{F}_{17} die gegeben ist durch

$$F(X, Y) = Y^2 - X^3 - 15X - 8 = Y^2 + 16X^3 + 2X + 9 \in \mathbb{F}_{17}[X, Y]$$

Diese besteht aus den Punkten

$$\begin{aligned} \bar{E} = & \{(0, 5), (0, 12), (4, 8), (4, 9), (5, 2), (5, 15), (6, 5), (6, 12), (10, 6), \\ & (10, 11), (11, 5), (11, 12), (14, 2), (14, 15), (15, 2), (15, 15), (16, 3), (16, 14), \infty\} \end{aligned}$$

Insbesondere gilt also $|\bar{E}| = 19$. Damit ist \bar{E} zyklisch von Primzahlordnung $r = 19$ und jedes Element $g \in E$ (also jedes $g \in \bar{E} \setminus \{\infty\}$) ist ein Erzeuger.

Alice und Bob einigen sich auf $p = 17$ und diese Kurve \bar{E} und auf $g = (4, 8)$ als Erzeuger (und damit auf $U_g = \bar{E}$).

Vorgehen von Alice:

1. Alice wählt als privaten Schlüssel $k_{\text{pr},A} = 7$.
2. Alice berechnet ihren öffentlichen Schlüssel $k_{\text{pub},A} = 7 \cdot (4, 8) = (5, 15)$.

3. Alice schickt $k_{\text{pub},A} = (5, 15)$ an Bob.

Vorgehen von Bob:

1. Bob wählt als privaten Schlüssel $k_{\text{pr},B} = 13$.
2. Bob berechnet seinen öffentlichen Schlüssel $k_{\text{pub},B} = 13 \cdot (4, 8) = (10, 11)$.
3. Bob schickt $k_{\text{pub},B} = (10, 11)$ an Alice.

Schlüsselfestlegung:

1. Alice berechnet $T_A = 7 \cdot k_{\text{pub},B} = 7 \cdot (10, 11) = (15, 15)$.
2. Bob berechnet $T_B = 4 \cdot k_{\text{pub},A} = 13 \cdot (5, 15) = (15, 15)$.

Alice und Bob haben sich also auf das gemeinsame Geheimnis $T_{A,B} = (15, 15)$ geeinigt.

Beispiel 14.8. Wir betrachten die elliptische Kurve \bar{E} über \mathbb{F}_8 (mit $\alpha^3 = \alpha + 1$), die gegeben ist durch

$$F(X, Y) = Y^2 + YX + X^3 + (\alpha^2 + \alpha + 1) \cdot X^2 + 1 \in \mathbb{F}_8[X, Y]$$

Die Kurve (und Gruppe) \bar{E} besteht aus 14 Elementen,

$$\begin{aligned} \bar{E} = & \{(0, 1), (\alpha, \alpha + 1), (\alpha, 1), (\alpha + 1, \alpha^2 + \alpha), \\ & (\alpha + 1, \alpha^2 + 1), (\alpha^2, 0), (\alpha^2, \alpha^2), (\alpha^2 + 1, 1), \\ & (\alpha^2 + 1, \alpha^2), (\alpha^2 + \alpha, \alpha), (\alpha^2 + \alpha, \alpha^2), \\ & (\alpha^2 + \alpha + 1, \alpha^2 + 1), (\alpha^2 + \alpha + 1, \alpha), \infty\} \end{aligned}$$

und das Element $g = (\alpha + 1, \alpha^2 + 1)$ erzeugt eine Untergruppe der Primzahlordnung $r = 7$.

Alice und Bob einigen sich auf $p = 2$, $q = 2^3$, diese Kurve \bar{E} , auf $g = (\alpha + 1, \alpha^2 + 1)$ und auf die von U_g erzeugte Untergruppe von \bar{E} von der Ordnung 7.

Vorgehen von Alice:

1. Alice wählt als privaten Schlüssel $k_{\text{pr},A} = 6$.
2. Alice berechnet ihren öffentlichen Schlüssel

$$k_{\text{pub},A} = 6 \cdot (\alpha + 1, \alpha^2 + 1) = (\alpha + 1, \alpha^2 + \alpha)$$

3. Alice schickt $k_{\text{pub},A} = (\alpha + 1, \alpha^2 + \alpha)$ an Bob.

Vorgehen von Bob:

1. Bob wählt als privaten Schlüssel $k_{\text{pr},B} = 5$.
2. Bob berechnet seinen öffentlichen Schlüssel

$$k_{\text{pub},B} = 5 \cdot (\alpha + 1, \alpha^2 + 1) = (\alpha^2 + \alpha + 1, \alpha^2 + 1)$$

3. Bob schickt $k_{\text{pub},B} = (\alpha^2 + \alpha + 1, \alpha^2 + 1)$ an Alice.

Schlüsselfestlegung:

1. Alice berechnet $T_A = 6 \cdot k_{\text{pub},B} = 6 \cdot (\alpha^2 + \alpha + 1, \alpha^2 + 1) = (\alpha^2 + \alpha + 1, \alpha)$.
2. Bob berechnet $T_B = 4 \cdot k_{\text{pub},A} = 5 \cdot (\alpha + 1, \alpha^2 + 1) = (\alpha^2 + \alpha + 1, \alpha)$.

Alice und Bob haben sich also auf das gemeinsame Geheimnis $T_{A,B} = (\alpha^2 + \alpha + 1, \alpha)$ geeinigt.

Beispiel 14.9. Wir betrachten die elliptische Kurve \bar{E} über \mathbb{F}_8 (mit $\alpha^3 = \alpha + 1$), die gegeben ist durch

$$F(X, Y) = Y^2 + YX + X^3 + (\alpha + 1) \cdot X^2 + 1 \in \mathbb{F}_8[X, Y]$$

Die Kurve (und Gruppe) \bar{E} besteht aus 14 Elementen,

$$\begin{aligned} \bar{E} = & \{(0, 1), (\alpha, \alpha^2), (\alpha, \alpha^2 + \alpha), (\alpha + 1, \alpha^2 + \alpha + 1), \\ & (\alpha + 1, \alpha^2), (\alpha^2, \alpha^2 + 1), (\alpha^2, 1), (\alpha^2 + 1, \alpha), \\ & (\alpha^2 + 1, \alpha^2 + \alpha + 1), (\alpha^2 + \alpha, 0), (\alpha^2 + \alpha, \alpha^2 + \alpha), \\ & (\alpha^2 + \alpha + 1, 1), (\alpha^2 + \alpha + 1, \alpha^2 + \alpha), \infty\} \end{aligned}$$

und das Element $g = (\alpha + 1, \alpha^2 + \alpha + 1)$ erzeugt eine Untergruppe der Primzahlordnung $r = 7$.

Alice und Bob einigen sich auf $p = 2$, $q = 2^3$, diese Kurve \bar{E} , auf $g = (\alpha + 1, \alpha^2 + \alpha + 1)$ und auf die von U_g erzeugte Untergruppe von \bar{E} von der Ordnung 7.

Vorgehen von Alice:

1. Alice wählt als privaten Schlüssel $k_{\text{pr},A} = 3$.
2. Alice berechnet ihren öffentlichen Schlüssel

$$k_{\text{pub},A} = 3 \cdot (\alpha + 1, \alpha^2 + \alpha + 1) = (\alpha^2 + 1, \alpha^2 + \alpha + 1)$$

3. Alice schickt $k_{\text{pub},A} = (\alpha^2 + 1, \alpha^2 + \alpha + 1)$ an Bob.

Vorgehen von Bob:

1. Bob wählt als privaten Schlüssel $k_{\text{pr},B} = 4$.
2. Bob berechnet seinen öffentlichen Schlüssel

$$k_{\text{pub},B} = 4 \cdot (\alpha + 1, \alpha^2 + \alpha + 1) = (\alpha^2 + 1, \alpha)$$

3. Bob schickt $k_{\text{pub},B} = (\alpha^2 + 1, \alpha)$ an Alice.

Schlüsselfestlegung:

1. Alice berechnet $T_A = 3 \cdot k_{\text{pub},B} = 3 \cdot (\alpha^2 + 1, \alpha) = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha)$.
2. Bob berechnet $T_B = 4 \cdot k_{\text{pub},A} = 4 \cdot (\alpha^2 + 1, \alpha^2 + \alpha + 1) = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha)$.

Alice und Bob haben sich also auf das gemeinsame Geheimnis $T_{A,B} = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha)$ geeinigt.

Bemerkung 14.2. Von der Struktur her ist ECDH sehr ähnlich zum Diffie–Hellman–Verfahren DH mit \mathbb{F}_p^* , hat aber einen entscheidenden Vorteil:

Der beste bekannte Algorithmus zur Bestimmung des diskreten Logarithmus ist der sogenannte Zahlkörper–Sieb, der über \mathbb{F}_p den diskreten Logarithmus größtenordnungsmäßig in

$$T_{\text{DH}}(l) = \exp \left(c_0 \cdot \sqrt[3]{l} \cdot \sqrt[4]{\ln(l \cdot \ln(2))^2} \right)$$

elementaren Operationen (Additionen, Multiplikationen, Quadrierungen) berechnet, wobei $l = \lfloor \log_2(p) \rfloor + 1$ und wobei $c_0 = \frac{4}{3} \cdot \sqrt[3]{3}$. Insbesondere hat dieser Algorithmus also subexponentielle Laufzeit.

Sieht man von einigen „schwachen“ elliptischen Kurven ab (die noch behandelt werden) und wählt man zyklische Untergruppen $U \subseteq \overline{E}$ über \mathbb{F}_q so, dass $r = |U| \sim q$, so benötigen die besten Algorithmen zur Berechnung des diskreten Logarithmus in U etwa (größenordnungsmäßig) $\sqrt{r} \sim \sqrt{q}$ viele elementare Rechenoperationen. Mit $l = \lfloor \log_2(q) \rfloor + 1$ haben diese Algorithmen also eine Laufzeit in der Größenordnung

$$T_{\text{ECDH}}(l) = 2^{\frac{l}{2}}$$

und damit exponentielle Laufzeit.

Bezeichnen wir daher mit $\ell = \lfloor \log_2(q) \rfloor + 1$ die Schlüssellänge von DH bzw. ECDH, so kann das gleiche Sicherheitsniveau bei ECDH mit einer deutlich kürzeren Schlüssellänge erreicht werden. Bezeichnen wir dazu genauer mit l die Schlüssellänge für einen ECDH–Algorithmus (bei wie oben gewähltem U), so ergibt ein Vergleich von T_{ECDH} und T_{DH} , dass für einen vergleichbaren Rechenaufwand, also für vergleichbare Sicherheit, bei Diffie–Hellman der Schlüssel um den Faktor

$$m_{\text{DH/ECDH}}(l) = \frac{2}{\ln(2)} \cdot c_0 \cdot l^{\frac{1}{3}} \cdot (\ln(l \cdot \ln(2)))^{\frac{2}{3}}$$

länger sein muss.

Einen konkreten Überblick darüber bietet die folgende Tabelle der Schlüssellängen (in Bit) für eine vergleichbare Sicherheit

AES	RSA / DH	ECDH
80	1024	160
112	2048	224
128	3072	256
192	7980	384
256	15360	521

Tabelle 1: Schlüssellängen für vergleichbare Sicherheit

14.3. ElGamal–Verschlüsselung mit elliptischen Kurven

Elliptische Kurven können auch benutzt werden, um eine verschlüsselte Nachricht zu verschicken. Das Vorgehen orientiert sich dabei am Verfahren der ElGamal–Verschlüsselung (und greift damit wie dieses auf Ideen und Konstruktionen des Diffie–Hellman–Schlüsselaustauschverfahrens zurück).

Das Verfahren etabliert eine Einwegkommunikation (mit Bob als Empfänger und Alice als Sender) und ist aufgebaut wie folgt:

Vorbereitung durch Bob:

Bob wählt für sich die folgenden Daten:

1. Eine Primzahl $p \neq 3$.
2. Ein $l > 1$ falls $p = 2$ bzw. $l = 1$ falls $p > 3$ und $q = p^l$.

3. Eine elliptische Kurve \overline{E} über \mathbb{F}_q , gegeben durch

$$F(X, Y) = Y^2 - X^3 - aX - b \in \mathbb{F}_q[X, Y]$$

falls $p > 3$ bzw.

$$F(X, Y) = Y^2 + XY + X^3 + aX^2 + b \in \mathbb{F}_q[X, Y]$$

falls $p = 2$.

4. Einen Punkt $g = (u, v) \in E$ der Ordnung $r = \text{ord}(g)$, wobei r eine Primzahl ist und die von g erzeugte Untergruppe $U = U_g = \langle g \rangle \subseteq \overline{E}$.

Schlüsselerstellung durch Bob:

Bob erzeugt ein Schlüsselpaar $(k_{\text{pr}}, k_{\text{pub}})$ wie folgt:

1. Bob wählt zufällig eine Zahl $u \in \{2, \dots, r - 1\}$.
2. Bob berechnet $v = u \cdot g$ in U_g .
3. Bobs privater Schlüssel ist $k_{\text{pr}} = (q, \overline{E}, g, u)$.
4. Bobs öffentlicher Schlüssel ist $k_{\text{pub}} = (q, \overline{E}, g, v)$.
5. Bob veröffentlicht k_{pub} .

Nachrichtenverschlüsselung durch Alice:

Alice will eine Nachricht $m \in U_g$ geheim und verschlüsselt an Bob schicken. Dazu geht sie vor wie folgt.

1. Alice wählt zufällig ein $t \in \{2, \dots, r - 1\}$.
2. Alice benutzt Bobs öffentlichen Schlüssel $k_{\text{pub}} = (q, \overline{E}, g, v)$ und berechnet

$$c_1 = t \cdot g, \quad c_2 = m + t \cdot v$$

3. Alice schickt $c = (c_1, c_2)$ an Bob.

Nachrichtenentschlüsselung durch Bob:

Bob empfängt die Nachricht $c = (c_1, c_2)$ und entschlüsselt sie wie folgt:

1. Bob berechnet $d = u \cdot c_1$.
2. Bob berechnet $\tilde{m} = c_2 - d$.
3. Bob arbeitet mit \tilde{m} als Nachricht.

Satz 14.3. *Es ist $\tilde{m} = m$.*

Beweis: Es gilt

$$d = u \cdot c_1 = u \cdot t \cdot g = t \cdot u \cdot g = t \cdot v$$

Damit ist

$$\tilde{m} = c_2 - d = c_2 - t \cdot v = m + t \cdot v - t \cdot v = m$$

Also hat Bob die Nachricht tatsächlich korrekt entschlüsselt.

Beispiel 14.10. Bob wählt $p = 17$ (und damit $l = 1$, $q = p$) und die elliptische Kurve \bar{E} über \mathbb{F}_{17} gegeben durch

$$F(X, Y) = Y^2 - X^3 - 15X - 8 = Y^2 + 16X^3 + 2X + 9 \in \mathbb{F}_{17}[X, Y]$$

Diese besteht (vergleiche Beispiel 14.7) aus den Punkten

$$\bar{E} = \{(0, 5), (0, 12), (4, 8), (4, 9), (5, 2), (5, 15), (6, 5), (6, 12), (10, 6), (10, 11), (11, 5), (11, 12), (14, 2), (14, 15), (15, 2), (15, 15), (16, 3), (16, 14), \infty\}$$

und \bar{E} ist zyklisch von Primzahlordnung $r = 19$. Jedes Element $g \in E$ (also jedes $g \in \bar{E} \setminus \{\infty\}$) ist ein Erzeuger.

Bob wählt $g = (4, 8)$ (und damit natürlich $U_g = \bar{E}$) und $u = 7$ und berechnet

$$v = 7 \cdot (4, 8) = (5, 15)$$

1. Bobs privater Schlüssel ist $k_{\text{pr}} = (17, \bar{E}, (4, 8), 7)$.
2. Bobs öffentlicher Schlüssel ist $k_{\text{pr}} = (17, \bar{E}, (4, 8), (5, 15))$.

Alice will die Nachricht $m = (11, 12)$ geheim an Bob schicken.

1. Alice wählt die Zahl $t = 5 \in \{2, \dots, 18\}$
2. Alice berechnet $c_1 = 5 \cdot g = 5 \cdot (4, 8) = (16, 14)$.

3. Alice berechnet $5 \cdot v = 5 \cdot (5, 15) = (14, 15)$ und

$$c_2 = (11, 12) + 5 \cdot (5, 15) = (11, 12) + (14, 15) = (10, 6)$$

4. Alice schickt $c = (c_1, c_2) = ((16, 14), (10, 6))$ an Bob.

Bob entschlüsselt die empfangene Nachricht $c = ((16, 14), (10, 6))$ wie folgt:

1. Bob berechnet $d = u \cdot c_1 = 7 \cdot (16, 14) = (14, 15)$.

2. Bob berechnet $-d = (14, -15) = (14, 2)$.

3. Bob berechnet

$$m = c_2 + (-d) = (10, 6) + (14, 2) = (11, 12)$$

Bob hat die Nachricht korrekt entschlüsselt.

Beispiel 14.11. Bob wählt $p = 19$ (und damit $l = 1, q = p$) und die elliptische Kurve \overline{E} über \mathbb{F}_{19} gegeben durch

$$F(X, Y) = Y^2 - X^3 - 11X - 7 = Y^2 + 18X^3 + 8X + 12 \in \mathbb{F}_{19}[X, Y]$$

Diese besteht aus den Punkten

$$\begin{aligned} \overline{E} = & \{(0, 8), (0, 11), (4, 1), (4, 18), (5, 4), (5, 15), (6, 2), (6, 17), (7, 16), \\ & (7, 3), (12, 9), (12, 10), (14, 6), (14, 13), (16, 2), (16, 17), \infty\} \end{aligned}$$

Also ist $|\overline{E}| = 17$ und damit ist \overline{E} zyklisch von Primzahlordnung $r = 17$. Jedes Element $g \in E$ (also jedes $g \in \overline{E} \setminus \{\infty\}$) ist ein Erzeuger.

Bob wählt $g = (6, 2)$ (und damit natürlich $U_g = \overline{E}$) und $u = 11$ und berechnet

$$v = 11 \cdot (6, 2) = (5, 4)$$

1. Bobs privater Schlüssel ist $k_{\text{pr}} = (19, \overline{E}, (6, 2), 11)$.

2. Bobs öffentlicher Schlüssel ist $k_{\text{pr}} = (19, \overline{E}, (6, 2), (5, 4))$.

Alice will die Nachricht $m = (14, 6)$ geheim an Bob schicken.

1. Alice wählt die Zahl $t = 6 \in \{2, \dots, 16\}$

2. Alice berechnet $c_1 = 6 \cdot g = 6 \cdot (6, 2) = (12, 9)$.

3. Alice berechnet $6 \cdot v = 6 \cdot (5, 4) = (12, 10)$ und

$$c_2 = (14, 6) + 6 \cdot (5, 4) = (14, 6) + (12, 10) = (16, 17)$$

4. Alice schickt $c = (c_1, c_2) = ((12, 9), (16, 17))$ an Bob.

Bob entschlüsselt die empfangene Nachricht $c = ((12, 9), (16, 17))$ wie folgt:

1. Bob berechnet $d = u \cdot c_1 = 11 \cdot (12, 9) = (12, 10)$.

2. Bob berechnet $-d = (12, -10) = (12, 9)$.

3. Bob berechnet

$$m = c_2 + (-d) = (16, 17) + (12, 9) = (14, 6)$$

Bob hat die Nachricht korrekt entschlüsselt.

Beispiel 14.12. Wir betrachten die elliptische Kurve \bar{E} über \mathbb{F}_8 (mit $\alpha^3 = \alpha + 1$), die gegeben ist durch

$$F(X, Y) = Y^2 + YX + X^3 + X^2 + 1 \in \mathbb{F}_8[X, Y]$$

(die wir schon in Beispiel 14.4 untersucht haben). Die Kurve (und Gruppe) \bar{E} besteht aus 14 Elementen, vergleiche auch dazu Beispiel 14.4,

$$\begin{aligned} \bar{E} = & \{(0, 1), (\alpha, \alpha^2 + \alpha + 1), (\alpha, \alpha^2 + 1), (\alpha + 1,), \\ & (\alpha + 1, \alpha + 1), (\alpha^2, \alpha + 1), (\alpha^2, \alpha^2 + \alpha + 1), (\alpha^2 + 1, 0), \\ & (\alpha^2 + 1, \alpha^2 + 1), (\alpha^2 + \alpha, \alpha + 1), (\alpha^2 + \alpha, \alpha + 1), \\ & (\alpha^2 + \alpha + 1, 0), (\alpha^2 + \alpha + 1), \infty\} \end{aligned}$$

und $g = (\alpha + 1, \alpha + 1)$ ist ein Element von \bar{E} von der Ordnung 7, erzeugt also eine Untergruppe $U \subseteq \bar{E}$ von der Primzahlordnung $r = 7$.

Bob wählt $p = 2$, $l = 3$ und diese Kurve \bar{E} über \mathbb{F}_8 sowie den Punkt $g = (\alpha + 1, \alpha + 1)$ der Ordnung 7 und die von ihm erzeugte Untergruppe U . Ferner wählt er $u = 5$ und berechnet

$$v = 5 \cdot g = 5 \cdot (\alpha + 1, \alpha + 1) = (\alpha^2 + \alpha + 1, 0)$$

Beachte dabei, dass $5 = 2^2 + 2^0$, also $\tau = (1, 0, 1)$.

$l = 2$: Setze $h_2 = g = (\alpha + 1, \alpha + 1)$.

$l = 1$: Berechne $\tilde{h}_1 = g + g = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1)$, da

$$m = r_1 + \frac{s_1}{r_1} = \alpha + 1 + \frac{\alpha + 1}{\alpha + 1} = \alpha$$

und damit ist $g + g = (\text{sum}_1, \text{sum}_2)$ mit

$$\text{sum}_1 = m^2 + m + a = \alpha^6 + \alpha^3 + 1 = \alpha^2 + \alpha + 1$$

und

$$\begin{aligned} \text{sum}_2 &= m \cdot (r_1 + \text{sum}_1) + \text{sum}_1 + s_1 = \alpha \cdot \alpha^2 + \alpha^2 + \alpha + 1 + \alpha + 1 \\ &= \alpha^2 + \alpha + 1 \end{aligned}$$

Da $\tau_1 = 0$, ist $h_1 = \tilde{h}_1 = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1)$.

$l = 0$: Berechne $\tilde{h}_0 = h_1 + h_1 = (\alpha^2 + 1, \alpha^2 + 1)$, da

$$m = \alpha^2 + \alpha + 1 + \frac{\alpha^2 + \alpha + 1}{\alpha^2 + \alpha + 1} = \alpha^2 + \alpha$$

und damit

$$\text{sum}_1 = \alpha^8 + \alpha^4 + 1 = \alpha^2 + 1, \quad \text{sum}_2 = \alpha^4 \cdot \alpha + \alpha^6 + \alpha^5 = \alpha^2 + 1$$

Da $\tau_0 = 1$ ist

$$h_0 = \tilde{h}_0 + g = (\alpha^2 + \alpha + 1, 0)$$

denn

$$m = \frac{s_1 + s_2}{r_1 + r_2} = \frac{\alpha^2 + \alpha}{\alpha^2 + \alpha} = 1$$

also

$$\begin{aligned} \text{sum}_1 &= m^2 + m + a + r_1 + r_2 = \alpha^2 + \alpha + 1, \\ \text{sum}_2 &= m \cdot (r_1 + \text{sum}_1) + \text{sum}_1 + s_1 = \alpha + \alpha^2 + \alpha + 1 + \alpha^2 + 1 = 0 \end{aligned}$$

Damit ist also $v = (\alpha^2 + \alpha + 1, 0)$.

1. Bobs privater Schlüssel ist $k_{\text{pr}} = (8, \overline{E}, (\alpha + 1, \alpha + 1), 5)$.
2. Bobs öffentlicher Schlüssel ist $k_{\text{pr}} = (8, \overline{E}, (\alpha + 1, \alpha + 1), (\alpha^2 + \alpha + 1, 0))$.

Alice will die Nachricht $m = (\alpha^2 + 1, 0)$ geheim an Bob schicken (Beachten Sie dabei, dass $m = 3 \cdot g \in U$).

1. Alice wählt die Zahl $t = 4 \in \{2, \dots, 7\}$

2. Alice berechnet $c_1 = 4 \cdot g = 4 \cdot (\alpha + 1, \alpha + 1) = (\alpha^2 + 1, \alpha^2 + 1)$.

3. Alice berechnet $4 \cdot v = 4 \cdot (\alpha^2 + \alpha + 1, 0) = (\alpha + 1, 0)$ und

$$c_2 = (\alpha^2 + 1, 0) + 4 \cdot (\alpha^2 + \alpha + 1, 0) = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1)$$

(beachte dabei, dass $m = \frac{s_1+s_2}{r_1+r_2} = 0$, also

$$\text{sum}_1 = a + r_1 + r_2 = \alpha^2 + \alpha + 1, \quad \text{sum}_2 = \text{sum}_1 + s_1 = \alpha^2 + \alpha + 1$$

4. Alice schickt $c = (c_1, c_2) = ((\alpha^2 + 1, \alpha^2 + 1), (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1))$ an Bob.

Bob entschlüsselt die empfangene Nachricht $c = ((\alpha^2 + 1, \alpha^2 + 1), (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1))$ wie folgt:

1. Bob berechnet $d = u \cdot c_1 = 5 \cdot (\alpha^2 + 1, \alpha^2 + 1) = (\alpha + 1, 0)$.

2. Bob berechnet $-d = (\alpha + 1, 0 + \alpha + 1) = (\alpha + 1, \alpha + 1)$.

3. Bob berechnet

$$m = c_2 + (-d) = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1) + (\alpha + 1, \alpha + 1) = (\alpha^2 + 1, 0)$$

Bob hat die Nachricht korrekt entschlüsselt.

Beispiel 14.13. Wir betrachten wieder die elliptische Kurve \overline{E} über \mathbb{F}_8 (mit $\alpha^3 = \alpha + 1$), die gegeben ist durch

$$F(X, Y) = Y^2 + YX + X^3 + X^2 + 1 \in \mathbb{F}_8[X, Y]$$

Die Kurve (und Gruppe) \overline{E} besteht aus 14 Elementen, vergleiche auch dazu Beispiel 14.4 bzw. Beispiel 14.12, und $g = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1)$ ist ein Element von \overline{E} von der Ordnung 7, erzeugt also eine Untergruppe $U \subseteq \overline{E}$ von der Primzahlordnung $r = 7$.

Bob wählt $p = 2$, $l = 3$ und diese Kurve \overline{E} über \mathbb{F}_8 sowie den Punkt $g = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1)$ der Ordnung 7 und die von ihm erzeugte Untergruppe U . Ferner wählt er $u = 3$ und berechnet

$$v = 3 \cdot g = 5 \cdot (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1) = (\alpha + 1, 0)$$

1. Bobs privater Schlüssel ist $k_{\text{pr}} = (8, \overline{E}, (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1), 3)$.

2. Bobs öffentlicher Schlüssel ist $k_{\text{pub}} = (8, \overline{E}, (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1), (\alpha + 1, 0))$.

Alice will die Nachricht $m = (\alpha^2 + 1, \alpha^2 + 1)$ geheim an Bob schicken (Beachten Sie dabei, dass $m = 3 \cdot g \in U$).

1. Alice wählt die Zahl $t = 5 \in \{2, \dots, 7\}$
2. Alice berechnet $c_1 = 5 \cdot g = 5 \cdot (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1) = (\alpha^2 + 1, 0)$.
3. Alice berechnet $5 \cdot v = 5 \cdot (\alpha + 1, 0) = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1)$ und

$$c_2 = (\alpha^2 + 1, \alpha^2 + 1) + 5 \cdot (\alpha + 1, 0) = (\alpha + 1, 0)$$
4. Alice schickt $c = (c_1, c_2) = ((\alpha^2 + 1, 0), (\alpha + 1, 0))$ an Bob.

Bob entschlüsselt die empfangene Nachricht $c = ((\alpha^2 + 1, 0), (\alpha + 1, 0))$ wie folgt:

1. Bob berechnet $d = u \cdot c_1 = 3 \cdot (\alpha^2 + 1, 0) = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1)$.
2. Bob berechnet $-d = (\alpha^2 + \alpha + 1, \alpha^2 + \alpha + 1 + \alpha^2 + \alpha + 1) = (\alpha^2 + \alpha + 1, 0)$.
3. Bob berechnet

$$m = c_2 + (-d) = (\alpha + 1, 0) + (\alpha^2 + \alpha + 1, 0) = (\alpha^2 + 1, \alpha^2 + 1)$$

Bob hat die Nachricht korrekt entschlüsselt.

Bob empfängt außerdem von Anton die Nachricht

$$c = (c_1, c_2) = ((\alpha^2 + \alpha + 1, 0), (\alpha^2 + \alpha + 1, 0))$$

und entschlüsselt diese wie folgt:

1. Bob berechnet $d = u \cdot c_1 = 3 \cdot (\alpha^2 + \alpha + 1, 0) = (\alpha + 1, \alpha + 1)$.
2. Bob berechnet $-d = (\alpha + 1, \alpha + 1 + \alpha + 1) = (\alpha + 1, 0)$.
3. Bob berechnet

$$m = c_2 + (-d) = (\alpha^2 + \alpha + 1, 0) + (\alpha + 1, 0) = (\alpha^2 + 1, \alpha^2 + 1)$$

Bob hat also ein anderes Chiffraum erhalten, obwohl ihm auch von Anton die Nachricht $(\alpha^2 + 1, \alpha^2 + 1)$ übermittelt wurde.

14.4. ECDSA - Digitale Signatur mit elliptischen Kurven

Elliptische Kurven können auch benutzt werden, um ein verlässliches Signaturverfahren zu entwickeln. Alice kann damit eine Nachricht unterschreiben, damit Bob überprüfen kann, ob diese Nachricht tatsächlich von Alice stammt.

Vorbereitung durch Alice:

Alice wählt für sich die folgenden Daten:

1. Eine Primzahl $p \neq 3$.
2. Ein $l > 1$ falls $p = 2$ bzw. $l = 1$ falls $p > 3$ und $q = p^l$.
3. Eine elliptische Kurve \bar{E} über \mathbb{F}_q , gegeben durch

$$F(X, Y) = Y^2 - X^3 - aX - b \in \mathbb{F}_q[X, Y]$$

falls $p > 3$ bzw.

$$F(X, Y) = Y^2 + XY + X^3 + aX^2 + b \in \mathbb{F}_q[X, Y]$$

falls $p = 2$.

4. Einen Punkt $g = (u, v) \in E$ der Ordnung $r = \text{ord}(g)$, wobei r eine Primzahl ist und die von g erzeugte Untergruppe $U = U_g = \langle g \rangle \subseteq \bar{E}$.

Schlüsselerzeugung:

Alice erzeugt ein Schlüsselpaar $(k_{\text{pr},A}, k_{\text{pub},A})$ wie folgt:

1. Alice wählt zufällig eine Zahl $u \in \{2, \dots, r - 1\}$.
2. Alice berechnet $v = u \cdot g$ in $U_g \subseteq \bar{E}$.
3. Der private Schlüssel von Alice ist $k_{\text{pr},A} = (q, \bar{E}, g, u)$.
4. Der öffentliche Schlüssel von Alice ist $k_{\text{pub},A} = (q, \bar{E}, g, v)$.
5. Alice veröffentlicht $k_{\text{pub},A}$.

Signatur durch Alice:

Alice will eine Nachricht m , die sie an Bob schickt, signieren, damit Bob sicher sein kann, dass sie von ihr ist. Dazu gehen wir davon aus, dass die Nachricht als Binärtupel vorliegt, $m = \mathbb{F}_2^*$. Üblicherweise wird für die Signatur ein verkürzter Hashwert der Nachricht (vergleiche Abschnitt 9 über Hashfunktionen) zugrundegelegt. Wir gehen hier davon aus, dass mit $t = \lfloor \log_2(r) \rfloor + 1$ die Nachricht bzw. ihr Hashwert bereits als binäres t -Tupel $m = (b_1, \dots, b_t) \in \mathbb{F}_2^t$ vorliegt. Dann geht Alice vor wie folgt.

1. Alice wandelt m via $m = \sum_{i=1}^t 2^{n-i} b_i$ in eine ganze Zahl m um.
2. Alice wählt zufällig ein $z \in \{2, \dots, r - 1\}$.
3. Alice berechnet $z \cdot g$ in $U_g \subseteq \overline{E}$ und schreibt $z \cdot g = (x_A, y_A) \in \mathbb{F}_q^2$.
4. Alice wandelt x_A , die x -Komponente von $z \cdot g \in \overline{E}$, in eine ganze Zahl um, und zwar wie folgt:
 - Falls q eine Primzahl ist, so identifiziert Alice \mathbb{F}_q mit $\mathbb{F}_q = \{0, 1, \dots, q - 1\}$

und nimmt

$$n = x_A \bmod r$$

- Falls $q = 2^l$, so schreibt Alice $x_A = (a_1, \dots, a_l) \in \mathbb{F}_2^l$ und nimmt

$$n = \sum_{i=1}^l 2^{l-i} \cdot a_i \bmod r$$

5. Falls $n = 0$, so geht Alice zurück zu Schritt (2).
 6. Alice berechnet $z^{-1} \in \mathbb{F}_r$ und setzt
- $$s = z^{-1} \cdot (m + n \cdot u) \bmod r$$
- (dh. sie betrachte alle Zahlen als Restklassen in \mathbb{F}_r).
7. Falls $s = 0$, so geht Alice zurück zu Schritt (2).
 8. Alice signiert ihre Nachricht mit $\text{sig}_A(m) = (n, s)$ und schickt $c = (m, \text{sig}_A(m))$ an Bob.

Bemerkung 14.3. Die Zahl m kann größer sein als r , sie hat aber die gleiche binäre Größenordnung wie r .

Verifikation durch Bob:

Bob empfängt die Nachricht $c = (m, (n, s))$ und überprüft, ob es sich bei (n, s) und die Signatur $\text{sig}_A(m)$ von Alice handelt wie folgt:

1. Bob überprüft ob $1 \leq n \leq r - 1$ und $1 \leq s \leq r - 1$. Ist das nicht der Fall, so lehnt er die Signatur ab.
2. Bob berechnet $s^{-1} \in \mathbb{F}_r$ und setzt

$$w_1 = m \cdot s^{-1} \pmod{r}, \quad w_2 = n \cdot s^{-1} \pmod{r}$$

und

$$R = w_1 \cdot g + w_2 \cdot v \quad (\text{in } \overline{E})$$

(wobei m die Nachricht ist und v aus dem öffentlichen Schlüssel $k_{\text{pub}, A}$ von Alice stammt).

3. Ist $R = \infty$, so lehnt Bob die Signatur ab.
4. Bob schreibt $R = (x_B, y_B) \in \mathbb{F}_q^2$ und wandelt x_B (wie oben beschrieben) in eine ganze Zahl $\tilde{n} \in \{0, \dots, r - 1\}$ um.
5. Bob akzeptiert die Signatur, falls $n = \tilde{n} \pmod{r}$, andernfalls lehnt er sie ab.

Hilfssatz 14.4. *Hat Alice die Nachricht korrekt mit $\text{sig}_A(m) = (n, s)$ signiert, so gilt*

$$n = \tilde{n} \pmod{r}$$

Beweis: Es ist $s = z^{-1} \cdot (m + n \cdot u)$ (in \mathbb{F}_r). Damit gilt

$$\begin{aligned} z &= s^{-1} \cdot (m + n \cdot u) \\ &= s^{-1} \cdot m + s^{-1} \cdot n \cdot u \\ &= m \cdot s^{-1} + n \cdot s^{-1} \cdot u \\ &= w_1 + w_2 \cdot u \end{aligned}$$

also

$$\begin{aligned} z \cdot g &= w_1 \cdot g + w_2 \cdot u \cdot g \\ &= w_1 \cdot g + w_2 \cdot v \\ &= R \end{aligned}$$

Daraus folgt

$$(x_A, y_A) = (x_B, y_B)$$

und damit natürlich auch

$$n = \tilde{n} \pmod{r}$$

Bemerkung 14.4. Falls die Signatur nicht korrekt ist, ist es immer noch möglich, dass $n = \tilde{n} \bmod r$, die Wahrscheinlichkeit dafür ist aber (bei hinreichend großem r) vernachlässigbar gering.

Bemerkung 14.5. Die Zahl z , die Alice zufällig für Ihre Signatur wählt, muss geheim bleiben. Andernfalls kann bei bekanntem m (also einem plaintext–ciphertext–Angriff) der private Schlüssel u von Alice bestimmt werden:

Es gilt nämlich $s = z^{-1} \cdot (m + n \cdot u)$, und damit

$$m + n \cdot u = z \cdot s$$

Daraus folgt

$$n \cdot u = z \cdot s - m$$

also

$$u = n^{-1} \cdot (z \cdot s - m)$$

Bemerkung 14.6. Die Zahl z , die Alice zufällig für Ihre Signatur wählt, darf nur einmal verwendet werden. Andernfalls kann bei zwei bekannten Nachrichten m_1 und m_2 (also wieder einem plaintext–ciphertext–Angriff, diesmal mit zwei Textpaaren) der private Schlüssel u von Alice bestimmt werden.

Für die Signaturen (n_1, s_1) und (n_2, s_2) gilt nämlich in diesem Fall $n_1 = n_2$ (beidemal ist es die x -Komponente von $g \cdot z$). Hieraus kann Catherine schon (mit sehr hoher Wahrscheinlichkeit) ableiten, dass Alice z wiederverwendet hat. Setzt sie $n := n_1 = n_2$, so erhält sie

$$s_1 = z^{-1} \cdot (m_1 + n \cdot u), \quad s_2 = z^{-1} \cdot (m_2 + n \cdot u)$$

Es folgt

$$s_1^{-1} \cdot (m_1 + n \cdot u) = z = s_2^{-1} \cdot (m_2 + n \cdot u)$$

also (nach Multiplikation mit $s_1 \cdot s_2$)

$$s_2 \cdot m_1 + s_2 \cdot n \cdot u = s_1 \cdot m_2 + s_1 \cdot n \cdot u$$

bzw.

$$s_2 \cdot n \cdot u - s_1 \cdot n \cdot u = s_1 \cdot m_2 - s_2 \cdot m_1$$

bzw.

$$u \cdot (s_2 \cdot n - s_1 \cdot n) = s_1 \cdot m_2 - s_2 \cdot m_1$$

und damit

$$u = \frac{s_1 \cdot m_2 - s_2 \cdot m_1}{(s_2 - s_1) \cdot n}$$

Beispiel 14.14. Wir betrachten wieder die elliptische Kurve \overline{E} über \mathbb{F}_{17} die gegeben ist durch

$$F(X, Y) = Y^2 - X^3 - 15X - 8 = Y^2 + 16X^3 + 2X + 9 \in \mathbb{F}_{17}[X, Y]$$

Diese besteht aus den Punkten

$$\begin{aligned} \overline{E} = & \{(0, 5), (0, 12), (4, 7), (4, 10), (5, 2), (5, 15), (6, 5), \\ & (6, 12), (10, 6), (10, 11), (11, 5), (11, 12), (14, 2), \\ & (14, 15), (15, 2), (15, 15), (16, 3), (16, 14), \infty\} \end{aligned}$$

Insbesondere gilt also $|\overline{E}| = 19$. Damit ist \overline{E} zyklisch von Primzahlordnung $r = 19$ und jedes Element $g \in E$ (also jedes $g \in \overline{E} \setminus \{\infty\}$) ist ein Erzeuger.

In diesem Fall ist $t = \lfloor \log_2(r) \rfloor + 1 = 5$.

Alice wählt $g = (4, 8)$ als Erzeuger (und damit auf $U_g = \overline{E}$) und $u = 11$ als privaten Schlüssel. Alice berechnet

$$v = 11 \cdot g = (6, 5)$$

1. Alice hat den privaten Schlüssel $k_{\text{pr}, A} = (17, \overline{E}, (4, 8), 11)$
2. Alice hat den öffentlichen Schlüssel $k_{\text{pub}, A} = (17, \overline{E}, (4, 8), (6, 5))$

Signatur durch Alice:

Alice will $m = (1, 0, 1, 1, 1)$ signieren.

1. Alice berechnet $m = 2^4 + 2^2 + 2^1 + 2^0 = 23$.
2. Alice wählt zufällig $z = 7$ (aus $\{0, 1, \dots, 18\} = \mathbb{F}_r$).
3. Alice berechnet $z \cdot g = 7 \cdot (4, 8) = (5, 15) = (x_A, y_A)$.
4. Alice setzt $n = x_A = 5$ (in \mathbb{F}_{19}).
5. Alice berechnet $z^{-1} = 11 \in \mathbb{F}_{19}$ und setzt

$$s = z^{-1} \cdot (m + n \cdot u) = 11 \cdot (23 + 5 \cdot 11) = 3 \quad \text{in } \mathbb{F}_{19}$$

6. Alice signiert ihre Nachricht mit $\text{sig}_A(m) = (5, 3)$ und schickt

$$c = ((1, 0, 1, 1, 1), (5, 3))$$

an Bob.

Signaturüberprüfung durch Bob:

Bob empfängt die Nachricht $c = ((1, 0, 1, 1, 1), (5, 3))$ und überprüft die Signatur $\text{sig}_A(m) = (5, 3)$ wie folgt:

1. Er prüft, dass $1 \leq 5 \leq 18$ und $1 \leq 3 \leq 18$.
2. Bob berechnet $s^{-1} = 3^{-1} = 13 \in \mathbb{F}_{19}$ und setzt

$$w_1 = m \cdot s^{-1} = 23 \cdot 13 = 14 \pmod{19}, \quad w_2 = n \cdot s^{-1} = 5 \cdot 13 = 8 \pmod{19}$$

und

$$R = w_1 \cdot g + w_2 \cdot v = 14 \cdot (4, 8) + 8 \cdot (6, 5) = (16, 3) + (5, 2) = (11, 5) \quad (\text{in } \overline{E})$$

3. Bob schreibt $R = (x_B, y_B) = (5, 15) \in \mathbb{F}_{17}^2$ und setzt

$$\tilde{n} = x_B = 5$$

4. Bob akzeptiert die Signatur, da $n = 5 = \tilde{n} \pmod{19}$.

Beispiel 14.15. Wir betrachten erneut die elliptische Kurve \overline{E} über \mathbb{F}_{17} die gegeben ist durch

$$F(X, Y) = Y^2 - X^3 - 15X - 8 = Y^2 + 16X^3 + 2X + 9 \in \mathbb{F}_{17}[X, Y]$$

Alice wählt wieder $g = (4, 8)$ als Erzeuger und $u = 11$ als privaten Schlüssel, sodass wieder gilt

1. Alice hat den privaten Schlüssel $k_{\text{pr}, A} = (17, \overline{E}, (4, 8), 11)$
2. Alice hat den öffentlichen Schlüssel $k_{\text{pub}, A} = (17, \overline{E}, (4, 8), (6, 5))$

Signatur durch Alice:

Alice will $m = (0, 1, 1, 1, 0)$ signieren.

1. Alice berechnet $m = 2^3 + 2^2 + 2^1 = 14$.
2. Alice wählt zufällig $z = 4$.
3. Alice berechnet $z \cdot g = 4 \cdot (4, 8) = (15, 2) = (x_A, y_A)$.
4. Alice setzt $n = x_A = 15$ (in \mathbb{F}_{19}).

5. Alice berechnet $z^{-1} = 4^{-1} = 5 \in \mathbb{F}_{19}$ und setzt

$$s = z^{-1} \cdot (m + n \cdot u) = 5 \cdot (14 + 15 \cdot 11) = 2 \quad \text{in } \mathbb{F}_{19}$$

6. Alice signiert ihre Nachricht mit $\text{sig}_A(m) = (15, 2)$ und schickt

$$c = ((0, 1, 1, 1, 0), (15, 2))$$

an Bob.

Signaturüberprüfung durch Bob:

Bob empfängt die Nachricht $c = ((0, 1, 1, 1, 0), (15, 2))$ und überprüft die Signatur $\text{sig}_A(m) = (15, 2)$ wie folgt:

1. Er prüft, dass $1 \leq 15 \leq 18$ und $1 \leq 2 \leq 18$.

2. Bob berechnet $s^{-1} = 2^{-1} = 10 \in \mathbb{F}_{19}$ und setzt

$$w_1 = m \cdot s^{-1} = 14 \cdot 10 = 7 \pmod{19}, \quad w_2 = 15 \cdot 10 = 17 \pmod{19}$$

und

$$R = w_1 \cdot g + w_2 \cdot v = 7 \cdot (4, 8) + 17 \cdot (6, 5) = (5, 15) + (14, 15) = (15, 2) \quad (\text{in } \overline{E})$$

3. Bob schreibt $R = (x_B, y_B) = (15, 2) \in \mathbb{F}_{17}^2$ und setzt

$$\tilde{n} = x_B = 15$$

4. Bob akzeptiert die Signatur, da $n = 15 = \tilde{n} \pmod{19}$.

Wir betrachten nun wieder einen Körper \mathbb{F}_q der Charakteristik p , wobei entweder $q = p > 3$ eine Primzahl oder $p = 2$ und $q = 2^l$ ist und eine elliptische Kurve \overline{E} über \mathbb{F}_q , gegeben durch

$$F(X, Y) = Y^2 - X^3 - aX - b \quad \in \mathbb{F}_q[X, Y]$$

falls $p > 3$ bzw.

$$F(X, Y) = Y^2 + XY + X^3 + aX^2 + b \quad \in \mathbb{F}_q[X, Y]$$

falls $p = 2$.

Definition 14.1. Die Kurve \overline{E} hat **fast–Primzahlordnung**, wenn $|\overline{E}| = t \cdot r$, wobei r eine Primzahl ist und der Kofaktor t sehr klein im Vergleich zu r ist.

Bemerkung 14.7. In allen praktischen Anwendungen werden elliptische Kurven von fast–Primzahlordnung so gewählt, dass $t \in \{1, 2, 3, 4\}$. Solche elliptische Kurven treten (für große q) relativ häufig auf und werden in der Regel durch pseudorandomisiertes Suchen gefunden.

Bemerkung 14.8. Hat \overline{E} fast Primzahlordnung, $|\overline{E}| = t \cdot r$, so hat \overline{E} eine eindeutig bestimmte Untergruppe $E_r \subseteq \overline{E}$ der Ordnung r . Diese heißt Primordnungsuntergruppe von \overline{E} .

Bemerkung 14.9. Hat \overline{E} fast–Primzahlordnung, $|\overline{E}| = t \cdot r$, so liegt die binäre Länge $\lfloor \log_2(r) \rfloor + 1$ der Primordnungsuntergruppe E_r nahe bei $\lfloor \log_2(q) \rfloor + 1$, der binären Größe des Grundkörpers.

14.5. Unsichere elliptische Kurven

Das Diskrete Logarithmusproblem auf elliptischen Kurven ist sehr schwer. Allerdings gibt es einige wenige Klassen von Kurven, für die Angriffe bekannt sind, die die Komplexität der Berechnungen signifikant reduzieren und das Problem auf einfachere Probleme zurückführen:

MOV–Angriff (Menezes, Okamoto, Vanstone, 1993):

Mithilfe der sogenannten Weil–Paarung ist es möglich, Primordnungsuntergruppen $E_r \subseteq \overline{E}$ über \mathbb{F}_q mit einer Untergruppe der multiplikativen Gruppe $\mathbb{F}_{q^n}^*$ für ein geeignetes $n \geq 1$ zu identifizieren. Für $\mathbb{F}_{q^n}^*$ kann das diskrete Logarithmusproblem in subexponentieller Zeit gelöst werden.

Damit $E_r \subseteq \mathbb{F}_{q^n}^*$ eingebettet werden kann, muss $|E_r|$ ein Teiler von $|\mathbb{F}_{q^n}^*| = q^n - 1$ sein, also $r|(q^n - 1)$ gelten. Daher ist dieser Angriff nicht effektiv, wenn $r \nmid (q^n - 1)$ für kleine n . Üblicherweise reicht es, alle $n \leq 20$ auszuschließen.

Frey–Rück–Angriff (Frey, Rück, 1994):

Der Frey–Rückangriff funktioniert ähnlich wie der MOV–Angriff, nutzt aber die Tate–Paarung anstelle der Weil–Paarung aus. Die Abwehr gegen den MOV–Angriff wehrt auch den Frey–Rückangriff ab.

Elliptische Kurven spezieller Ordnung:

Ein elliptische Kurve $\overline{E}/\mathbb{F}_p$ heißt **Primkörper–anomal**, wenn $|\overline{E}| = p$ ($= |\mathbb{F}_p|$). In diesem Fall gibt es Algorithmen in polynomialer Laufzeit, die \overline{E} mit $(\mathbb{F}_p, +)$ identifizieren, und in $(\mathbb{F}_p, +)$ ist das diskrete Logarithmusproblem einfach. Solche Algorithmen wurden gefunden von Semaev (1998), Smart (1999) und Satoh und Araki (1998).

14.6. Empfohlene elliptische Kurven

Vom National Institute of Standards and Technology (NIST) gibt es Empfehlungen und Richtlinien zur Wahl elliptischer Kurven für kryptographische Zwecke. Für elliptische Kurven \overline{E} über einem Primkörper \mathbb{F}_p wird empfohlen, dass

$$\lfloor \log_2(p) \rfloor + 1 \in \{193, 224, 256, 384, 521\}$$

Für elliptische Kurven \overline{E} über \mathbb{F}_q mit $q = 2^l$ soll gelten

$$l \in \{163, 233, 239, 283, 409, 571\}$$

Für jede dieser Größen werden auch Beispielkurven angegeben, die die Anforderungen von NIST erfüllen. Für Kurven über Primkörpern \mathbb{F}_p sind diese immer gegeben durch

$$F(X, Y) = Y^2 - X^3 + 3X - b \in \mathbb{F}_q[X, Y]$$

etwa die Kurve P-192 mit

$$\begin{aligned} p &= 2^{192} - 2^{64} - 1 \\ &= 6\,277\,101\,735\,386\,680\,763\,835\,789\,423\,207\,666\,416\,083\,908\,700\,390\,324\,961\,279 \end{aligned}$$

und (in Hexadezimaldarstellung)

$$b = 0x\,64210519\,e59c80e7\,0fa7e9ab\,72243049\,feb8deec\,c146b961$$

In diesem Fall gilt

$$\overline{E} = E_r$$

dh. \overline{E} ist von Primzahlordnung und

$$r = 6\,277\,101\,735\,386\,680\,763\,835\,789\,423\,176\,059\,013\,767\,194\,773\,182\,842\,284\,081$$

Für $q = 2^l$ sind diese Kurven $\overline{E}/\mathbb{F}_q$ immer gegeben durch Gleichungen der Form

$$F(X, Y) = Y^2 + XY + X^3 + X^2 + b \in \mathbb{F}_q[X, Y]$$

etwa für $l = 163$ und $\mathbb{F}_{2^{163}}$, gegeben durch

$$\alpha^{163} = \alpha^7 + \alpha^6 + \alpha^3 + 1$$

die Kurve B-163 mit

$$b = 0x\,00000002\,0a601907\,68c953ca\,1481eb10\,512f\,7874\,4a3205fd$$

(in Hexadezimaldarstellung). Für diese Kurve gilt

$$\overline{E} = \mathbb{Z}_2 \times \mathbb{Z}_r$$

wobei

$$r = 5\,846\,006\,549\,323\,611\,672\,814\,572\,442\,876\,390\,689\,256\,843\,201\,587$$

15. Gitter–basierte Kryptosysteme

Die Technik der Quantencomputer gefährdet viele der klassischen kryptologischen Verfahren. Die Quantentechnologie ermöglicht Faktorisierungsalgorithmen mit polynomia-ler Laufzeit und auch Ansätze zur Lösung des diskreten Logarithmus–Problems in \mathbb{F}_q^* , wodurch die darauf basierenden public–key–Verfahren angreifbar werden. Noch ist das Zukunftsmusik und die aktuellen Quantencomputer sind noch weit von der Rechenleis-tung, die erforderlich sind, um große Zahlen zu faktorisieren, entfernt, aber das kann sich innerhalb des nächsten Jahrzehnts ändern. Daher ist es wichtig, jetzt schon Ansätze zu entwickeln, die auch der Quantentechnologie standhalten. Verfahren, die das nach aktuellem Wissensstand erfüllen, sind kryptographische Verschlüsselungsverfahren, die auf Gittern basieren.

15.1. Gitter

Definition 15.1. Ein n –dimensionales **Gitter** ist eine Teilmenge $\mathcal{L} \subseteq \mathbb{R}^n$ mit der fol-genden Eigenschaft:

Es gibt eine Basis $\{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_n\}$ von \mathbb{R}^n , sodass

$$\mathcal{L} = \{\vec{v} = \sum_{i=1}^n z_i \cdot \vec{b}_i \mid z_i \in \mathbb{Z}\}$$

Die Basis $\{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_n\}$ heißt **Basis** des Gitters \mathcal{L} .

Ist ein Gitter \mathcal{L} durch eine Basis $B = \{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_n\}$ von \mathbb{R}^n , so schreiben wir $\mathcal{L}(B)$ für dieses Gitter. Ferner benutzen wir auch die Notation

$$B = (\vec{b}_1 \vec{b}_2 \dots \vec{b}_n)$$

für den Matrix mit den Vektoren \vec{b}_i als Spalten.

Beispiel 15.1. Durch die Basis $E = \{\vec{e}_1, \vec{e}_2\}$ mit

$$\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \vec{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

wird ein 2–dimensionales Gitter $\mathcal{L}_1 = \mathbb{Z}^2 \subseteq \mathbb{R}^2$ definiert.

Beispiel 15.2. Durch die Basis $B = \{\vec{b}_1, \vec{b}_2\}$ mit

$$\vec{b}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \vec{b}_2 = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

wird ein 2–dimensionales Gitter \mathcal{L}_2 definiert. Das Gitter \mathcal{L}_2 ist ein Untergitter des Gitters \mathcal{L}_1 aus Beispiel 15.1

Beispiel 15.3. Durch die Basis $B' = \{\vec{b}'_1, \vec{b}'_2\}$ mit

$$\vec{b}'_1 = \begin{pmatrix} 11 \\ 5 \end{pmatrix}, \quad \vec{b}'_2 = \begin{pmatrix} 6 \\ 3 \end{pmatrix}$$

wird ebenfalls das 2-dimensionale Gitter \mathcal{L}_2 definiert. Die Basis eines Gitters ist also nicht eindeutig bestimmt.

Es gilt nämlich

$$\vec{b}'_1 = 11 \cdot \vec{b}_1 - 2 \cdot \vec{b}_2, \quad \vec{b}'_2 = 6 \cdot \vec{b}_1 - \vec{b}_2$$

Damit sind \vec{b}'_1 und \vec{b}'_2 in dem von B erzeugten Gitter \mathcal{L}_2 , also offensichtlich auch das ganze von B' erzeugte Gitter $\mathcal{L}(B')$, womit $\mathcal{L}(B') \subseteq \mathcal{L}_2$ gezeigt ist. Umgekehrt ist aber auch

$$\vec{b}_1 = 2 \cdot \vec{b}'_1 - \vec{b}'_2, \quad \vec{b}_2 = (-6) \cdot \vec{b}'_1 + 11 \cdot \vec{b}'_2$$

Damit sind \vec{b}_1 und \vec{b}_2 in dem von B' erzeugten Gitter $\mathcal{L}(B')$, also offensichtlich auch das ganze von B erzeugte Gitter \mathcal{L}_2 , womit auch $\mathcal{L}_2 \subseteq \mathcal{L}(B')$ nachgerechnet ist. Also sind die beiden Gitter gleich.

Regel 15.1. Zwei Gitter $\mathcal{L}(B)$ und $\mathcal{L}(B')$ sind genau dann gleich, wenn es eine Matrix $U \in \text{Matr}(n \times n, \mathbb{Z})$ mit $\det(U) = \pm 1$ gibt mit

$$B' = B \cdot U$$

In diesem Fall ist dann U eine invertierbare Matrix mit $U^{-1} \in \text{Matr}(n \times n, \mathbb{Z})$ (da $\det(U) = \pm 1$) und $B = B' \cdot U^{-1}$.

Beweis: Da $\mathcal{L}(B) = \mathcal{L}(B')$ ist sicherlich $\vec{b}'_j \in \mathcal{L}(B)$ für alle j , also

$$\vec{b}'_j = u_{1,j} \cdot \vec{b}_1 + u_{2,j} \cdot \vec{b}_2 + \cdots + u_{n,j} \cdot \vec{b}_n$$

für alle $j \in \{1, \dots, n\}$ mit geeigneten $u_{i,j} \in \mathbb{Z}$. Mit

$$U = (u_{i,j})_{1 \leq i,j \leq n} \in \text{Matr}(n \times n, \mathbb{Z})$$

schreiben sich diese Beziehungen gerade in der Form

$$B' = B \cdot U$$

Entsprechend ist auch $\vec{b}_j \in \mathcal{L}(B')$, sodass wir auch eine analoge Beziehung

$$B = B' \cdot U'$$

für ein geeignetes mit $U' \in \text{Matr}(n \times n, \mathbb{Z})$ erhalten. Setzen wir die beidne Beziehungen zusammen, so ergibt sich

$$B' = B' \cdot U \cdot U'$$

Da die Spalten von B' eine Basis von \mathbb{R}^n bilden, ist B' eine invertierbare Matrix, und daher folgt daraus

$$E_n = U \cdot U'$$

also $U' = U^{-1}$. Da sowohl U^{-1} als auch U ganzzahlige Matrizen sind, muss notwendig auch $\det(U)$ und $\det(U^{-1})$ ganzzahlig sein, und das kann wegen

$$\det(U^{-1}) = \frac{1}{\det(U)}$$

nur dann der Fall sein, wenn $\det(U) = \pm 1$.

Definition 15.2. Ist $\mathcal{L} = \mathcal{L}(B)$ eine Gitter, gegeben durch $B = (\vec{b}_1 \dots \vec{b}_n)$ so heißt

$$\mathcal{F}(B) = \left\{ \vec{x} = \sum_{i=1}^n r_i \cdot \vec{b}_i \mid 0 \leq r_i < 1 \quad \text{für alle } i \right\}$$

die **Fundamentalmasche** des Gitter \mathcal{L} zur Basis B .

Bemerkung 15.1. Ist $\mathcal{L} = \mathcal{L}(B)$ eine Gitter und ist $\vec{v} \in \mathbb{R}^n$ beliebig, so gibt es eindeutig bestimmte Vektoren $\vec{g} \in \mathcal{L}$ und $\vec{x} \in \mathcal{F}(B)$ mit

$$\vec{v} = \vec{g} + \vec{x}$$

Da nämlich $\{\vec{b}_1, \dots, \vec{b}_n\}$ eine Basis von \mathbb{R}^n ist, können wir

$$\vec{v} = \sum_{i=1}^n v_i \cdot \vec{b}_i$$

mit geeigneten $v_i \in \mathbb{R}$ schreiben. Setzen wir dann $z_i = \lfloor v_i \rfloor$, die Abrundung von v_i , und $r_i = v_i - z_i$, so ist $0 \leq r_i < 1$ für alle i , also

$$\vec{g} = \sum_{i=1}^n z_i \cdot \vec{b}_i \in \mathcal{L}, \quad \vec{x} = \sum_{i=1}^n r_i \cdot \vec{b}_i \in \mathcal{F}(B)$$

und offensichtlich gilt

$$\vec{v} = \vec{g} + \vec{x}$$

Gäbe es noch eine zweite Darstellung $\vec{v} = \vec{g}' + \vec{x}'$ dieser Art, so müsste gelten

$$\vec{g} - \vec{g}' = \vec{x}' - \vec{x}$$

Nun gilt

$$\overrightarrow{g} - \overrightarrow{g}' = \sum_{i=1}^n (z_i - z'_i) \cdot \overrightarrow{b}_i$$

mit $z_i - z'_i \in \mathbb{Z}$ für alle i und

$$\overrightarrow{x} - \overrightarrow{x}' = \sum_{i=1}^n (r'_i - r_i) \cdot \overrightarrow{b}_i$$

mit $-1 < r'_i - r_i < 1$ für alle i . Daher kann diese Gleichung nur gelten, wenn

$$z_i - z'_i = 0, \quad r'_i - r_i = 0 \quad \text{für alle } i$$

also wenn die beiden Darstellungen gleich sind.

Definition 15.3. Ist $\mathcal{L} = \mathcal{L}(B)$ ein Gitter, so heißt

$$\det(\mathcal{L}) = |\det(B)|$$

die Determinante von \mathcal{L} .

Bemerkung 15.2. Die Basis B eines Gitters ist zwar nicht eindeutig bestimmt, trotzdem ist $\det(\mathcal{L})$ unabhängig von der Wahl der Basis B von \mathcal{L} . Ist nämlich B' eine weitere Basis von \mathcal{L} , so gilt nach Regel 15.1, dass $B' = B \cdot U$ mit einer invertierbaren ganzzahligen Matrix U . Damit gilt

$$\det(B') = \det(B \cdot U) = \det(B) \cdot \det(U)$$

Da aber $\det(U) = \pm 1$, ist

$$|\det(B')| = |\det(B)|$$

und die Definition von $\det(\mathcal{L})$ ist unabhängig von der Wahl einer Basis von \mathcal{L} .

Mit

$$\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$$

bezeichnen wir das euklidische Skalarprodukt von \mathbb{R}_n , dh.

$$\langle \overrightarrow{v}, \overrightarrow{w} \rangle = \sum_{i=1}^n v_i \cdot w_i$$

Definition 15.4. Ist $\mathcal{L} \subseteq \mathbb{R}^n$ ein n -dimensionales Gitter, so heißt

$$\mathcal{L}^\vee = \{ \overrightarrow{v} \in \mathbb{R}^n \mid \langle \overrightarrow{v}, \overrightarrow{w} \rangle \in \mathbb{Z} \quad \text{für alle } \overrightarrow{w} \in \mathcal{L} \}$$

das zu \mathcal{L} **duale Gitter**.

Regel 15.2.

a) Ist $\mathcal{L} = \mathcal{L}(B)$, so gilt

$$\mathcal{L}^\vee = \mathcal{L}\left(\left(B^{-1}\right)^\top\right)$$

b) Es gilt

$$\det(\mathcal{L}^\vee) = \frac{1}{\det(\mathcal{L})}$$

Beweis: Da $\mathcal{L} = \mathcal{L}(B)$, sind die Vektoren \vec{w} in \mathcal{L} gerade die Vektoren der Gestalt

$$\vec{w} = B \cdot \vec{z}$$

mit einem $\vec{z} \in \mathbb{Z}^n$. Da B eine Basis von \mathbb{R}^n ist, definiert auch $(B^{-1})^\top$ eine Basis von \mathbb{R}^n , und daher schreibt sich jedes $\vec{w} \in \mathbb{R}^n$ als

$$\vec{w} = \left(B^{-1}\right)^\top \cdot \vec{a}$$

mit einem geeigneten $\vec{a} \in \mathbb{R}^n$. Wenn wir nun zeigen

$$\left\langle \left(B^{-1}\right)^\top \cdot \vec{a}, B \cdot \vec{z} \right\rangle \in \mathbb{Z} \quad \text{für alle } \vec{z} \in \mathbb{Z}^n \iff \vec{a} \in \mathbb{Z}^n$$

so bedeutet das offensichtlich, dass \mathcal{L}^\vee ein Gitter ist und $(B^{-1})^\top$ eine Basis davon. Dazu benutzen wir die Regeln zu Rechnen mit Skalarprodukten, die besagen, dass

$$\begin{aligned} \left\langle \left(B^{-1}\right)^\top \cdot \vec{a}, B \cdot \vec{z} \right\rangle &= \left\langle \vec{a}, \left(\left(B^{-1}\right)^\top\right)^\top \cdot B \cdot \vec{z} \right\rangle = \left\langle \vec{a}, B^{-1} \cdot B \cdot \vec{z} \right\rangle \\ &= \langle \vec{a}, \vec{z} \rangle \end{aligned}$$

Damit ist zunächst klar:

Ist $\vec{a} \in \mathbb{Z}^n$, so ist $\left\langle \left(B^{-1}\right)^\top \cdot \vec{a}, B \cdot \vec{z} \right\rangle \in \mathbb{Z}$, für alle $\vec{z} \in \mathbb{Z}^n$ also ist \Rightarrow aus Teil a) gezeigt.

Gilt umgekehrt $\left\langle \left(B^{-1}\right)^\top \cdot \vec{a}, B \cdot \vec{z} \right\rangle \in \mathbb{Z}$ für jedes $\vec{z} \in \mathbb{Z}^n$, und betrachten wir speziell den Vektor $\vec{e}_i \in \mathbb{Z}^n$, so bedeutet das

$$\left\langle \left(B^{-1}\right)^\top \cdot \vec{a}, B \cdot \vec{e}_i \right\rangle \in \mathbb{Z}$$

also mit obiger Umformung

$$\langle \vec{a}, \vec{e}_i \rangle \in \mathbb{Z}$$

Da aber $\langle \vec{a}, \vec{e}_i \rangle = a_i$, heißt das also, dass $a_i \in \mathbb{Z}$, und da $i \in \{1, \dots, n\}$ beliebig war, bedeutet das, dass $\vec{a} \in \mathbb{Z}^n$. Damit ist auch die Richtung \Leftarrow aus Teil a) gezeigt.

Teil b) ist dann offensichtlich, denn

$$\det\left(\left(B^{-1}\right)^\top\right) = \det(B^{-1}) = \frac{1}{\det(B)}$$

In der Kryptographie werden häufig Gitter mit speziellen Eigenschaften betrachtet.

Definition 15.5. Ein n -dimensionales Gitter \mathcal{L} heißt q -Gitter für ein $q \in \mathbb{N}$, wenn

$$q \cdot \mathbb{Z}^n \subseteq \mathcal{L} \subseteq \mathbb{Z}^n$$

Beispiel 15.4. Das Gitter $\mathcal{L}(B)$ mit der Basis $B = \{\vec{b}_1, \vec{b}_2\}$ mit

$$\vec{b}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \vec{b}_2 = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

aus Beispiel 15.3 ist ein 3-Gitter. Offensichtlich ist nämlich $\mathcal{L} \subseteq \mathbb{Z}^2$. Ferner ist auch

$$\vec{b}_3 = \begin{pmatrix} 3 \\ 0 \end{pmatrix} = 3 \cdot \vec{b}_1 - \vec{b}_2 \in \mathcal{L}$$

und damit gilt für jeden Vektor

$$\vec{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \in \mathbb{Z}^2$$

dass

$$3 \cdot \vec{z} = z_1 \cdot \vec{b}_3 + z_2 \cdot \vec{b}_2 \in \mathcal{L}$$

Das Gitter \mathcal{L} ist aber kein 2-Gitter.

Wir identifizieren nun den Ring $Z_q = \mathbb{Z}/q \cdot \mathbb{Z}$ mit seinen Repräsentanten $\{0, 1, \dots, q-1\}$ in \mathbb{Z} (und schreiben dann auch $z \bmod q$ für diesen Repräsentanten) und für einen Vektor $\vec{z} \in \mathbb{Z}^n$ schreiben wir

$$\vec{z} \bmod q = \begin{pmatrix} z_1 \bmod q \\ z_2 \bmod q \\ \vdots \\ z_n \bmod q \end{pmatrix} \in \mathbb{Z}^n$$

Eine Übungsaufgabe in linearer Algebra und im modulo-Rechnen zeigt nun

Bemerkung 15.3.

a) Ist \mathcal{L} ein q -Gitter, so gilt für ein $\vec{v} \in \mathbb{Z}^n$:

$$\vec{v} \in \mathcal{L} \iff \vec{v} \bmod q \in \mathcal{L}$$

b) Jedes Gitter $\mathcal{L} \subseteq \mathbb{Z}^n$ ist ein q -Gitter für ein geeignetes $q \in \mathbb{N}$.

Solche q -Gitter können relativ einfach konstruiert werden. Dazu betrachten wir ein beliebige ganzzahlige $n \times m$ -Matix A , also $A \in \text{Matr}(n \times m, \mathbb{Z})$, wobei eigentlich nur die Koeffizienten modulo q von Interesse sind, dh. wir können die Matrix A auch also $A \in \text{Matr}(n \times m, \mathbb{Z}_q)$ auffassen. Dann setzen wir

$$\begin{aligned}\Lambda_q(A) &= \{\vec{v} \in \mathbb{Z}^n \mid \vec{v} = A^\top \cdot \vec{s} \text{ mod } q \text{ für ein } \vec{s} \in \mathbb{Z}^n\} \\ \Lambda_q^\perp(A) &= \{\vec{v} \in \mathbb{Z}^n \mid A \cdot \vec{v} = \vec{0} \text{ mod } q\}\end{aligned}$$

Hierfür erhält man (mit etwas mehr linearer Algebra als uns aktuell zur Verfügung steht).

Bemerkung 15.4.

a) Die Mengen $\Lambda_q(A)$ und $\Lambda_q^\perp(A)$ sind n -dimensionale q -Gitter.

b) Es gilt

$$\begin{aligned}\Lambda_q(A) &= q \cdot \Lambda_q^\perp(A)^\vee \\ \Lambda_q^\perp(A) &= q \cdot \Lambda_q(A)^\vee\end{aligned}$$

15.2. Gitterprobleme

Mit Gittern verbunden sind viele im allgemeinen numerisch sehr harte Fragestellungen und Probleme, die als Basis für public-key-Kryptosysteme in Frage kommen:

Shortest Vector Problem SVP

Gegeben ist ein n -dimensionales Gitter \mathcal{L} und gesucht ist ein kürzestes nicht-verschwindender Vektor in \mathcal{L} , also ein Vektor $\vec{v} \in \mathcal{L} \setminus \{\vec{0}\}$ mit

$$|\vec{v}| \leq |\vec{w}| \quad \text{für alle } \vec{w} \in \mathcal{L} \setminus \{\vec{0}\}$$

Dabei ist

$$|\vec{v}| = \sqrt{v_1^2 + \dots + v_n^2}$$

die euklidische Norm des Vektors \vec{v} .

Closest Vector Problem CVP

Gegeben ist ein n -dimensionales Gitter \mathcal{L} und ein Vektor $\vec{x} \in \mathbb{R}^n$, und gesucht ist ein dazu nächstliegender Vektor in \mathcal{L} , also ein $\vec{v} \in \mathcal{L}$ mit

$$|\vec{x} - \vec{v}| \leq |\vec{x} - \vec{w}| \quad \text{für alle } \vec{w} \in \mathcal{L}$$

Shortest Independent Vector Problem SIVP

Gegeben ist ein n -dimensionales Gitter \mathcal{L} . Für einen Satz

$$S = \{\vec{s}_1, \dots, \vec{s}_n\} \subseteq \mathcal{L}$$

von linear unabhängigen Vektoren in \mathcal{L} setze

$$\|S\| = \max\{|\vec{s}_i| \mid i = 1, \dots, n\}$$

Gesucht ist ein solcher Satz S von n linear unabhängigen Vektoren in \mathcal{L} , sodass

$$\|S\| \leq \|S'\|$$

für jeden weiteren Satz S' von n linear unabhängigen Vektoren in \mathcal{L} .

Bemerkung 15.5. Die Lösung dieser Probleme muss nicht notwendig eindeutig sein.

Beispiel 15.5. Das Gitter $\mathcal{L}(B)$ mit der Basis $B = \{\vec{b}_1, \vec{b}_2\}$ mit

$$\vec{b}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \vec{b}_2 = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

hat kürzesten Vektor

$$\vec{v} = \vec{b}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Das ist leicht zu sehen, denn weder \vec{e}_1 noch \vec{e}_2 sind in $\mathcal{L}(B)$.

Dagegen ist B keine kürzeste Basis, denn $\|B\| = |\vec{b}_2| = 3$. Kürzer ist $S = \{\vec{s}_1, \vec{s}_2\}$ mit

$$\vec{s}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \vec{s}_2 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

denn $\|S\| = |\vec{s}_2| = \sqrt{5} < 3$. Dieses S ist dann in der Tat auch ein kürzestes System linear unabhängiger Vektoren.

Die Probleme SVP, CVP und SIVP sind im allgemeinen hart und auch Quanten-hart. Es gibt etwa aktuell keine Indikatoren, die gegen folgende Vermutung sprechen

Vermutung:

Es gibt keinen Algorithmus und keinen Quantenalgorithmen mit polynomialer Laufzeit in n , der das kürzeste Vektorproblem für ein n -dimensionales Gitter bis auf einen polynomialem Faktor löst.

Notwendig für ein kryptographisches Problem ist aber auch hier, dass die Probleme eine „Falltür“ haben.

Beispiel 15.6. Wir betrachten ein n -dimensionales Gitter $\mathcal{L}(B)$ mit einer Basis

$$B = \{\overrightarrow{b_1}, \overrightarrow{b_2}, \dots, \overrightarrow{b_n}\}$$

wobei wir annehmen, dass die Vektoren $\overrightarrow{b_i}$ paarweise orthogonal sind, dass also gilt

$$\langle \overrightarrow{b_i}, \overrightarrow{b_j} \rangle = 0 \quad \text{für } i \neq j$$

und dass

$$|\overrightarrow{b_1}| \leq |\overrightarrow{b_2}| \leq \dots \leq |\overrightarrow{b_n}|$$

In diesem Fall gilt für ein beliebiges $\overrightarrow{x} \in R^n$ mit einer Darstellung

$$\overrightarrow{x} = r_1 \cdot \overrightarrow{b_1} + r_2 \cdot \overrightarrow{b_2} + \dots + r_n \cdot \overrightarrow{b_n}$$

dass

$$|\overrightarrow{x}|^2 = |r_1|^2 \cdot |\overrightarrow{b_1}|^2 + |r_2|^2 \cdot |\overrightarrow{b_2}|^2 + \dots + |r_n|^2 \cdot |\overrightarrow{b_n}|^2$$

Hieraus folgt sofort, dass $\overrightarrow{b_1}$ ein kürzester Vektor in $\mathcal{L}(B)$ ist. Ist nämlich $\overrightarrow{v} \in \mathcal{L} \setminus \{\overrightarrow{0}\}$ beliebig, so können wir schreiben

$$\overrightarrow{v} = z_1 \cdot \overrightarrow{b_1} + z_2 \cdot \overrightarrow{b_2} + \dots + z_n \cdot \overrightarrow{b_n}$$

mit $z_i \in \mathbb{Z}$, wobei mindestens ein $z_{i_0} \neq 0$. Dann gilt aber

$$|\overrightarrow{v}|^2 \geq |z_{i_0}|^2 \cdot |\overrightarrow{b_{i_0}}|^2 \geq |\overrightarrow{b_1}|^2$$

und daraus ergibt sich diese Aussage.

Die Basis B ist auch die kürzeste linear unabhängige Teilmenge in \mathcal{L} bestehend aus n Vektoren. Ist nämlich $S = \{\overrightarrow{s_1}, \dots, \overrightarrow{s_n}\}$ eine beliebige Menge von n linear unabhängigen Vektoren von \mathcal{L} , so gibt es mindestens einen Vektor $\overrightarrow{s_i}$, sodass in der Darstellung

$$\overrightarrow{s_i} = z_{i,1} \cdot \overrightarrow{b_1} + z_{i,2} \cdot \overrightarrow{b_2} + \dots + z_{i,n} \cdot \overrightarrow{b_n} \quad (z_{i,j} \in \mathbb{Z})$$

der Koeffizient $z_{i,n}$ nicht verschwindet (denn andernfalls wären alle $\overrightarrow{s_j}$ schon in dem von den Vektoren $\overrightarrow{b_1}, \dots, \overrightarrow{b_{n-1}}$ erzeugten $n-1$ -dimensionalen Untervektorraum von \mathbb{R}^n , könnten also nicht linear unabhängig sein). Dann gilt aber

$$|\overrightarrow{s_i}|^2 \geq |z_{i,n}|^2 \cdot |\overrightarrow{b_n}|^2 \geq |\overrightarrow{b_n}|^2 = \|B\|^2$$

und damit auch $\|S\| \geq \|B\|$.

Ist nun schließlich $\overrightarrow{x} \in \mathbb{R}^n$ beliebig, schreiben wir

$$\overrightarrow{x} = r_1 \cdot \overrightarrow{b_1} + r_2 \cdot \overrightarrow{b_2} + \dots + r_n \cdot \overrightarrow{b_n}$$

mit $r_i \in \mathbb{R}$ und setzen wir

$$z_i = [r_i] \quad (\text{die Rundung von } r_i)$$

und

$$\overrightarrow{v} = z_1 \cdot \overrightarrow{b_1} + z_2 \cdot \overrightarrow{b_2} + \cdots + z_n \cdot \overrightarrow{b_n}$$

so ist \overrightarrow{v} ein dem Vektor \overrightarrow{x} nächstliegender Vektor in \mathcal{L} . Ist nämlich $\overrightarrow{w} \in \mathcal{L}$ beliebig,

$$\overrightarrow{w} = a_1 \cdot \overrightarrow{b_1} + a_2 \cdot \overrightarrow{b_2} + \cdots + a_n \cdot \overrightarrow{b_n}$$

mit $a_i \in \mathbb{Z}$, so gilt aufgrund der definierenden Eigenschaft der Rundung, dass

$$|r_i - a_i| \geq |r_i - z_i| \quad \text{für alle } i \in \{1, \dots, n\}$$

und daraus folgt, dass

$$|\overrightarrow{x} - \overrightarrow{w}| \geq |\overrightarrow{x} - \overrightarrow{v}|$$

Im Gegensatz zum \mathbb{R}^n selbst oder zu seinen Untervektorräumen hat nicht jedes n -dimensionale Gitter eine orthogonale Basis wie im Beispiel 15.6. Aus angriffstechnischer Sicht sind solche Gitter, die eine orthogonale Basis besitzen vielleicht nicht einmal besonders günstig. Für viele Anwendungen reicht es aber, eine Basis B zu kennen, bei der die Abweichung von der Orthogonalität „nicht zu groß“ ist, in dem Sinn, dass $\langle \overrightarrow{b}_i, \overrightarrow{b}_j \rangle$ für $i \neq j$ „klein“ im Vergleich zu den Größen $|\overrightarrow{b}_i|^2$ ist.

Definition 15.6. Für eine Basis $B = \{\overrightarrow{b_1}, \dots, \overrightarrow{b_n}\}$ von \mathbb{R}^n heißt

$$\text{od}(B) = \frac{\prod_{i=1}^n |\overrightarrow{b}_i|}{|\det(B)|}$$

der **Orthogonalitätsdefekt** von B .

Schreiben wir

$$(B^{-1})^\top = \left(\widehat{\overrightarrow{b}_1} \dots \widehat{\overrightarrow{b}_n} \right)$$

so heißt

$$\text{od}^\vee(B) = \frac{\prod_{i=1}^n |\widehat{\overrightarrow{b}_i}|}{|\det(B^{-1})|} = |\det(B)| \cdot \prod_{i=1}^n |\widehat{\overrightarrow{b}_i}|$$

der **duale Orthogonalitätsdefekt** von B .

Bemerkung 15.6. Der duale Orthogonalitätsdefekt $\text{od}^\vee(B)$ ist der Orthogonalitätsdefekt der zu B dualen Basis $(B^{-1})^\top$ des zu $\mathcal{L}(B)$ dualen Gitters.

Bemerkung 15.7. Es gilt

$$B \cdot B^\top = \left(\langle \vec{b}_i, \vec{b}_j \rangle \right)_{1 \leq i, j \leq n}$$

Ist also B eine orthogonale Basis, so ist $B \cdot B^\top$ eine Diagonalmatrix mit Diagonaleinträgen $\langle \vec{b}_i, \vec{b}_i \rangle = |\vec{b}_i|^2$, und daher gilt in diesem Fall

$$\det(B)^2 = \det(B \cdot B^\top) = \prod_{i=1}^n |\vec{b}_i|^2$$

also

$$\text{od}(B) = 1$$

Man kann ferner zeigen, dass $\text{od}(B) > 1$, falls B nicht orthogonal ist, sodass also B genau dann orthogonal ist wenn $\text{od}(B) = 1$. Je größer $\text{od}(B)$ ist, umso mehr weicht B von einer orthogonalen Basis ab.

Für kryptographische Zwecke werden daher häufig zwei Basen P und O betrachtet, wobei P eine Basis ist, für die $\text{od}(P)$ (oder $\text{od}^\vee(P)$) nahe bei 1 liegt, während $\text{od}(O) \gg 1$. Die Basis P spielt dann eine wichtige Rolle im privaten Schlüssel, die Basis O wird im öffentlichen Schlüssel verwendet.

15.3. Gitterbasierte Verschlüsselung - Erster Versuch

Ein erster Ansatz zu einem gitterbasierten public-key-Verfahren ist der folgende, stark vereinfachte Ansatz einer Technik von Goldreich, Goldwasser und Halevi und beruht auf dem CLP–Problem:

Wir betrachten ein n –dimensionales Gitter $\mathcal{L} \subseteq \mathbb{Z}^n$ das eine orthogonale Basis $P = \{\vec{p}_1, \dots, \vec{p}_n\}$ besitzt und wir betrachten eine weitere Basis $O = P \cdot U$ (mit einer Matrix $U \in \text{Matr}(n \times n, \mathbb{Z})$ mit $\det(U) = \pm 1$) von \mathcal{L} mit sehr hohem Orthogonalitätsdefekt.

Der private Schlüssel des Verfahrens ist das Gitter \mathcal{L} und die Basis P , der öffentliche Schlüssel ist das Gitter \mathcal{L} und die Basis O . Die Sicherheit des Verfahrens beruht darauf, dass das CLP–Problem mit der Basis O numerisch nicht in akzeptabler Zeit lösbar ist.

Verschlüsselung durch Bob:

Bob leitet mithilfe des öffentlichen Schlüssel O aus seiner Nachricht m einen Gitterpunkt ab und stört diesen dann hinreichend stark, sodass der Gitterpunkt nicht mehr erkennbar ist. Dazu geht er vor wie folgt:

1. Bob hat seine Nachricht im Format $\vec{m} = (m_1, \dots, m_n)^\top \in \mathbb{Z}^n$ vorliegen.
2. Bob bestimmt einen Fehler- oder Störvektor $\vec{e} = (e_1, \dots, e_n)^\top \in \mathbb{R}^n$ mit „kleinen“ Einträgen (z.B. Bob wählt zufällig die e_i aus einer Menge $\{-a, -b, a, b\}$ aus, wobei jedes Element mit einer Wahrscheinlichkeit von $p = \frac{1}{4}$ gezogen wird und a unb b in einem von Alice vorgegebenem Bereich liegen).
3. Bob berechnet $c = O \cdot \vec{m} + \vec{e}$
4. Bob schickt c an Alice.

Bemerkung 15.8. Die Forderung, dass die Einträge des Fehlervektors \vec{e} „klein“ sind, ist so zu verstehen:

Schreiben wir

$$P^{-1} \cdot \vec{e} = \sum_{i=1}^n r_i \cdot \vec{p}_i$$

so ist $[r_i] = 0$ für alle $i \in \{1, \dots, n\}$. Das kann durch geeignete Wahl des Gitters und seiner orthogonalen Basis, durch angemessene Einschränkungen an die Größe der Komponenten des Fehlervektors und durch die Tatsache, dass die Komponenten des Fehlervektors im Mittel 0 ergeben sollen, sichergestellt werden. Die Kenntnis von P ist dafür nicht explizit erforderlich.

Entschlüsselung durch Alice:

Alice entschlüsselt die Nachricht von Bob indem sie mit ihrem privaten Schlüssel P den nächstgelegenen Gitterpunkt bestimmt und daraus die Nachricht m zurückberechnet. Sie nutzt dabei aus, das $O = P \cdot U$ und geht vor wie folgt:

1. Alice berechnet

$$\vec{x} = P^{-1} \cdot \vec{c} = P^{-1} \cdot (P \cdot U \cdot \vec{m} + \vec{e}) = U \cdot \vec{m} + P^{-1} \cdot \vec{e}$$

2. Alice schreibt

$$\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

(stellt also \vec{x} mit der Standardbasis dar) und setzt

$$\vec{y} = [\vec{x}] = \begin{pmatrix} [x_1] \\ \vdots \\ [x_n] \end{pmatrix}$$

(sie runden also alle Einträge von \vec{x}).

3. Alice berechnet

$$\vec{m}^* = U^{-1} \cdot [\vec{x}]$$

und arbeitet mit \vec{m}^* als von Bob übermittelte Nachricht.

Bemerkung 15.9. Bei diesem Vorgehen handelt es sich tatsächlich um ein Verschlüsselungsverfahren, dh. Alice erhält in der Tat die von Bob gesendete Nachricht \vec{m} zurück,

$$\vec{m}^* = \vec{m}$$

Dazu ist zu beachten, dass $U \cdot \vec{m}$ ein Vektor mit ganzzahligen Komponenten ist (da $\vec{m} \in \mathbb{Z}^n$ und U eine Matrix mit ganzzahligen Koeffizienten). Nach Voraussetzung sind die Komponenten von \vec{e} klein in dem Sinn, dass

$$[P^{-1} \cdot \vec{e}] = \vec{0}$$

und daher gilt

$$[\vec{x}] = [U \cdot \vec{m} + P^{-1} \cdot \vec{e}] = U \cdot \vec{m}$$

also ist

$$\vec{m}^* = U^{-1} \cdot [\vec{x}] = U^{-1} \cdot U \cdot \vec{m} = \vec{m}$$

Bemerkung 15.10. Durch die Operation $\vec{x} = P^{-1} \cdot \vec{c}$ wird die Darstellung des Vektors \vec{c} mithilfe der Basis P , bestimmt, dh. es gilt

$$\vec{c} = x_1 \cdot \vec{p}_1 + x_2 \cdot \vec{p}_2 + \cdots + x_n \cdot \vec{p}_n$$

Durch die Opteration $\vec{y} = [\vec{x}]$ wird der zu \vec{c} nächstgelegene Gitterpunkt bestimmt, dh.

$$\vec{d} = [x_1] \cdot \vec{p}_1 + [x_2] \cdot \vec{p}_2 + \cdots + [x_n] \cdot \vec{p}_n$$

ist der Gitterpunkt, der \vec{c} am nächsten liegt. Die Operation $\vec{m}^* = U^{-1} \cdot \vec{y}$ rechnet diese Darstellung in die Basis $O = \{\vec{o}_1, \vec{o}_2, \dots, \vec{o}_n\}$, dh.

$$\vec{d} = m_1^* \cdot \vec{o}_1 + m_2^* \cdot \vec{o}_2 + \cdots + m_n^* \cdot \vec{o}_n$$

Da \vec{e} als Störvektor „klein“ angenommen war, ist der nächstgelegene Gitterpunkt zum Punkt $c = O \cdot \vec{m} + \vec{e}$ der Punkt $O \cdot \vec{m}$, und deshalb muss

$$\vec{d} = O \cdot \vec{m}$$

gelten, und damit ist nochmal begründet, dass $m_i^* = m_i$ für alle $i \in \{1, \dots, n\}$.

Bemerkung 15.11. Bei der Wahl des Gitters \mathcal{L} ist es wichtig, darauf zu achten, dass die Determinante $\det(\mathcal{L})$ des Gitters hinreichend groß ist. Es gilt nämlich die **Minkowski–Schranke**:

In $\mathcal{L} \setminus \{\vec{0}\}$ gibt es einen Gitterpunkt \vec{g} mit

$$|\vec{g}| \leq \sqrt{n} \cdot \sqrt{\det(\mathcal{L})}$$

Ist also $\mathcal{L} \subseteq \mathbb{Z}^n$ ein Gitter mit kleiner Determinante, so kann ein kürzester Vektor leicht durch ausprobieren gefunden werden.

Beispiel 15.7. Alice betrachtet das 4–dimensionale Gitter \mathcal{L} mit der orthogonalen Basis $P = \{\vec{p}_1, \vec{p}_2, \vec{p}_3, \vec{p}_4\}$ mit

$$\vec{p}_1 = \begin{pmatrix} 89 \\ 89 \\ 89 \\ 89 \end{pmatrix}, \quad \vec{p}_2 = \begin{pmatrix} 73 \\ -73 \\ 73 \\ -73 \end{pmatrix}, \quad \vec{p}_3 = \begin{pmatrix} 79 \\ 79 \\ -79 \\ -79 \end{pmatrix}, \quad \vec{p}_4 = \begin{pmatrix} 97 \\ -97 \\ -97 \\ 97 \end{pmatrix}$$

also mit dem privaten Schlüssel

$$P = \begin{pmatrix} 89 & 73 & 79 & 97 \\ 89 & -73 & 79 & -97 \\ 89 & 73 & -79 & -97 \\ 89 & -73 & -79 & 97 \end{pmatrix}$$

Für dieses Gitter gilt

$$\det(\mathcal{L}) = 796\,584\,176$$

kürzeste Vektoren in diesem Gitter können also nur mit erheblichem numerischen Aufwand durch Ausprobieren gefunden werden.

Ferner wählt Alice die Matrix U mit

$$U = \begin{pmatrix} 9 & 28 & 9 & 12 \\ 8 & 17 & -6 & 23 \\ -9 & -13 & 20 & -30 \\ 7 & 12 & -8 & 30 \end{pmatrix}$$

Hierfür gilt in der Tat $\det(U) = 1$ und

$$U^{-1} = \begin{pmatrix} 2812 & -8718 & -2762 & 2797 \\ -956 & 2964 & 939 & -951 \\ 389 & -1206 & -382 & 387 \\ -170 & 527 & 167 & -169 \end{pmatrix}$$

Der öffentliche Schlüssel ist dann

$$O = P \cdot U = \begin{pmatrix} 1353 & 3879 & 1167 & 3287 \\ -1173 & -940 & 3595 & -5891 \\ 1417 & 3596 & -441 & 2207 \\ 1607 & 3442 & -1117 & 4669 \end{pmatrix}$$

Hierfür gilt dann

$$\text{od}(O) \approx 1063.46$$

diese Basis hat also einen sehr hohen Orthogonalitätsdefekt.

Bob will nun die Nachricht $\vec{m} = (13, 7, -17, 11)^\top$ verschlüsseln und an Alice schicken. Als Störvektor wählt er $\vec{e} = (11, -19, 23, -13)^\top$. Dieser Vektor ist „klein“ in dem geforderten Sinn, da

$$P^{-1} \cdot \vec{e} = \begin{pmatrix} 0.006 \\ 0.226 \\ -0.057 \\ -0.015 \end{pmatrix}$$

sodass also in der Tat

$$[P^{-1} \cdot \vec{e}] = \vec{0}$$

Damit berechnet Bob

$$\vec{c} = O \cdot \vec{m} + \vec{e} = \begin{pmatrix} 61\,008 \\ -147\,764 \\ 75\,390 \\ 115\,320 \end{pmatrix}$$

Diesen Vektor \vec{c} schickt er an Alice.

Alice empfängt \vec{c} und berechnet zunächst (gerundet auf drei Nachkommastellen)

$$\vec{x} = P^{-1} \cdot \vec{c} = \begin{pmatrix} 292.006 \\ 578.226 \\ -878.057 \\ 640.985 \end{pmatrix}$$

Sie rundet diesen Vektor und erhält

$$\vec{y} = [\vec{x}] = \begin{pmatrix} 292 \\ 578 \\ -878 \\ 641 \end{pmatrix}$$

Nun ermittelt sie

$$\overrightarrow{m^*} = U^{-1} \cdot \overrightarrow{y} = \begin{pmatrix} 13 \\ 7 \\ -17 \\ 11 \end{pmatrix}$$

Alice hat damit also tatsächlich \overrightarrow{m} , also Bobs Nachricht zurückgewonnen.

An diesem Beispiel kann auch verdeutlicht werden, dass das Closest Vector Problem bei beliebiger Basis des Gitters kompliziert ist. Man kann sich nämlich natürlich fragen, warum Alice bei der Darstellung des Vektors \overrightarrow{c} erst den Umweg über die Basis P macht und dann in die Basis O umrechnet, und warum Sie nicht gleich den Vektor \overrightarrow{c} mit der Basis O ausdrückt und in dieser Darstellung die Koeffizienten runden.

Um \overrightarrow{c} mit der Basis O zu beschreiben, berechnet Alice (gerundet auf drei Nachkommastellen)

$$\overrightarrow{a} = O^{-1} \cdot \overrightarrow{c} = \begin{pmatrix} -1827.633 \\ 632.793 \\ -271.629 \\ 122.262 \end{pmatrix}$$

Die Rundung dieses Vektors ergibt

$$\overrightarrow{b} = [\overrightarrow{a}] = \begin{pmatrix} -1828 \\ 633 \\ -272 \\ 122 \end{pmatrix}$$

und dieser Vektor unterscheidet sich sehr deutlich von dem Vektor \overrightarrow{m} .

Durch diesen Vektor \overrightarrow{b} wird zwar tatsächlich ein Gitterpunkt definiert,

$$\overrightarrow{g} = b_1 \cdot \overrightarrow{o_1} + \cdots + b_4 \cdot \overrightarrow{o_4} = \begin{pmatrix} 60\,016 \\ -147\,318 \\ 75\,198 \\ 114\,632 \end{pmatrix}$$

und die Komponenten dieses Gitterpunktes unterscheiden sich (prozentual betrachtet) auch nur relativ wenig von den Komponenten von \overrightarrow{c} , aber trotzdem ist

$$|\overrightarrow{c} - \overrightarrow{g}| = \sqrt{1\,693\,188}$$

wohingegen

$$|\overrightarrow{c} - O \cdot \overrightarrow{m}| = |\overrightarrow{e}| = \sqrt{1180}$$

dh. der Abstand von \vec{g} zu \vec{c} ist signifikant größer als der von $O \cdot \vec{m}$ zu \vec{c} , und aus der Beschreibung von \vec{c} mit O lassen sich keine Rückschlüsse auf den nächstgelegenen Gitterpunkt ziehen.

Nicht jedes Gitter ist geeignet für gitterbasiertes public-key-Verfahren. Es wäre z.B. naheliegend, ein Gitter mit einer orthogonalen Basis P der Form $\vec{p}_i = z_i \cdot \vec{e}_i$ für $i = 1, \dots, n$ (mit den Standardbasisvektoren \vec{e}_i und z_i hinreichend groß) zu betrachten. Ein solches Gitter ist jedoch ungeeignet aufgrund der folgenden Aussage

Satz 15.3 (Hermite). *Jedes Gitter \mathcal{L} hat eine eindeutig bestimmte und effektiv berechenbare Basis $H = \{\vec{h}_1, \dots, \vec{h}_n\}$, sodass die zugehörige Matrix $H = (h_{i,j})$ die folgenden Eigenschaften hat:*

1. H ist eine untere Dreiecksmatrix, dh. $h_{i,j} = 0$ für $j > i$.
2. $h_{i,i} > 0$ für alle $i \in \{1, \dots, n\}$.
3. $0 \leq h_{i,j} < h_{i,i}$ für $j < i$.

Definition 15.7. Die Basis H aus Satz 15.3 heißt **Gitterbasis in Hermite—Normalform** des Gitters \mathcal{L} .

Der Nachweis der Aussage erfolgt durch Anwendung elementarer Spaltenumformungen, wobei allerdings auf Division und Multiplikation mit Spalten (außer mit der -1) verzichtet wird, dh. es werden Spalten vertauscht, es werden ganzzahlige Vielfache einer Spalte von einer anderen abgezogen oder es werden Spalten mit -1 multipliziert. Diese Spaltenumformungen sind Äquivalenzoperationen, sie ändern also das von den Spalten erzeugte Gitter nicht.

Statt eines formalen Beweises wollen wir hier ein Beispiel durchrechnen, das auch zeigt, dass eine Basis P der Form $\vec{p}_i = z_i \cdot \vec{e}_i$ ($i = 1, \dots, n$) ungeeignet ist.

Beispiel 15.8. Wir betrachten das Gitter \mathcal{L} mit Basis

$$B = \{178 \cdot \vec{e}_1, 146 \cdot \vec{e}_2, 158 \cdot \vec{e}_3, 194 \cdot \vec{e}_4\}$$

in Matrixschreibweise also

$$B = \begin{pmatrix} 178 & 0 & 0 & 0 \\ 0 & 146 & 0 & 0 \\ 0 & 0 & 158 & 0 \\ 0 & 0 & 0 & 194 \end{pmatrix}$$

Hierfür gilt $\det(\mathcal{L}) = 796\,584\,176$, die Grundmasche des Gitters hat also das gleiche Volumen wie die des Gitters aus Beispiel 15.7

Für U wählen wir die gleiche Matrix wie in Beispiel 15.7 und erhalten damit

$$Q = B \cdot U = \begin{pmatrix} 1602 & 4984 & 1602 & 2136 \\ 1168 & 2482 & -876 & 3358 \\ -1422 & -2054 & 3160 & -4740 \\ 1358 & 2328 & -1552 & 5820 \end{pmatrix}$$

Hierfür gilt

$$\text{od}(Q) = 752\,910$$

der Orthogonalitätsdefekt ist also beachtlich.

Aus dieser Matrix wollen wir durch elementare Spaltenoperationen eine Hermite–Normalform herleiten. Dazu gehen wir vor wie folgt:

Wir subtrahieren die erste Spalte dreimal von der zweiten und einmal von der dritten und der vierten Spalte und erhalten

$$Q_1 = \begin{pmatrix} 1602 & 178 & 0 & 534 \\ 1168 & -1022 & -2044 & 2190 \\ -1422 & 2212 & 4582 & -3318 \\ 1358 & -1746 & -2910 & 4492 \end{pmatrix}$$

Wir subtrahieren die zweite Spalte neun mal von der ersten und dreimal von der dritten:

$$Q_2 = \begin{pmatrix} 0 & 178 & 0 & 0 \\ 10366 & -1022 & -2044 & 5256 \\ -21330 & 2212 & 4582 & -9954 \\ 17072 & -1746 & -2910 & 9700 \end{pmatrix}$$

Vertauschung der ersten beiden Spalten führt zu

$$Q_3 = \begin{pmatrix} 178 & 0 & 0 & 0 \\ -1022 & 10\,366 & -2044 & 5256 \\ 2212 & -21\,330 & 4582 & -9954 \\ -1746 & 17\,072 & -2910 & 9700 \end{pmatrix}$$

Wir addieren die dritte Spalte fünfmal zur zweiten und zweimal zur dritten

$$Q_4 = \begin{pmatrix} 178 & 0 & 0 & 0 \\ -1022 & 146 & -2044 & 1168 \\ 2212 & 1580 & 4582 & -790 \\ -1746 & 2522 & -2910 & 3880 \end{pmatrix}$$

Wir addieren die zweite Spalte vierzehn mal zur dritten und subtrahieren sie achtmal von der vierten:

$$Q_5 = \begin{pmatrix} 178 & 0 & 0 & 0 \\ -1022 & 146 & 0 & 0 \\ 2212 & 1580 & 26702 & -13430 \\ -1746 & 2522 & 32398 & -16296 \end{pmatrix}$$

Wir subtrahieren die vierte Spalte zweimal von der dritten und multiplizieren sie dann mit -1 :

$$Q_6 = \begin{pmatrix} 178 & 0 & 0 & 0 \\ -1022 & 146 & 0 & 0 \\ 2212 & 1580 & 158 & -13430 \\ -1746 & 2522 & 194 & -16296 \end{pmatrix}$$

Addition des fünfundachtzigfachen der dritten Spalte zur vierten ergibt

$$Q_7 = \begin{pmatrix} 178 & 0 & 0 & 0 \\ -1022 & 146 & 0 & 0 \\ 2212 & 1580 & 158 & 0 \\ -1746 & 2522 & 194 & 194 \end{pmatrix}$$

Nun subtrahiere die vierte Spalte einmal von der dritten, dreizehnmal von der zweiten und addiere sie neunmal zur ersten:

$$Q_8 = \begin{pmatrix} 178 & 0 & 0 & 0 \\ -1022 & 146 & 0 & 0 \\ 2212 & 1580 & 158 & 0 \\ 0 & 0 & 0 & 194 \end{pmatrix}$$

Subtraktion des zehnfachen der dritten Spalte von der zweiten und des vierzehnfachen der dritten Spalte von der ersten liefert

$$Q_9 = \begin{pmatrix} 178 & 0 & 0 & 0 \\ -1022 & 146 & 0 & 0 \\ 0 & 0 & 158 & 0 \\ 0 & 0 & 0 & 194 \end{pmatrix}$$

und Addition des siebenfachen der zweiten Spalte zur ersten führt zur Hermite–Normalform

$$H = \begin{pmatrix} 178 & 0 & 0 & 0 \\ 0 & 146 & 0 & 0 \\ 0 & 0 & 158 & 0 \\ 0 & 0 & 0 & 194 \end{pmatrix}$$

Wir haben also die orthogonale Ausgangsbasis zurückgewonnen.

Damit kann auch Catherine diese Basis gewinnen und somit die Verschlüsselung brechen.

Beispiel 15.9. Die Basis in Hermite–Normalform für das Gitter aus Beispiel 15.7 ist

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 76\,078\,265 & 119\,726\,714 & 83\,640\,774 & 199\,146\,044 \end{pmatrix}$$

Hierfür gilt

$$\text{od}(H) \approx 1.9 \cdot 10^{23}$$

Die Gitterbasis in Hermite–Normalform hat also einen extrem hohen Orthogonalitätsdefekt und ist hier nicht geeignet, den nächstgelegenen Gitterpunkt zu einem allgemeinen Punkt der \mathbb{R}^4 zu finden.

Bemerkung 15.12. Da Catherine die Hermite–Normalform sowieso effektiv berechnen kann, wird als öffentlicher Schlüssel üblicherweise eine Gitterbasis in Hermite–Normalform genommen.

Voraussetzung ist dann natürlich, dass diese Gitterbasis einen hohen Orthogonalitätsdefekt besitzt.

Bemerkung 15.13. Die wenigsten n –dimensionalen Gitter haben eine orthogonale Basis. Gitter mit orthogonalen Basen sind möglicherweise auch anfälliger für numerische Angriffe als allgemeine Gitter.

Um das Verfahren durchzuführen, reicht es aber, eine Gitter mit einer Gitterbasis mit vergleichsweise niedrigem Orthogonalitätsdefekt zu finden, für das die Gitterbasis in Hermite–Form einen sehr hohen Orthogonalitätsdefekt besitzt.

15.4. Learning With Errors LWE

Wir betrachten eine Primzahl $p > 2$ und ein q –Gitter $\Lambda_q(A)$, das gegeben ist durch eine Matrix $A \in \text{Matr}(n \times m, \mathbb{F}_q)$, also

$$\Lambda_q(A) = \{\vec{v} \in \mathbb{Z}^n \mid \vec{v} = A^\top \cdot \vec{s} \bmod p \text{ für ein } \vec{s} \in \mathbb{Z}^n\}$$

Bei dem Learning–With–Errors–Problem geht es im wesentlichen darum, Methoden zu entwickeln, die entscheiden, ob ein gegebener Vektor $\vec{w} \in \mathbb{Z}^n$ zufällig aus \mathbb{Z}^n ausgewählt wurde oder ob er von der Form

$$\vec{v} = \vec{g} + \vec{e}$$

mit einem zufällig ausgewählten Gitterpunkt $\vec{g} \in \Lambda_q(A)$ und einer Störvektor \vec{e} ist, wobei die Komponenten von \vec{e} „klein“ sind und zufällig (nach einer Zufallsverteilung mit Mittelwert 0) ausgewählt wurden. Hierbei kann es naturgemäß kein Verfahren geben, dass von Fall zu Fall exakt entscheidet, es läuft wieder darauf hinaus, herauszufinden, ob ein gegebener Vektor hinreichend nahe an einem Gitterpunkt liegt oder nicht. Nach aktuellem Stand der Algorithmik handelt es sich dabei um ein sehr schweres Problem, zu dem kein effizientes Lösungsverfahren bekannt ist.

In einem LWE-basierten asymmetrischen Verschlüsselungsverfahren werden binäre Nachrichten verschlüsselt. Dazu identifizieren wir zunächst

$$\mathbb{F}_p = \left\{ -\frac{p-1}{2}, -\frac{p-1}{2} + 1, \dots, \frac{p-1}{2} - 1, \frac{p-1}{2} \right\}$$

und betrachten die Abbildung

$$f : \mathbb{F}_2 \longrightarrow \mathbb{F}_p$$

mit

$$f(0) = 0, \quad f(1) = \frac{p-1}{2}$$

sowie

$$f^{-1} : \mathbb{F}_p \longrightarrow \mathbb{F}_2$$

mit

$$f^{-1}(a) = \begin{cases} 0 & \text{falls } -\lfloor \frac{p-1}{4} \rfloor \leq a \leq \lfloor \frac{p-1}{4} \rfloor \\ 1 & \text{sonst} \end{cases}$$

(sodass also f^{-1} eine linksinverse Abbildung zu f ist, dh.

$$f^{-1}(f(a)) = a \quad \text{für alle } a \in \mathbb{F}_2$$

wobei aber nicht $f(f^{-1}(x)) = x$ für alle $x \in \mathbb{F}_p$ gilt).

Der Nachrichtenraum für die Verschlüsselung ist \mathbb{F}_2^l für ein $l \in \mathbb{N}$, die Nachrichtenlänge ist also l .

Schlüsselerzeugung:

Alice erzeugt ein Schlüsselpaar wie folgt:

1. Alice wählt Parameter $n, m, r \in \mathbb{N}$ und eine Primzahl p , wobei r klein im Vergleich zu p ist.
2. Alice wählt zufällig eine Matrix $S \in \text{Matr}(n \times l, \mathbb{F}_p)$. Dieses S ist ihr privater Schlüssel,

$$k_{\text{pr}} = S$$

3. Alice wählt zufällig eine Matrix $A \in \text{Matr}(n \times m, \mathbb{F}_p)$ und eine Fehlermatrix $E \in \text{Matr}(n \times l, \mathbb{F}_p)$, wobei die Einträge von E klein (und gemäß einer vorgegebenen Zufallsverteilung mit Erwartungswert 0) gewählt sind.

4. Der öffentliche Schlüssel von Alice ist

$$k_{\text{pub}} = (A, O = A \cdot S + E)$$

Verschlüsselung:

Bob verschlüsselt eine Nachricht $\vec{m} \in \mathbb{F}_2^l$ wie folgt:

1. Bob wählt zufällig einen Vektor

$$\vec{a} \in \{-1, -r+1, \dots, r-1, r\}^m$$

2. Bob setzt

$$\vec{u} = A^\top \cdot \vec{a}, \quad \vec{c} = O^\top \cdot \vec{a} + f(\vec{m})$$

3. Bob schickt das Paar (\vec{u}, \vec{c}) an Alice.

Entschlüsselung:

Alice entschlüsselt die Nachricht wie folgt:

1. Alice empfängt (\vec{u}, \vec{c}) .
2. Alice entschlüsselt diese Nachricht zu

$$\vec{m}^* = f^{-1} (\vec{c} - S^\top \cdot \vec{u})$$

Bemerkung 15.14. Es ist

$$\begin{aligned} f^{-1} (\vec{c} - S^\top \cdot \vec{u}) &= f^{-1} (O^\top \cdot \vec{a} + f(\vec{m}) - S^\top \cdot A^\top \cdot \vec{a}) \\ &= f^{-1} ((A \cdot S + E)^\top \cdot \vec{a} + f(\vec{m}) - S^\top \cdot A^\top \cdot \vec{a}) \\ &= f^{-1} (E^\top \cdot \vec{a} + f(\vec{m})) \end{aligned}$$

Wenn also die Komponenten von $E^\top \cdot \vec{a}$ nicht übermäßig groß ist (wobei man bei richtiger Wahl von r und der Wahrscheinlichkeitsverteilung für E ausgehen kann), so erhält Alice die Nachricht von Bob zurück.

Bemerkung 15.15. Die Sicherheit des Verfahrens beruht darauf, dass es nicht möglich ist, zwischen einem öffentlichen Schlüssel (A, O) und einem zufällig gewählten Paar (A, O) von Matrizen zu unterscheiden und damit auf dem LWE–Problem.

Kryptologie

Anhänge

A. Rechnen mit Restklassen ganzer Zahlen

In der Kryptologie sind ganze Zahlen und ihre Restklassenringe von sehr großer Bedeutung. Die grundlegenden Eigenschaften sollen hier noch einmal wiederholt werden.

Wir betrachten dazu den Ring $R = \mathbb{Z}$ der ganzen Zahlen und eine natürliche Zahl $n \geq 2$.

Die Relation \sim auf \mathbb{Z} , gegeben durch

$$a \sim b \iff n|(b-a) \quad (\text{dh. } n \text{ teilt } b-a)$$

ist eine **Äquivalenzrelation** auf \mathbb{Z} , dh. sie erfüllt

1. für $a \in \mathbb{Z}$ ist $a \sim a$ (sie ist *reflexiv*).
2. für $a, b \in \mathbb{Z}$ mit $a \sim b$ gilt auch $b \sim a$ (sie ist *symmetrisch*)
3. für $a, b, c \in \mathbb{Z}$ mit $a \sim b$ und $b \sim c$ gilt auch $a \sim c$ (sie ist *transitiv*)

Eine Teilmenge $A \subseteq \mathbb{Z}$ heißt **Äquivalenzklasse** von \sim , wenn gilt

- Sind $a, b \in A$, so ist $a \sim b$.
- Ist $a \in A$ und $b \in \mathbb{Z}$ mit $a \sim b$, so ist $b \in A$.

Ist $a \in \mathbb{Z}$, so heißt $[a] := \{b \in \mathbb{Z} : b \sim a\}$ die **Äquivalenzklasse von a** .

Bemerkung A.1. Sind $a, b \in \mathbb{Z}$, so gilt entweder $[a] = [b]$ oder $[a]$ und $[b]$ sind disjunkt. Das folgt sofort aus den Definitionen.

Die Äquivalenzrelation \sim zerlegt also \mathbb{Z} in disjunkte Teilmengen, die Äquivalenzklassen. Wir schreiben $\mathbb{Z}/n\mathbb{Z}$ oder \mathbb{Z}_n für die Menge der Äquivalenzklassen, also

$$\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z} = \{[a] \mid a \in \mathbb{Z}\}$$

Bezeichnung:

Sind $a, b \in \mathbb{Z}$ mit $[a] = [b]$, so schreiben wir auch

$$a = b \pmod{n}$$

(a ist kongruent zu b modulo n).

Ist $A \subseteq \mathbb{Z}$ ein Äquivalenzklasse (bezüglich \sim), so heißt ein beliebiges Element $a \in A$ ein **Repräsentant** der Äquivalenzklasse A .

Ein **Repräsentantsystem** der Äquivalenzrelation \sim ist eine Teilmenge $P \subseteq \mathbb{Z}$ die genau einen Repräsentanten jeder Äquivalenzklasse enthält.

Bemerkung A.2. Für die Äquivalenzrelation \sim gibt es nur endlich viele Äquivalenzklassen. Genauer gilt

$$\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z} = \{[0], [1], \dots, [n-1]\}$$

Die Menge $\{0, 1, \dots, n-1\}$ bildet also ein Repräsentantensystem der Relation \sim (das *Standardrepräsentantensystem*). Es gibt aber noch viele weitere Repräsentantensysteme, etwa $\{1, 2, \dots, n\}$ oder $\{n, n+1, \dots, 2n-1\}$ oder $\{0, n+1, 2n+2, 3n+3, \dots, n^2-1\}$. Zur Vereinfachung der Notation schreiben wir oft

$$\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z} = \{0, 1, \dots, n-1\}$$

(falls es dadurch nicht zu Verwechslungen kommen kann).

Die Äquivalenzrelation \sim ist eine **Kongruenzrelation**, dh. sie erfüllt

- Sind $a_1, a_2, b_1, b_2 \in Z$ mit $a_1 \sim b_1$ und $a_2 \sim b_2$, so gilt $a_1 + b_1 \sim a_2 + b_2$.
- Sind $a_1, a_2, b_1, b_2 \in Z$ mit $a_1 \sim b_1$ und $a_2 \sim b_2$, so gilt $a_1 \cdot b_1 \sim a_2 \cdot b_2$.

Die Relation \sim ist also verträglich mit der Addition und der Multiplikation auf \mathbb{Z} und definiert daher eine Addition $+$ und eine Multiplikation \cdot auf den Restklassen via

- $[a] + [b] = [a + b]$ für alle $[a], [b] \in \mathbb{Z}_n$.
- $[a] \cdot [b] = [a \cdot b]$ für alle $[a], [b] \in \mathbb{Z}_n$.

Bemerkung A.3. $(\mathbb{Z}_n, +, \cdot)$ ist ein kommutativer Ring mit Nullelement $[0]$ und Eins-element $[1]$. Die Ringeigenschaften ergeben sich dabei leicht aus den entsprechenden Eigenschaften von $(\mathbb{Z}, +, \cdot)$.

Definition A.1. Ein Element $[a] \in \mathbb{Z}_n$, $[a] \neq [0]$, heißt **Nullteiler** von \mathbb{Z}_n , wenn es ein Element $[b] \in \mathbb{Z}_n$, $[b] \neq [0]$, gibt mit

$$[a] \cdot [b] = [0]$$

Ein Element $[a] \in \mathbb{Z}_n$ heißt **Einheit** von \mathbb{Z}_n , wenn es ein Element $[b] \in \mathbb{Z}_n$ gibt mit

$$[a] \cdot [b] = [1]$$

Mit $E(\mathbb{Z}_n)$ oder \mathbb{Z}_n^* bezeichnen wir die Menge aller Einheiten von \mathbb{Z}_n ,

$$\mathbb{Z}_n^* = E(\mathbb{Z}_n) = \{[a] \in \mathbb{Z}_n \mid \exists [b] \in \mathbb{Z}_n \text{ mit } [a] \cdot [b] = [1]\}$$

und wir setzen

$$\varphi(n) = |E(\mathbb{Z}_n)|$$

dh. $\varphi(n)$ bezeichnet die Anzahl der Einheiten von \mathbb{Z}_n .

Bemerkung A.4. Die Funktion φ heißt **Eulersche- φ -Funktion**.

Bemerkung A.5. Ist $[a] \in \mathbb{Z}_n$ ein Nullteiler, so ist $[a]$ keine Einheit von \mathbb{Z}_n .

Ist nämlich $[a]$ ein Nullteiler, so gibt es ein $[b] \in \mathbb{Z}_n$, $[b] \neq [0]$, mit $[a] \cdot [b] = [0]$. Wäre $[a]$ auch eine Einheit, so gibt es außerdem ein $[c] \in \mathbb{Z}_n$ mit $[c] \cdot [a] = [1]$. Dann würde gelten

$$[b] = [1] \cdot [b] = ([c] \cdot [a]) \cdot [b] = [c] \cdot ([a] \cdot [b]) = [c] \cdot [0] = [0]$$

ein Widerspruch zu $[b] \neq [0]$. Also ist $[a]$ keine Einheit.

Entsprechend gilt: Ist $[a] \in \mathbb{Z}_n$ eine Einheit, so ist $[a]$ kein Nullteiler von \mathbb{Z}_n .

Beispiel A.1. Wir betrachten $\mathbb{Z}_6 = \{[0], [1], [2], [3], [4], [5]\}$. Dann gilt

[1] ist eine Einheit, denn $[1] \cdot [1] = [1]$.

[2] ist ein Nullteiler, denn $[2] \cdot [3] = [6] = [0]$. Speziell ist also [2] keine Einheit.

[3] ist ein Nullteiler, denn $[3] \cdot [2] = [6] = [0]$. Speziell ist also [3] keine Einheit.

[4] ist ein Nullteiler, denn $[4] \cdot [3] = [12] = [0]$. Speziell ist also [4] keine Einheit.

[5] ist eine Einheit, denn $[5] \cdot [5] = [25] = [1]$.

Das Element [0] ist weder Einheit (denn $[0] \cdot [a] = [0] \neq [1]$ für alle $[a]$) noch Nullteiler.

Damit ist

$$E(\mathbb{Z}_6) = \{[1], [5]\}$$

Bemerkung A.6. Die Menge $E(\mathbb{Z}_n)$, zusammen mit \cdot , ist eine endliche Gruppe mit $\varphi(n)$ Elementen. Insbesondere gilt für jedes $[a] \in E(\mathbb{Z}_n)$:

$$[a]^{\varphi(n)} = [1]$$

Ob ein Element $[a]$ eine Einheit von \mathbb{Z}_n ist, kann effizient nachgerechnet werden.

Bemerkung A.7. Ist $a \in \mathbb{Z}$ eine natürliche Zahl mit $\text{ggT}(a, n) > 1$, so ist $[a]$ keine Einheit in \mathbb{Z}_n .

Ist nämlich $g = \text{ggT}(a, n)$, so können wir

$$n = g \cdot n', \quad a = g \cdot a'$$

mit einem $n' < n$ schreiben. Damit ist sicherlich $[n'] \neq [0]$ und

$$[a] \cdot [n'] = [a'] \cdot [g] \cdot [n'] = [a'] \cdot [n] = [a'] \cdot [0] = [0]$$

Also ist $[a]$ Nullteiler von \mathbb{Z}_n und daher keine Einheit.

Es bleiben die Restklassen $[a]$ von Elementen $a \in \mathbb{Z}$ mit $\text{ggT}(a, n) = 1$ zu untersuchen.
Hier hilft der

Der euklidische Algorithmus:

- **Vorbereitungsschritt:** Wir ordnen a und n so an, dass $n \geq a$. Gegebenenfalls vertauschen wir dazu die Rollen von a und n , denn $\text{ggT}(a, n) = \text{ggT}(n, a)$. Wir setzen $i = 0$ und $r_0 = n$, $r_1 = a$.
- **Verarbeitungsschritt:** Wir dividieren r_i durch r_{i+1} mit Rest:

$$r_i = q \cdot r_{i+1} + b$$

mit einer natürlichen Zahl q und einem Rest $b \in \{0, 1, \dots, r_{i+1} - 1\}$.

- Falls $b = 0$ (dh. die Division geht ohne Rest auf) \rightarrow **STOPP**.
- Falls $b \neq 0$ setze $r_{i+2} = b$ und $i = i + 1$. Wiederhole den Verarbeitungsschritt.
- **Ergebnisschritt:** Nach endlich vielen Verarbeitungsschritten (höchstens a vielen) geht die Division erstmals ohne Rest auf, d.h.

$$r_i = q \cdot r_{i+1} + 0$$

mit $r_{i+1} \neq 0$. Das STOPP-Kriterium wird also immer erreicht. Dann ist r_{i+1} der größte gemeinsame Teiler von m und n , $r_{i+1} = \text{ggT}(m, n)$.

Beispiel A.2. Wir betrachten die Zahlen $m = 234$ und $n = 138$. Hier gilt bereits $m \geq n$, und wir setzen $r_0 = 234$ und $r_1 = 138$.

$$i = 0: \quad 234 = 1 \cdot 138 + 96. \text{ Wir setzen } r_2 = 96.$$

$$i = 1: \quad 138 = 1 \cdot 96 + 42. \text{ Wir setzen } r_3 = 42.$$

$$i = 2: \quad 96 = 2 \cdot 42 + 12. \text{ Wir setzen } r_4 = 12.$$

$$i = 3: \quad 42 = 3 \cdot 12 + 6. \text{ Wir setzen } r_5 = 6.$$

$$i = 4: \quad 12 = 2 \cdot 6 + 0. \rightarrow \text{STOPP.}$$

Ergebnis: $\text{ggT}(234, 138) = 6$.

Beispiel A.3. Wir betrachten die Zahlen $m = 19$ und $n = 234$. Hier müssen wir zuerst die Rollen von m und n vertauschen, setzen also $r_0 = 234$ und $r_1 = 19$.

$i = 0: 234 = 12 \cdot 19 + 6$. Wir setzen $r_2 = 6$.

$i = 1: 19 = 3 \cdot 6 + 1$. Wir setzen $r_3 = 1$.

$i = 2: 6 = 6 \cdot 1 + 0$. \longrightarrow **STOPP**.

Ergebnis: $\text{ggT}(234, 19) = 1$. Die Zahlen 234 und 19 sind also teilerfremd.

Bemerkung A.8. Der euklidische Algorithmus liefert ein effizientes Verfahren zur Berechnung des größten gemeinsamen Teilers, das für große Zahlen sehr viel schneller ist als der Weg über die Primfaktorzerlegung und die gemeinsamen Primfaktoren

Die für uns wichtigste Folgerung aus dem euklidischen Algorithmus ist

Satz A.1 (Lemma von Bezout). *Sind $m, n \in \mathbb{N} \setminus \{0\}$ mit $\text{ggT}(m, n) = g$, so gibt es ganze Zahlen r, s mit*

$$r \cdot m + s \cdot n = g$$

Der Beweis des Satzes besteht darin, dass wir aus dem vorletzten Schritt des euklidischen Algorithmus rückwärtsrechnen, also die Beziehung $r_{i-1} = q \cdot r_i + r_{i+1}$, in der r_{i+1} der größte gemeinsame Teiler von m und n ist, nach r_{i+1} auflösen, $r_{i+1} = r_{i-1} - q \cdot r_i$ und dann sukzessive die verschiedenen Schritte des Algorithmus zurückgehen. Das sieht man am besten direkt an Beispielen

Beispiel A.4. In Beispiel A.2 haben wir gesehen, dass $\text{ggT}(234, 138) = 6$. Um 6 mit 234 und 138 zu beschreiben, gehen wir vor wie folgt:

Aus dem Schritt $i = 3$ erhalten wir

$$6 = 42 - 3 \cdot 12$$

Aus $i = 2$ folgt $12 = 96 - 2 \cdot 42$. Setzen wir das ein, so ergibt sich

$$6 = 42 - 3 \cdot 12 = 42 - 3 \cdot (96 - 2 \cdot 42) = 7 \cdot 42 - 3 \cdot 96$$

Aus $i = 1$ erhalten wir $42 = 138 - 1 \cdot 96$. Setzen wir das ein, so ergibt sich

$$6 = 7 \cdot 42 - 3 \cdot 96 = 7 \cdot (138 - 1 \cdot 96) - 3 \cdot 96 = 7 \cdot 138 - 10 \cdot 96$$

Aus $i = 0$ folgt schließlich $96 = 234 - 1 \cdot 138$. Setzen wir das ein, so ergibt sich

$$6 = 7 \cdot 138 - 10 \cdot 96 = 7 \cdot 138 - 10 \cdot (234 - 138) = 17 \cdot 138 - 10 \cdot 234$$

Damit haben wir eine Darstellung

$$6 = 17 \cdot 138 + (-10) \cdot 234$$

wie gewünscht gefunden.

Beispiel A.5. In Beispiel A.3 haben wir gesehen, dass $\text{ggT}(234, 19) = 1$. Wir wollen nun 1 mit 19 und 234 darstellen.

1. Aus Schritt $i = 1$ erhalte: $1 = 19 - 3 \cdot 6$.

2. Aus Schritt $i = 0$ erhalte:

$$1 = 19 - 3 \cdot 6 = 19 - 3 \cdot (234 - 12 \cdot 19) = 37 \cdot 19 - 3 \cdot 234$$

Damit haben wir die gewünschte Darstellung

$$1 = 37 \cdot 19 + (-3) \cdot 234$$

gefunden.

Folgerung A.2. Ist $a \in \mathbb{Z}$ eine natürliche Zahl, für die $\text{ggT}(a, n) = 1$ ist, so ist $[a]$ eine Einheit in \mathbb{Z}_n .

Es gibt dann nämlich ganze Zahlen r, s mit

$$1 = r \cdot a + s \cdot n$$

Rechnen modulo n liefert dann

$$1 = r \cdot a + s \cdot n \bmod n = r \cdot a \bmod n$$

(da $s \cdot n = 0 \bmod n$), also

$$[1] = [r] \cdot [a]$$

und damit ist $[a]$ eine Einheit in \mathbb{Z}_n .

Beispiel A.6. In Beispiel A.5 haben wir gesehen, dass $\text{ggT}(234, 19) = 1$ und dass

$$1 = 37 \cdot 19 + (-3) \cdot 234$$

Daher ist $[a] = [19]$ eine Einheit in $\mathbb{Z}/234\mathbb{Z}$ und

$$1 = 37 \cdot 19 + (-3) \cdot 234 \bmod 234 = 37 \cdot 19 \bmod 234$$

Also ist $[b] = [37]$ das multiplikative Inverse zu $[a] = [19]$,

$$\frac{1}{[19]} = [37]$$

oder auch

$$\frac{1}{19} = 37 \quad \text{in } \mathbb{Z}_{234}$$

Beispiel A.7. Ist $p \in \mathbb{Z}$ eine Primzahl, so ist

$$\mathbb{Z}_p = \{[0], [1], [2], \dots, [p-1]\}$$

und die Zahlen $1, 2, \dots, p-1$ sind teilerfremd zu p . Daher sind alle Äquivalenzklassen $[1], [2], \dots, [p-1]$ Einheiten in $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$. Die $[0]$ ist offensichtlich keine Einheit, und daher erhalten wir

$$\varphi(p) = p - 1$$

und alle Elemente aus $\mathbb{Z}_p \setminus \{0\}$ sind Einheiten,

$$E(\mathbb{Z}_p) = \{[1], [2], \dots, [p-1]\}$$

(dh. $\mathbb{Z}/p\mathbb{Z}$ ist ein **Körper**).

Beispiel A.8. Ist $n = p \cdot q$ das Produkt von zwei Primzahlen p und q mit $p \neq q$, so ist

$$\mathbb{Z}_n = \{[0], [1], [2], \dots, [n-1]\}$$

und die Zahlen aus $1, 2, \dots, n-1$ die nicht teilerfremd zu n sind, sind die Zahlen von der Form

$$r = a \cdot p \quad (a = 1, \dots, q-1) \quad \text{und} \quad s = b \cdot q \quad (b = 1, \dots, p-1)$$

die auch alle paarweise verschieden sind. Daher sind $(q-1) + (p-1)$ dieser Zahlen nicht teilerfremd zu n und also

$$p \cdot q - 1 - ((p-1) + (q-1)) = p \cdot q - p - q + 1 = (p-1) \cdot (q-1)$$

der Zahlen $1, 2, \dots, n-1$ teilerfremd zu $p \cdot q$. Deshalb erhalten wir

$$\varphi(p \cdot q) = (p-1) \cdot (q-1)$$

Beispiel A.9. Ist $p \in \mathbb{Z}$ eine Primzahl und $k \geq 1$, so ist

$$\mathbb{Z}_{p^k} = \{[0], [1], [2], \dots, [p^k-1]\}$$

und die Zahlen aus $1, 2, \dots, p^k-1$, die nicht teilerfremd zu p^k sind, sind die Zahlen von der Form

$$r = a \cdot p, \quad 1 \in \{1, \dots, p^{k-1}-1\}$$

und das sind $p^{k-1}-1$ viele. Damit sind also

$$p^k - 1 - (p^{k-1}-1) = p^k - p^{k-1} = (p-1) \cdot p^{k-1}$$

der Zahlen $1, 2, \dots, p^k - 1$ teilerfremd zu p^k . Deshalb erhalten wir

$$\varphi(p^k) = (p - 1) \cdot p^{k-1}$$

In diesem Fall ist bekannt, dass die Einheitengruppe $E(\mathbb{Z}_{p^k})$ immer zyklisch ist, dh. es gibt ein $g \in E(\mathbb{Z}_{p^k})$ mit

$$E(\mathbb{Z}_{p^k}) = \{g, g^2, g^3, \dots, g^{(p-1) \cdot p^{k-1}}\}$$

Beim modulo-Rechnen gibt es einige Regeln für das Rechnen mit Potenzen, die für kryptographische Protokolle sehr wesentlich sind, und mit denen wir uns nun etwas genauer beschäftigen wollen.

Satz A.3 (Frobenius-Identität). *Ist $p \in \mathbb{Z}$ eine Primzahl und $a \in \{0, \dots, p - 1\}$, so gilt*

$$a^p = a \bmod p$$

dh. in \mathbb{Z}_p gilt für jedes Element x :

$$x^p = x$$

Beweis: Von der Einheitengruppe $E(\mathbb{Z}_p)$ wissen wir, dass

$$E(\mathbb{Z}_p) = \{[1], [2], \dots, [p - 1]\}$$

eine Gruppe mit $p - 1$ Elementen ist. Daher gilt für jedes $x \in E(\mathbb{Z}_p)$:

$$x^{p-1} = [1]$$

nach dem Satz von Fermat, und daher

$$x^p = x \cdot x^{p-1} = x$$

Für $[0]$ schließlich gilt offensichtlich

$$[0]^p = [0]$$

und damit ist die Aussage nachgerechnet.

Satz A.4 (RSA-Identität). *Ist $N = p \cdot q$ das Produkt von zwei Primzahlen $p \neq q$, so gilt für jedes $a \in \{0, \dots, N - 1\}$ und jedes positive $t \in \mathbb{N}$:*

$$a^{t \cdot (p-1) \cdot (q-1) + 1} = a \bmod N$$

dh. in \mathbb{Z}_N gilt für jedes Element x :

$$x^{t \cdot \varphi(N) + 1} = x$$

Beweis: Die Aussage ist wieder klar für $a = 0$.

Für jedes a , das teilerfremd zu N ist, ist $[a]$ ein Element der Einheitengruppe von \mathbb{Z}_N , und daher gilt

$$[a]^{\varphi(N)} = [1]$$

nach dem Satz von Fermat, also auch

$$[a]^{t \cdot \varphi(N)} = [1]^t = [1]$$

und damit

$$[a]^{t \cdot (-1) \cdot (q-1)+1} = [a] \cdot [a]^{t \cdot \varphi(N)} = [a]$$

Wir müssen also nur noch die Elemente betrachten, die nicht teilerfremd zu N sind, die also die Form

$$a = r \cdot q \quad \text{oder} \quad a = s \cdot p$$

(mit $r \in \{1, \dots, p-1\}$ bzw. $s \in \{1, \dots, q-1\}$) haben.

Wir betrachten zunächst den Fall $a = r \cdot q$. Dann gilt nach dem Satz von Fermat

$$r^{t \cdot (q-1) \cdot (p-1)} = \left(r^{(p-1)}\right)^{t \cdot (q-1)} = 1 \pmod{p}$$

und

$$q^{t \cdot (q-1) \cdot (p-1)} = \left(q^{(p-1)}\right)^{t \cdot (q-1)} = 1 \pmod{p}$$

(da q und p teilerfremd). Wir können also schreiben

$$r^{t \cdot (q-1) \cdot (p-1)} = 1 + l_1 \cdot p, \quad q^{t \cdot (q-1) \cdot (p-1)} = 1 + l_2 \cdot p$$

Damit erhalten wir

$$\begin{aligned} a^{t \cdot (p-1) \cdot (q-1)+1} &= a \cdot a^{t \cdot (p-1) \cdot (q-1)} \\ &= r \cdot q \cdot (r \cdot q)^{t \cdot (p-1) \cdot (q-1)} \\ &= r \cdot q \cdot r^{t \cdot (q-1) \cdot (p-1)} \cdot q^{t \cdot (q-1) \cdot (p-1)} \\ &= r \cdot q \cdot (1 + l_1 \cdot p) \cdot (1 + l_2 \cdot p) \\ &= r \cdot q \cdot (1 + (l_1 + l_2) \cdot p + l_1 \cdot l_2 \cdot p^2) \\ &= r \cdot q + r \cdot (l_1 + l_2 + l_1 \cdot l_2 \cdot p) \cdot q \cdot p \\ &= r \cdot q \pmod{N} \\ &= a \pmod{N} \end{aligned}$$

Der Fall $a = s \cdot p$ geht dann vollkommen analog.

Die Aussage von Satz A.4 steht in engem Zusammenhang mit dem **chinesischen Restsatz (CRT)**:

Satz A.5 (Chinesischer Restsatz I). *Ist $N = p \cdot q$ das Produkt von zwei Primzahlen $p \neq q$, so gibt es für jedes $a \in \{0, \dots, p-1\}$ und jedes $b \in \{1, \dots, q-1\}$ genau ein $c \in \{0, \dots, N-1\}$ mit*

$$c \bmod p = a \bmod p, \quad c \bmod q = b \bmod q$$

Speziell gibt es daher eine bijektive Abbildung

$$\varepsilon : \mathbb{Z}_N \longrightarrow \mathbb{Z}_p \times \mathbb{Z}_q$$

mit $\varepsilon(x \bmod N) = (x \bmod p, x \bmod q)$.

Beweis: Da p und q voneinander verschieden sind, sind sie auch teilerfremd, und daher gibt es ganze Zahlen r, s mit

$$r \cdot p + s \cdot q = 1$$

Daraus folgt sofort

$$r \cdot p = 1 \bmod q, \quad s \cdot q = 1 \bmod p$$

Wir setzen nun

$$c' = b \cdot r \cdot p + a \cdot s \cdot q \in \mathbb{Z}$$

Dann ist möglicherweise $c' \geq N$ oder $c' \leq 0$, aber es gibt ein $c \in \{0, 1, \dots, n-1\}$ mit $c' \bmod N = c \bmod N$. Da $p|N$ und $q|N$ gilt dann auch

$$c \bmod p = c' \bmod p \quad \text{und} \quad c \bmod q = c' \bmod q$$

und wir erhalten

$$\begin{aligned} c \bmod q &= c' \bmod p \\ &= b \cdot r \cdot p + a \cdot s \cdot q \bmod p \\ &= a \cdot s \cdot q \bmod p \\ &= a \cdot 1 \bmod p \\ &= a \bmod p \end{aligned}$$

und genauso rechnen wir nach, dass $c \bmod q = b \bmod q$.

Es bleibt noch, nachzurechnen, dass c eindeutig ist: Angenommen, es gibt noch ein weiteres c^* mit

$$c^* \bmod p = a \bmod p, \quad c^* \bmod q = b \bmod q$$

Dann gilt natürlich $c^* \bmod p = c \bmod p$ und $c^* \bmod q = c \bmod q$ und daraus folgt, dass

$$p|(c^* - c) \quad \text{und} \quad q|(c^* - c)$$

Da p und q teilerfremd sind, heißt das aber, dass $p \cdot q|(c^* - c)$ also $N|(c^* - c)$ und daraus folgt, dass $c^* \geq N$ oder $c^* < 0$, im Widerspruch zu unserer Annahme.

Beispiel A.10. Wir betrachten die beiden Primzahlen $p = 37$ und $q = 17$ (also $n = p \cdot q = 629$) sowie $a = 19$ in $\mathbb{Z}/37 \cdot \mathbb{Z}$ und $b = 14$ in $\mathbb{Z}/17 \cdot \mathbb{Z}$. Dann ist

$$1 = 6 \cdot 37 - 13 \cdot 17$$

und wir setzen

$$c' = 14 \cdot 6 \cdot 37 - 19 \cdot 13 \cdot 17 = -1091$$

und $c = c' + 2 \cdot 629 = 167$. Dann gilt hierfür in der Tat

$$167 \bmod 37 = 19, \quad 167 \bmod 17 = 14$$

und c ist die einzige Zahl zwischen 0 und 628, die das erfüllt.

Aus der Begründung des chinesischen Restsatzes sehen wir sofort, dass die Aussage verallgemeinert werden kann. Die Aussage beruht nämlich auf der Tatsache, dass es Zahlen r und s gibt mit

$$1 = r \cdot p + s \cdot q$$

Dafür ist es aber nicht erforderlich, dass p und q zwei voneinander verschiedene Primzahlen sind. Auch für beliebige teilerfremde Zahlen m und n gibt es ganze Zahlen r und s mit

$$1 = r \cdot m + s \cdot n$$

und daraus erhalten wir sofort

Satz A.6 (Chinesischer Restsatz II). *Ist $N = n \cdot m$ das Produkt von zwei teilerfremden Zahlen n und m , so gibt es für jedes $a \in \{0, \dots, n-1\}$ und jedes $b \in \{1, \dots, m-1\}$ genau ein $c \in \{0, \dots, N-1\}$ mit*

$$c \bmod n = a \bmod n, \quad c \bmod m = b \bmod m$$

Speziell gibt es daher eine bijektive Abbildung

$$\varepsilon : \mathbb{Z}/N\mathbb{Z} \longrightarrow \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$$

mit $\varepsilon(x \bmod N) = (x \bmod n, x \bmod m)$.

In einigen kryptographischen Protokollen ist es notwendig, die Restklassen von sehr hohen Potenzen von sehr großen Zahlen modulo n zu betrachten. Die Potenzen sind dabei in der Regel so groß, dass es selbst mit der besten Technologie nicht möglich ist, diese Potenzen in \mathbb{Z} zu berechnen und dann die Restklasse modulo n zu bilden. Das Problem dieser extrem großen Zahlen, kann dadurch vermieden werden, dass die einzelnen Potenzoperationen unmittelbar modulo n durchzuführen und nicht erst x^k in \mathbb{Z} zu berechnen und dann die Restklasse modulo n zu bilden, also

$$x \bmod n \rightarrow x^2 = x \cdot x \bmod n \rightarrow x^3 = x^2 \cdot x \bmod n \rightarrow \dots \rightarrow x^n = x^{n-1} \cdot x \bmod n$$

(*naive Potenzierung*). Das führt aber zu sehr vielen einzelnen Multiplikationen. Soll etwa $x^{16} \bmod n$ berechnet werden, so sind in der Kette

$$x \bmod n \rightarrow x^2 \bmod n \rightarrow x^3 \bmod n \rightarrow \dots \rightarrow x^{15} \bmod n \rightarrow x^{16} \bmod n$$

insgesamt 15 Multiplikationen durchzuführen. Sehr viel schneller geht

$$\begin{aligned} x \bmod n &\rightarrow x^2 \bmod n \rightarrow x^4 \bmod n = (x^2)^2 \bmod n \\ &\rightarrow x^8 \bmod n = (x^4)^2 \bmod n \rightarrow x^{16} \bmod n = (x^8)^2 \bmod n \end{aligned}$$

Bei diesem Ansatz kommen wir mit vier Multiplikationen aus.

Ganz allgemein wird die Potenzierung durch iteriertes Quadrieren ersetzt, wobei nach jedem Quadrat bzw. nach jeder Produktoperation modulo n gerechnet wird. Das funktioniert auch, wenn der Exponent keine Zweierpotenz ist. Die Grundform dieses Verfahrens lässt sich am besten an einem Beispiel erläutern.

Beispiel A.11. Wir betrachten $x = 3427$, $e = 277$ und $n = 9047$ und berechnen die Zahl

$$a = x^e \bmod n$$

Dazu betrachten wir die Binärdarstellung von e ,

$$e = 256 + 16 + 4 + 1 = 2^8 + 2^4 + 2^2 + 2^0$$

Damit gilt

$$x^{277} = x^{2^8} \cdot x^{2^4} \cdot x^{2^2} \cdot x^{2^0} \bmod n$$

Für die einzelnen Faktoren gilt etwa

$$x^{2^4} = \left(\left((x^2)^2 \right)^2 \right)^2$$

und diese Quadrate werden jetzt einzeln modulo n berechnet. Dazu setzen wir

$$c_0 = x = 3427$$

und berechnen

$$\begin{aligned} c_1 &= c_0^2 = 3427^2 = 1323 \mod n \\ c_2 &= c_1^2 = 1323^2 = 4258 \mod n \\ c_3 &= c_2^2 = 4258^2 = 376 \mod n \\ c_4 &= c_3^2 = 376^2 = 5671 \mod n \\ c_5 &= c_4^2 = 5671^2 = 7203 \mod n \\ c_6 &= c_5^2 = 7203^2 = 7711 \mod n \\ c_7 &= c_6^2 = 7711^2 = 2637 \mod n \\ c_8 &= c_7^2 = 2637^2 = 5673 \mod n \end{aligned}$$

(sodass also jetzt $c_8 = x^{2^8} \mod n$). Damit gilt nun

$$a = c_8 \cdot c_4 \cdot c_2 \cdot c_0 \mod n$$

Auch dieses Produkt wird schrittweise berechnet und wir setzen

$$\begin{aligned} a_1 &= c_8 \cdot c_4 = 5673 \cdot 5671 = 451 \mod n \\ a_2 &= a_1 \cdot c_2 = 451 \cdot 4258 = 2394 \mod n \\ a_3 &= a_2 \cdot c_0 = 2394 \cdot 3427 = 7656 \mod n \end{aligned}$$

und wir haben

$$3427^{277} = 7656 \mod n$$

Bemerkung A.9. Die betrachteten Zahlen werden binär abgelegt und verarbeitet. Für das Quadrieren von Binärzahlen gibt es schnelle und effiziente Algorithmen (die sich im wesentlichen aus Shift–Operationen und Additionen zusammensetzen).

Bemerkung A.10. Das Prinzip des wiederholten Quadrierens liegt allen modernen Verfahren zur modularen Potenzberechnung zugrunde. Im Detail kann die Methode noch weiter optimiert werden. Beim **Square and Multiply**–Algorithmus kann durch geeignete zwischenzeitliche Multiplikationen auf die abschließende Multiplikation der einzelnen Potenzen verzichtet werden. Dadurch entfällt die Notwendigkeit, die einzelnen Potenzen (in Beispiel A.11 etwa c_4 , c_2 und c_0) zwischenzuspeichern.

Bemerkung A.11. Ist der Exponent eine 1024–Bit–Zahl, so sind bei der naiven Potenzierung bis zu $2^{1024} \approx 10^{300}$ Multiplikationen erforderlich. Die Methode des iterierten Quadrierens reduziert das auf maximal $2 \cdot 1024$ also 2048 Multiplikationen (im Mittel ist mit 1536 Multiplikationen zu rechnen).

B. Endliche Körper – Primkörper

Zu Wiederholung: Ein **Körper** ist eine nicht-leere Menge K mit zwei ausgezeichneten Elementen 0 und 1, wobei $0 \neq 1$, und mit zwei inneren Verknüpfungen (also Abbildungen) $+$ und \cdot .

$$\begin{aligned} + &: K \times K \longrightarrow K, & (a, b) &\longmapsto a + b \\ \cdot &: K \times K \longrightarrow K, & (a, b) &\longmapsto a \cdot b \end{aligned}$$

so dass gilt

1. $(K, +)$ ist eine kommutative Gruppe mit neutralem Element 0.
2. $(K \setminus \{0\}, \cdot)$ ist eine kommutative Gruppe mit neutralem Element 1.
3. Es gilt das Distributivgesetz, d.h. für alle $a, b, c \in K$ gilt

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

Beispiel B.1. Die Mengen \mathbb{Q} , \mathbb{R} und \mathbb{C} (mit den bekannten Additionen und Multiplikationen) sind Körper.

Die Menge \mathbb{Z} (mit der bekannten Addition und Subtraktion) ist ein Ring aber kein Körper, denn die Menge $(\mathbb{Z} \setminus \{0\}, \cdot)$ ist keine Gruppe.

Bemerkung B.1. Ein kommutativer Ring R ist genau dann ein Körper, wenn es zu jedem $x \in R \setminus \{0\}$ ein multiplikatives Inverses gibt, wenn also ein Element $y \in R$ existiert mit $x \cdot y = 1$.

Definition B.1. Ein Körper $(K, +, \cdot)$ heißt *endlicher Körper*, wenn $|K| < \infty$.

Beispiel B.2. Ist p eine Primzahl, so ist der Ring $\mathbb{Z}_p \mathbb{Z}/p\mathbb{Z}$ ein endlicher Körper, denn nach Beispiel A.7 ist jedes Element $x \in \mathbb{Z}_p \setminus \{0\}$ eine Einheit, hat also ein Inverses.

Notation:

Ist p eine Primzahl, so schreiben wir \mathbb{F}_p für den Körper $\mathbb{Z}/p\mathbb{Z}$. Ferner benutzen wir für die Restklassen einer ganzen Zahl a wieder die Bezeichnung $[a]$, schreiben also kurz

$$\mathbb{F}_p = \{0, 1, \dots, p - 1\}$$

(und verzichten auf die Darstellung als $[a]$).

Bezeichnung:

Für eine Primzahl p heißt der Körper \mathbb{F}_p **Primkörper** der Charakteristik p .

Bemerkung B.2. Addition, Subtraktion bzw. Multiplikation in \mathbb{F}_p sind klar und ergeben sich unmittelbar aus der Addition, Subtraktion bzw. Multiplikation der Repräsentanten. Komplizierter ist die Division, also die Berechnung von Elementen $c = \frac{a}{b}$ für $a, b \in \mathbb{F}_p$. Hierzu wird zunächst das Inverse $\frac{1}{b}$ von b mithilfe des euklidischen Algorithmus berechnet und dann $c = a \cdot \frac{1}{b}$ ermittelt.

Beispiel B.3. Wir berechnen im Körper \mathbb{F}_{79} die Elemente

$$a = \frac{1}{42}, \quad b = \frac{7}{40}$$

und drücken die Ergebnisse mit den Standardrepräsentanten $0, 1, \dots, 78$ aus. Beachten Sie dabei, dass wir in der Regel die Klammern $[]$ bei der Darstellung von Elementen von \mathbb{R}_p weglassen werden; aus dem Kontext ist immer klar, ob wir von ganzen Zahlen oder von Restklassen sprechen.

In beiden Fällen ist die Benutzung des erweiterten euklidischen Algorithmuses möglich:

$$\begin{aligned} 79 &= 1 \cdot 42 + 37 \\ 42 &= 1 \cdot 37 + 5 \\ 37 &= 7 \cdot 5 + 2 \\ 5 &= 2 \cdot 2 + 1 \\ 2 &= 2 \cdot 1 + 0 \end{aligned}$$

Rückrechnung ergibt

$$\begin{aligned} 1 &= 5 - 2 \cdot 2 = 5 - 2 \cdot (37 - 7 \cdot 5) = 15 \cdot 5 - 2 \cdot 37 \\ &= 15 \cdot (42 - 37) - 2 \cdot 37 = 15 \cdot 42 - 17 \cdot 37 = 15 \cdot 42 - 17 \cdot (79 - 42) \\ &= 32 \cdot 42 - 17 \cdot 79 \end{aligned}$$

also

$$1 = 32 \cdot 42 - 17 \cdot 79 \bmod 79 = 32 \cdot 42 \bmod 79$$

und damit

$$\frac{1}{42} = 32$$

Für b berechnen wir zunächst $\frac{1}{40}$:

$$\begin{aligned} 79 &= 1 \cdot 40 + 39 \\ 40 &= 1 \cdot 39 + 1 \\ 39 &= 39 \cdot 1 + 0 \end{aligned}$$

Rückrechnung ergibt

$$1 = 40 - 1 \cdot 39 = 40 - 1 \cdot (79 - 40) = 2 \cdot 40 - 79$$

also

$$\frac{1}{40} = 2$$

(das kann natürlich auch durch Ausprobieren bestimmt werden) und damit

$$b = 7 \cdot \frac{1}{40} = 7 \cdot 2 = 14$$

Beispiel B.4. Der Ring \mathbb{Z}_4 ist kein Körper, denn in \mathbb{Z}_4 gilt

$$2 \cdot 2 = 0$$

also ist 2 ein Nullteiler von \mathbb{Z}_4 und damit (nach A.7) keine Einheit.

Genauer gilt

Satz B.1. *Der Ring \mathbb{Z}_n ist genau dann ein Körper, wenn n eine Primzahl ist.*

Beweis: Das \mathbb{Z}_n ein Körper ist, wenn n eine Primzahl ist, haben wir schon gesehen. Ist nun n keine Primzahl, so können wir $n = r \cdot s$ mit ganzen Zahlen r, s und $1 < r, s < n$ schreiben. Dann sind die Restklassen $[r], [s] \in \mathbb{Z}/n\mathbb{Z}$ von Null verschiedene Elemente, aber

$$[r] \cdot [s] = [r \cdot s] = [n] = [0]$$

also ist $[r]$ ein Nullteiler und daher keine Einheit. Folglich ist $\mathbb{Z}/n\mathbb{Z}$ in diesem Fall kein Körper.

Für die Kryptologie ist das Arbeiten in endlichen Körpern sehr wichtig. Speziell ist es notwendig, Gleichungen über solchen Körpern zu lösen, vor allem quadratische Gleichungen, und Quadratwurzeln zu bestimmen.

Definition B.2. Ist K ein beliebiger Körper, und sind $a, b \in K$, so heißt a eine **Quadratwurzel** von b , wenn $a^2 = b$ (in K) gilt. In diesem Fall heißt das Element b ein **Quadrat** in K .

Für kleine Primzahlen p können die Quadratwurzeln durch Ausprobieren einfach bestimmt werden.

Beispiel B.5. In \mathbb{F}_7 gilt

$$\begin{array}{lll} 0^2 & = & 0 \\ 1^2 & = & 1 \\ 2^2 & = & 4 \\ 3^2 & = & 2 \end{array} \quad \begin{array}{lll} 4^2 & = & 2 \\ 5^2 & = & 4 \\ 6^2 & = & 1 \end{array}$$

also sind 0, 1, 2 und 4 Quadrate in \mathbb{F}_7 . Dabei ist 0 die einzige Quadratwurzel von 0, 1 und 6 sind die Quadratwurzeln von 1, 3 und 4 die Quadratwurzeln von 2 und 2 und 5 die Quadratwurzeln von 4.

In der Kryptologie wird aber mit sehr großen Primzahlen gearbeitet, ein Ausprobieren ist in diesem Fall nicht mehr möglich. Es gibt aber auch für große Primzahlen eine algorithmische Vorgehensweise. Grundlage hierfür ist ein Satz aus der Algebra:

Satz B.2 (Fermat). *Ist K ein beliebiger endlicher Körper mit q Elementen, so ist die Einheitengruppe $K^* = K \setminus \{0\}$ zyklisch von der Ordnung $q - 1$.*

Als unmittelbare Folgerung hieraus erhalten wir

Folgerung B.3. *Ist $p \geq 3$ eine Primzahl, so ist ein $x \in \mathbb{F}_p^*$ genau dann ein Quadrat, wenn $x^{\frac{p-1}{2}} = 1$.*

Genau die Hälfte der Elemente von \mathbb{F}_p^ sind Quadrate.*

Beweis: Ist x ein Quadrat, $x = a^2$, so gilt:

$$x^{\frac{p-1}{2}} = (a^2)^{\frac{p-1}{2}} = a^{2 \cdot \frac{p-1}{2}} = a^{p-1} = 1$$

nach dem Satz B.2 von Fermat.

Gilt umgekehrt $x^{\frac{p-1}{2}} = 1$, so wähle einen Erzeuger g von \mathbb{F}_p^* und schreibe

$$x = g^r$$

Dann gilt

$$1 = x^{\frac{p-1}{2}} = (g^r)^{\frac{p-1}{2}} = g^{\frac{r \cdot (p-1)}{2}}$$

Da g die Ordnung $p - 1$ hat, bedeutet das, dass $\frac{r \cdot (p-1)}{2}$ ein Vielfaches von $p - 1$ ist, und das wiederum bedeutet, dass r gerade sein muss, $r = 2s$. Setzen wir also $a = g^s$, so gilt hierfür

$$a^2 = (g^s)^2 = g^{2s} = g^r = x$$

und damit ist x ein Quadrat.

Da $a^2 = (-a)^2$ und da in \mathbb{F}_p wegen $p \geq 3$ gilt, dass $-a \neq a$ für jedes $a \in \mathbb{F}_p^*$, ergeben also immer zwei Zahlen in \mathbb{F}_p^* ein Quadrat, und damit sind genau die Hälfte der Elemente von \mathbb{F}_p^* Quadrate.

Damit erhalten wir schon ein erstes Ergebnis.

Folgerung B.4. *Ist p eine Primzahl, für die 4 ein Teiler von $p + 1$ ist, und ist $x \in \mathbb{F}_p^*$ ein Quadrat, so ist*

$$a = x^{\frac{p+1}{4}}$$

eine Quadratwurzel von x .

Beweis: Da x ein Quadrat ist, wissen wir schon, dass

$$x^{\frac{p-1}{2}} = 1$$

Damit gilt

$$a^2 = \left(x^{\frac{p+1}{4}}\right)^2 = x^{\frac{p+1}{2}} = x^{\frac{p-1}{2}} \cdot x = 1 \cdot x = x$$

Im Fall $4|(p+1)$ ist eine Quadratwurzel aus einer Zahl, die ein Quadrat ist, also leicht zu finden. Für den Fall $4 \nmid (p+1)$ ist das Vorgehen etwas komplizierter, hierfür gibt es aber auch einen (probabilistischen) Algorithmus:

Wir betrachten eine Primzahl p mit $4 \nmid (p+1)$ und ein $x \in \mathbb{F}_p$, das ein Quadrat ist, das also

$$x^{\frac{p-1}{2}} = 1 \quad (\text{B.1})$$

erfüllt. Zum Finden einer Wurzel von x gehen wir vor wie folgt:

Wähle ein $b \in \mathbb{F}_p^*$, das kein Quadrat ist, also ein b mit $b^{\frac{p-1}{2}} \neq 1$ (die Hälfte aller Elemente von \mathbb{F}_p^* erfüllt diese Bedingung, eine Zufallsauswahl sollte also in wenigen Schritten zu einem solchen Element führen). Beachte, dass dann notwendigerweise gilt, dass

$$b^{\frac{p-1}{2}} = -1$$

Da $4 \nmid (p+1)$, gilt notwendigerweise, dass $4|(p-1)$, und damit können wir schreiben

$$\frac{p-1}{2} = 2^l \cdot t$$

mit einem $l \geq 1$ und einem ungeraden t . Induktiv konstruieren wir zunächst Zahlen $n_0, n_1, \dots, n_l \in \mathbb{Z}$ mit den folgenden Eigenschaften

1. n_i ist $(l-i)$ -mal durch 2 teilbar.
2. $x^{2^{l-i} \cdot t} \cdot b^{2n_i} = 1$.

Induktionsanfang $i = 0$:

Setze $n_0 = 0$. Dann gilt hierfür:

1. n_0 ist beliebig oft, also sicherlich $(l-0)$ -mal durch 2 teilbar.
2. $x^{2^{l-0} \cdot t} \cdot b^{2n_0} = x^{2^l t} \cdot b^0 = x^{\frac{p-1}{2}} \cdot 1 = 1$ (wobei wir auch Gleichung (B.1) ausgenutzt haben).

Induktionsschluss $i \rightarrow i+1$ ($mit i+1 \leq l$)

Wir nehmen als Induktionsvoraussetzung an, dass wir schon ein n_i gefunden haben mit

1. n_i ist $(l - i)$ -mal durch 2 teilbar.

2. $x^{2^{l-i} \cdot t} \cdot b^{2n_i} = 1$.

Setze

$$c_i = x^{2^{l-(i+1)} \cdot t} \cdot b^{n_i}$$

Dann gilt

$$c_i^2 = (x^{2^{l-(i+1)} \cdot t} \cdot b^{n_i})^2 = x^{2 \cdot 2^{l-(i+1)} \cdot t} \cdot b^{2 \cdot n_i} = x^{2^{l-i}} \cdot b^{2n_i} = 1$$

(nach Induktionsvoraussetzung), und deshalb ist $c_i = 1$ oder $c_i = -1$.

Falls nun $c_i = 1$, so setze $n_{i+1} = \frac{n_i}{2}$. Das ist möglich, denn n_i ist $(l - i)$ -mal durch 2 teilbar und $i < i + 1 \leq l$. Dann gilt hierfür

1. n_{i+1} ist $(l - (i + 1))$ -mal durch 2 teilbar.

2. $x^{2^{l-(i+1)} \cdot t} \cdot b^{2n_{i+1}} = x^{2^{l-(i+1)}} \cdot b^{n_i} = c_i = 1$.

Falls dagegen $c_i = -1$, so setze

$$n_{i+1} = \frac{n_i}{2} + \frac{p-1}{4} = \frac{n_i}{2} + 2^{l-1} \cdot t$$

Dann gilt hierfür

1. n_{i+1} ist $(l - (i + 1))$ -mal durch 2 teilbar.

2. $x^{2^{l-(i+1)} \cdot t} \cdot b^{2n_{i+1}} = x^{2^{l-(i+1)}} \cdot b^{n_i} \cdot b^{\frac{p-1}{2}} = c_i \cdot b^{\frac{p-1}{2}} = (-1) \cdot (-1) = 1$.

Damit ist also der Induktionsschritt erfolgreich beendet und die Konstruktion abgeschlossen.

Setze nun $n = n_l$. Dann gilt hierfür nach Konstruktion:

$$x^t \cdot b^{2n} = x^{2^{l-l} \cdot t} \cdot b^{2n_l} = 1$$

woraus folgt

$$x^{t+1} \cdot b^{2n} = x \cdot x^t \cdot b^{2n} = x \cdot 1 = x$$

Da t ungerade ist, ist $t + 1$ gerade und daher existiert

$$a = x^{\frac{t+1}{2}} \cdot b^n$$

und hierfür gilt

$$a^2 = (x^{\frac{t+1}{2}} \cdot b^n)^2 = x^{t+1} \cdot b^{2n} = x$$

und damit ist a eine Quadratwurzel von x .

Der gesamte Algorithmus zur Konstruktion von Quadratwurzeln lässt sich damit wie folgt beschreiben:

Gegeben ist ein $x \in \mathbb{F}_p^*$.

1. Berechne $x^{\frac{p-1}{2}}$ in \mathbb{F}_p (durch iteriertes Quadrieren). Falls $x^{\frac{p-1}{2}} \neq 1$: STOPP, x ist kein Quadrat in F_p .

2. Falls $4|(p+1)$, so setze

$$a = x^{\frac{p+1}{4}}$$

Falls $4 \nmid (p+1)$, wähle ein $b \in \mathbb{F}_p$ mit $b^{\frac{p-1}{2}} = -1$, schreibe $\frac{p-1}{2} = 2^l \cdot t$ wie oben und konstruiere

$$a = x^{\frac{t+1}{2}} \cdot b^n$$

wie oben.

3. Die Zahl a ist eine Quadratwurzel von x in \mathbb{F}_p .

Beispiel B.6. Wir betrachten $p = 11$. In diesem Fall ist $p+1 = 12$ durch 4 teilbar mit $\frac{p+1}{4} = 3$.

a) Für $x = 2$ gilt

$$x^{\frac{p-1}{2}} = 2^5 = 10 = -1 \quad (\text{in } F_{11})$$

also ist x kein Quadrat \rightarrow STOPP.

b) Für $x = 5$ gilt

$$x^{\frac{p-1}{2}} = 5^5 = 1 \quad (\text{in } F_{11})$$

also ist x ein Quadrat und

$$a = x^{\frac{p+1}{4}} = 5^3 = 4 \quad (\text{in } F_{11})$$

ist eine Quadratwurzel aus x .

Die zweite Quadratwurzel aus 5 ist dann $-4 = 7$.

Beispiel B.7. Wir betrachten $p = 17$. In diesem Fall ist $p+1 = 18$ nicht durch 4 teilbar und $\frac{p-1}{2} = 8 = 2^3 \cdot 1$, also $l = 3$ und $t = 1$.

a) Für $x = 2$ gilt

$$x^{\frac{p-1}{2}} = 2^8 = 1 \quad (\text{in } F_{17})$$

also ist x ein Quadrat.

b) Zum Finden einer Wurzel wird noch ein $b \in \mathbb{F}_{17}$ benötigt, das kein Quadrat ist.

Wir prüfen $b = 3$:

$$b^{\frac{p-1}{2}} = 3^8 = 9^4 = 13^2 = 16 = -1 \quad (\text{in } F_{17})$$

also ist b kein Quadrat.

Setze $n_0 = 0$. Dann ist

$$c_0 = x^{2^{3-1}} \cdot b^0 = 2^4 \cdot 1 = 16 = -1$$

also setzen wir

$$n_1 = \frac{n_0}{2} + \frac{p-1}{4} = 0 + 4 = 4$$

Dann ist

$$c_1 = x^{2^{3-2}} \cdot b^4 = 2^2 \cdot 3^4 = 1$$

also ist

$$n_2 = \frac{n_1}{2} = 2$$

Dann ist

$$c_2 = x^{2^{3-3}} \cdot b^2 = 2^1 \cdot 3^2 = 1$$

also ist

$$n_3 = \frac{n_2}{2} = 1$$

Damit ist die Konstruktion beendet und wir setzen $n = 1$ und erhalten, dass

$$a = 2^{\frac{t+1}{2}} \cdot b^n = 2^1 \cdot 3^1 = 6$$

eine Wurzel von $x = 2$ ist. In der Tat gilt

$$6^2 = 2 \quad \text{in } \mathbb{F}_{17}$$

Die zweite Quadratwurzel aus 2 ist dann $-6 = 11$.

Beispiel B.8. Wir betrachten $p = 13$. In diesem Fall ist $p+1 = 14$ nicht durch 4 teilbar und $\frac{p-1}{2} = 2^1 \cdot 3$, also $l = 1$ und $t = 3$.

a) Für $x = 2$ gilt

$$x^{\frac{p-1}{2}} = 2^6 = 12 = -1 \quad (\text{in } F_{13})$$

also ist x kein Quadrat \rightarrow STOPP.

b) Für $x = 10$ gilt

$$x^{\frac{p-1}{2}} = 10^6 = 1 \quad (\text{in } F_{13})$$

also ist x ein Quadrat.

Wir wählen $b = 2$, denn nach Teil a) wissen wir schon, dass b kein Quadrat ist, und wir setzen $n_0 = 0$ und

$$c_0 = x^{2^{1-1 \cdot 3}} \cdot 2^0 = 10^3 = 12 = -1 \quad (\text{in } F_{13})$$

Daher ist

$$n = n_1 = \frac{n_0}{2} + \frac{p-1}{4} = 3$$

und wir setzen

$$a = x^{\frac{t+1}{2}} \cdot b^n = 10^2 \cdot 2^3 = 7 \quad (\text{in } \mathbb{F}_{13})$$

Hierfür gilt in der Tat

$$a^2 = 7^2 = 10 \quad (\text{in } \mathbb{F}_{13})$$

Die zweite Quadratwurzel aus 10 ist $-7 = 4$.

Für Primzahlen $p \geq 3$ können auch quadratische Gleichungen über \mathbb{F}_p gelöst werden.

Die abc -Formel überträgt sich hier unmittelbar auf diese Situation

Regel B.5. ist $p \geq 3$ so hat die quadratische Gleichung

$$ax^2 + bx + c = 0$$

über \mathbb{F}_p genau dann eine Lösung, wenn $b^2 - 4ac$ ein Quadrat in \mathbb{F}_p ist. Ist in diesem Fall $\sqrt{b^2 - 4ac}$ eine Quadratwurzel von $b^2 - 4ac$, so hat die Gleichung die Lösungen

$$x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Beispiel B.9. Wir betrachten $p = 11$ und die quadratische Gleichung

$$x^2 + 4x + 1 = 0$$

(also $a = 1$, $b = 4$ und $c = 1$) über \mathbb{F}_{11} . In diesem Fall ist

$$b^2 - 4ac = 4^2 - 4 \cdot 1 \cdot 1 = 5 - 4 = 1$$

ein Quadrat und 1 ist eine Quadratwurzel daraus. Damit hat die Gleichung die beiden Lösungen

$$x_1 = \frac{-4 - 1}{2} = \frac{6}{2} = 3$$

und

$$x_2 = \frac{-4 + 1}{2} = \frac{8}{2} = 4$$

Beispiel B.10. Wir betrachten wieder $p = 11$ und die quadratische Gleichung

$$2x^2 + 3x + 5 = 0$$

(also $a = 2$, $b = 3$ und $c = 5$) über \mathbb{F}_{11} . Dann ist

$$b^2 - 4ac = 3^2 - 4 \cdot 2 \cdot 5 = 9 - 7 = 2$$

und 2 ist kein Quadrat in \mathbb{F}_{11} wie wir schon in Beispiel B.6 gesehen haben.

Also hat auch diese quadratische Gleichung keine Lösungen in \mathbb{F}_{11} .

C. Endliche Körper – Erweiterungskörper

Wir haben die Körper \mathbb{F}_p (p eine Primzahl) kennengelernt. Es gibt jedoch noch viele weitere endliche Körper, die für die Kryptologie ebenfalls wichtig sind.

Beispiel C.1. Wir betrachten die Menge $M = \{0, 1, \alpha, \alpha + 1\}$ mit Addition und Multiplikation, die durch die folgenden Tafeln gegeben sind:

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

bzw.

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

Dann ist $(M, +, \cdot)$ ein endlicher Körper mit 4 Elementen, den wir mit \mathbb{F}_4 bezeichnen.

Dabei ist allerdings \mathbb{F}_4 nicht isomorph zum Ring $\mathbb{Z}/4\mathbb{Z}$, denn $\mathbb{Z}/4\mathbb{Z}$ ist ja kein Körper, wie wir schon wissen.

Ist K ein (endlicher oder unendlicher) Körper, so schreiben wir kurz

$$n = n \cdot 1 = \underbrace{1 + 1 + \cdots + 1}_{n\text{-mal}}$$

und für $a \in K$ beliebig

$$n \cdot a = \underbrace{a + a + \cdots + a}_{n\text{-mal}}$$

Bemerkung C.1. Ist K ein endlicher Körper, so gibt es immer eine Zahl $n \in \mathbb{N}$, $n > 0$ mit

$$n = 0 \quad \text{in } K$$

Da K endlich ist, muss es nämlich $r > 0$ und $s > 0$ mit $r \neq s$ geben mit

$$r \cdot 1 = s \cdot 1$$

wobei wir annehmen können, dass $s > r$ (andernfalls vertauschen wir die Rollen von r und s). Dann gilt aber

$$0 = s \cdot 1 - r \cdot 1 = \underbrace{1 + 1 + \cdots + 1}_{s-r\text{-mal}}$$

Setzen wir also $n = s - r$, so haben wir das gesuchte n gefunden.

Definition C.1. Ist K ein endlicher Körper, so heißt das kleinste $n \in \mathbb{N}$, $n > 0$ mit

$$n \cdot 1 = 0 \quad \text{in } K$$

die **Charakteristik** von K .

Ist n die Charakteristik von K , so schreiben wir

$$\text{char}(K) = n$$

Beispiel C.2. Für jede Primzahl p hat der Körper \mathbb{F}_p die Charakteristik p .

Satz C.1. Ist K ein endlicher Körper, so ist $\text{char}(K)$ eine Primzahl.

Beweis: Wir schreiben $n = \text{char}(K)$ und nehmen an, n ist keine Primzahl, also $n = r \cdot s$ mit echten Teilern r und s . Dann gilt

$$r \cdot 1 \neq 0 \quad s \cdot 1 \neq 0$$

da $r, s < n$, aber

$$(r \cdot 1) \cdot (s \cdot 1) = rs \cdot 1 = n \cdot 1 = 0$$

Das ist ein Widerspruch zur Nullteilerfreiheit von K , und damit ist n eine Primzahl.

Ist nun $p = \text{char}(K)$, so können wir \mathbb{F}_p als Teilmenge von K betrachten, wenn wir $n \in \mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ mit

$$\underbrace{1 + \cdots + 1}_{n\text{-mal}} = n \cdot 1 \in K$$

identifizieren. Wegen $p \cdot 1 = 0$ ist das unabhängig vom Repräsentanten n (zunächst nur für $n > 0$, aber wie man leicht sieht geht das auch für $n < 0$). Dadurch wird \mathbb{F}_p sogar zu einem Teilkörper (Unterkörper) von K .

Definition C.2. Ist K ein endlicher Körper der Charakteristik $p > 0$, so heißt \mathbb{F}_p der Primkörper von K .

Bemerkung C.2. Ist K ein endlicher Körper der Charakteristik $p > 0$, so ist K ein \mathbb{F}_p -Vektorraum. Wir wissen bereits, dass $\mathbb{F}_p \subseteq K$ ein Unterkörper ist. Die Vektoraddition

von K ist dabei die übliche Addition im Körper K , und die Skalarmultiplikation entsteht durch Einschränkung der Körpermultiplikation auf \mathbb{F}_p , d.h. $[n] \cdot r = (n \cdot 1) \cdot r$.

Damit gilt insbesondere, dass es ein $l \in \mathbb{N}$ gibt, so dass

$$|K| = p^l$$

und dabei ist l die Dimension von K als \mathbb{F}_p -Vektorraum.

Außer den Körpern \mathbb{F}_p kennen wir bis jetzt nur einen endlichen Körper, nämlich den Körper mit 4 Elementen aus Beispiel C.1. Der folgende Satz liefert uns nicht nur eine Fülle solcher Körper sondern auch noch eine Methodik, deren Arithmetik mit Hilfe der Arithmetik der Körper \mathbb{F}_p zu beschreiben.

Satz C.2. *Ist $q = p^l$ so gibt es einen (bis auf Isomorphie eindeutigen) Körper \mathbb{F}_q mit q Elementen. Dabei hat \mathbb{F}_q die Charakteristik $p > 0$ und es gibt ein $\alpha \in \mathbb{F}_q$ so dass $1, \alpha, \alpha^2, \dots, \alpha^{l-1}$ ist eine Basis von K als \mathbb{F}_p -Vektorraum ist. Insbesondere haben wir also eine Relation*

$$\alpha^l = r_{l-1} \cdot \alpha^{l-1} + r_{l-2} \cdot \alpha^{l-2} + \dots + r_1 \cdot \alpha + r_0$$

Definition C.3. Eine Relation

$$\alpha^l = r_{l-1} \cdot \alpha^{l-1} + r_{l-2} \cdot \alpha^{l-2} + \dots + r_1 \cdot \alpha + r_0$$

heißt **definierende Relation** des Körpers \mathbb{F}_q .

Regel C.3. *Mit Hilfe einer definierenden Relation lassen sich die Addition und die Multiplikation im Körper \mathbb{F}_q vollständig beschreiben:*

- Ist $x \in \mathbb{F}_q$ beliebig, so gibt es eindeutig bestimmte Elemente $a_0, \dots, a_{l-1} \in \mathbb{F}_p$ mit

$$x = a_{l-1} \cdot \alpha^{l-1} + \dots + a_1 \cdot \alpha + a_0$$

- Sind $x = a_{l-1} \cdot \alpha^{l-1} + \dots + a_1 \cdot \alpha + a_0$ und $y = b_{l-1} \cdot \alpha^{l-1} + \dots + b_1 \cdot \alpha + b_0$ zwei Elemente in \mathbb{F}_q , so gilt

$$x + y = (a_{l-1} + b_{l-1}) \cdot \alpha^{l-1} + (a_{l-2} + b_{l-2}) \cdot \alpha^{l-2} + \dots + (a_1 + b_1) \cdot \alpha + a_0 + b_0$$

- Die Multiplikation in \mathbb{F}_q ist gegeben durch

$$\alpha^i \cdot \alpha^j = \alpha^{i+j}$$

unter Ausnutzung der definierenden Relation, d.h. immer wenn $i + j \geq l$, so wird ein α^l durch $r_{l-1} \cdot \alpha^{l-1} + r_{l-2} \cdot \alpha^{l-2} + \dots + r_1 \cdot \alpha + r_0$ ersetzt.

Beispiel C.3. Der Körper \mathbb{F}_4 aus Beispiel C.1 wird gegeben durch die defnierende Relation

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

Beispiel C.4. Der Körper \mathbb{F}_8 kann definiert werden durch die Relation

$$\alpha^3 = \alpha + 1$$

so dass also

$$\mathbb{F}_8 = \{0, 1, \alpha, \alpha + 1, \alpha^2, \alpha^2 + 1, \alpha^2 + \alpha, \alpha^2 + \alpha + 1\}$$

mit den Rechenvorschriften

+	0	1	α	$\alpha + 1$	α^2	$\alpha^2 + 1$	$\alpha^2 + \alpha$	$\alpha^2 + \alpha + 1$
0	0	1	α	$\alpha + 1$	α^2	$\alpha^2 + 1$	$\alpha^2 + \alpha$	$\alpha^2 + \alpha + 1$
1	1	0	$\alpha + 1$	α	$\alpha^2 + 1$	α^2	$\alpha^2 + \alpha + 1$	$\alpha^2 + \alpha$
α	α	$\alpha + 1$	0	1	$\alpha^2 + \alpha$	$\alpha^2 + \alpha + 1$	α^2	$\alpha^2 + 1$
$\alpha + 1$	$\alpha + 1$	α	1	0	$\alpha^2 + \alpha + 1$	$\alpha^2 + \alpha$	$\alpha^2 + 1$	α^2
α^2	α^2	$\alpha^2 + 1$	$\alpha^2 + \alpha$	$\alpha^2 + \alpha + 1$	0	1	α	$\alpha + 1$
$\alpha^2 + 1$	$\alpha^2 + 1$	α^2	$\alpha^2 + \alpha + 1$	$\alpha^2 + \alpha$	1	0	$\alpha + 1$	α
$\alpha^2 + \alpha$	$\alpha^2 + \alpha$	$\alpha^2 + \alpha + 1$	α^2	$\alpha^2 + 1$	α	$\alpha + 1$	0	1
$\alpha^2 + \alpha + 1$	$\alpha^2 + \alpha + 1$	$\alpha^2 + \alpha$	$\alpha^2 + 1$	α^2	$\alpha + 1$	α	1	0

und

.	1	α	$\alpha + 1$	α^2	$\alpha^2 + 1$	$\alpha^2 + \alpha$	$\alpha^2 + \alpha + 1$
1	1	α	$\alpha + 1$	α^2	$\alpha^2 + 1$	$\alpha^2 + \alpha$	$\alpha^2 + \alpha + 1$
α	α	α^2	$\alpha^2 + \alpha$	$\alpha + 1$	1	$\alpha^2 + \alpha + 1$	$\alpha^2 + 1$
$\alpha + 1$	$\alpha + 1$	$\alpha^2 + \alpha$	$\alpha^2 + 1$	$\alpha^2 + \alpha + 1$	α^2	1	α
α^2	α^2	$\alpha + 1$	$\alpha^2 + \alpha + 1$	$\alpha^2 + \alpha$	α	$\alpha^2 + 1$	1
$\alpha^2 + 1$	$\alpha^2 + 1$	1	α^2	α	$\alpha^2 + \alpha + 1$	$\alpha + 1$	$\alpha^2 + \alpha$
$\alpha^2 + \alpha$	$\alpha^2 + \alpha$	$\alpha^2 + \alpha + 1$	1	$\alpha^2 + 1$	$\alpha + 1$	α	α^2
$\alpha^2 + \alpha + 1$	$\alpha^2 + \alpha + 1$	$\alpha^2 + 1$	α	1	$\alpha^2 + \alpha$	α^2	$\alpha + 1$

Dabei ergeben sich die Formeln etwa wie folgt

$$\begin{aligned}
 \alpha^2 \cdot (\alpha^2 + 1) &= \alpha^4 + \alpha^2 \\
 &= \alpha \cdot \alpha^3 + \alpha^2 \\
 &= \alpha \cdot (\alpha + 1) + \alpha^2 \\
 &= \alpha^2 + \alpha + \alpha^2 \\
 &= 2 \cdot \alpha^2 + \alpha \\
 &= \alpha
 \end{aligned}$$

wobei wir auch noch ausgenutzt haben, dass $2 = 0$ im Primkörper \mathbb{F}_2 .

Beispiel C.5. Der Körper \mathbb{F}_8 kann auch definiert werden durch die Relation

$$\alpha^3 = \alpha^2 + 1$$

Eine definierende Relation ist also nicht eindeutig. Allerdings kann eine definierende Relation auch nicht beliebig sein, denn die Relation

$$\alpha^3 = \alpha^2 + \alpha + 1$$

etwa definiert den Körper \mathbb{F}_8 nicht. Entscheidend ist dabei immer, ob durch die Relation eine Multiplikation erklärt wird, für die gilt

$$x \cdot y \neq 0 \quad \text{wenn immer } x \neq 0 \text{ und } y \neq 0$$

Bei der Relation $\alpha^3 = \alpha^2 + 1$ ist das der Fall, bei der Relation $\alpha^3 = \alpha^2 + \alpha + 1$ jedoch nicht, denn bei dieser Relation gilt

$$\begin{aligned} (\alpha + 1) \cdot (\alpha^2 + 1) &= \alpha^3 + \alpha^2 + \alpha + 1 \\ &= \alpha^2 + \alpha + 1 + \alpha^2 + \alpha + 1 \\ &= 0 \end{aligned}$$

In diesem Fall wäre also das Produkt der beiden von Null verschiedenen Elementen $\alpha + 1$ und $\alpha^2 + 1$ gleich 0, was in einem Körper nicht sein kann.

Bemerkung C.3. Die Relation

$$\alpha^l = r_{l-1} \cdot \alpha^{l-1} + r_{l-2} \cdot \alpha^{l-2} + \cdots + r_1 \cdot \alpha + r_0$$

beschreibt genau dann den Körper \mathbb{F}_q (mit $q = p^l$), wenn sie **irreduzibel** ist:

Dazu schreiben wir zunächst die Relation als

$$\alpha^l - r_{l-1}\alpha^{l-1} - \cdots - r_1 \cdot \alpha - r_0$$

ersetzen die α durch eine Polynomvariable X und erhalten

$$f(X) = X^l - r_{l-1} \cdot X^{l-1} - \cdots - r_1 \cdot X - r_0 \in \mathbb{F}_p[X]$$

Die Relation heißt dann irreduzibel, wenn es keine Polynome $g(X)$ und $h(X)$ vom Grad höchstens $l - 1$ gibt mit

$$f(X) = g(X) \cdot h(X)$$

(wobei das Polynomprodukt über \mathbb{F}_p zu berechnen ist).

Definition C.4. Ist

$$\alpha^l = r_{l-1} \cdot \alpha^{l-1} + r_{l-2} \cdot \alpha^{l-2} + \cdots + r_1 \cdot \alpha + r_0$$

eine Relation, die den Körper \mathbb{F}_q (mit $q = p^l$) beschreibt und ist

$$f(X) = X^l - r_{l-1} \cdot X^{l-1} - \cdots - r_1 \cdot X - r_0 \in \mathbb{F}_p[X]$$

das (wie oben) daraus abgeleitete Polynom, so heißt $f(X)$ **Minimalpolynom** von $\mathbb{F}_q/\mathbb{F}_p$.

Für Relationen niedrigen Grades lässt sich sehr leicht überprüfen, ob eine Relation die definierende Relation eines Körpers ist:

Eine Relation

$$\alpha^2 = r_1 \cdot \alpha + r_0$$

definiert genau dann den Körper \mathbb{F}_{p^2} , wenn das Polynom $f(X) = X^2 - r_1X - r_0$ keine Nullstelle in \mathbb{F}_p hat. Die Beziehung $\alpha^2 = \alpha + 1$ definiert daher \mathbb{F}_4 über \mathbb{F}_2 , denn für $f(X) = X^2 + X + 1$ gilt:

$$f(0) = 1 \neq 0, \quad f(1) = 1 \neq 0$$

Dagegen definiert $\alpha^2 = 1$, den Körper \mathbb{F}_4 über \mathbb{F}_2 nicht, denn für $f(X) = X^2 + 1$ gilt:

$$f(1) = 0$$

Eine Relation

$$\alpha^3 = r_2 \cdot \alpha^2 + r_1 \cdot \alpha + r_0$$

definiert genau dann den Körper \mathbb{F}_{p^3} , wenn das daraus abgeleitete Polynom $f(X) = X^3 - r_2 \cdot X^2 - r_1 \cdot X - r_0$ keine Nullstelle in \mathbb{F}_p hat. Die Beziehung $\alpha^3 = \alpha^2 + 1$ definiert daher \mathbb{F}_8 über \mathbb{F}_2 , denn für $f(X) = X^3 + X^2 + 1$ gilt:

$$f(0) = 1 \neq 0, \quad f(1) = 1 \neq 0$$

Dagegen definiert $\alpha^3 = \alpha^2 + \alpha + 1$, den Körper \mathbb{F}_8 über \mathbb{F}_2 nicht, denn für $f(X) = X^3 + X^2 + X + 1$ gilt:

$$f(1) = 0$$

Bemerkung C.4. Der Beweis von Satz C.2, insbesondere der Nachweis der Existenz und der Eindeutigkeit (in einem geeigneten Sinn) sowie der Existenz eines Elements α wie behauptet übersteigt den Rahmen dieser Veranstaltung. Ist jedoch ein Element

α gefunden, so dass $1, \alpha, \dots, \alpha^{l-1}$ eine \mathbb{F}_p -Basis von \mathbb{F}_q ist, so gibt es, da ja $\alpha^l \in \mathbb{F}_q$, notwendigerweise $r_0, \dots, r_{l-1} \in \mathbb{F}_p$ mit

$$\alpha^l = r_{l-1} \cdot \alpha^{l-1} + r_{l-2} \cdot \alpha^{l-2} + \dots + r_1 \cdot \alpha + r_0$$

da sich α^l mit Hilfe der Basis darstellen lassen muss. Die Regeln für die Addition und die Multiplikation ergeben sich daraus und aus den Vektorraumeigenschaften.

Bemerkung C.5. Obwohl es (bis auf Isomorphie) nur einen Körper mit $q = p^l$ Elementen gibt, kann dieser in der Regel durch viele definierende Relation beschrieben werden. Die Umrechnung von einer Darstellung in eine andere (also der Wechsel von einem α zu einem anderen) ist dabei nicht offensichtlich.

Wir betrachten nun wieder einen Körper \mathbb{F}_q mit $q = p^l$ Elementen.

Satz C.4. *Der Körper \mathbb{F}_q kann beschrieben werden durch ein α mit definierender Relation*

$$\alpha^l = r_{l-1} \cdot \alpha^{l-1} + r_{l-2} \cdot \alpha^{l-2} + \dots + r_1 \cdot \alpha + r_0$$

so dass

$$\mathbb{F}_q \setminus \{0\} = \{\alpha, \alpha^2, \dots, \alpha^{q-2}, \alpha^{q-1} = 1\}$$

Auch ein Nachweis dieses Aussage übersteigt den Rahmen dieser Veranstaltung.

Der wichtigste Körper für die digitale Datenverarbeitung ist der Körper \mathbb{F}_{256} mit $2^8 = 256$ Elementen (also mit den Bytes als Elementen). Für diesen Körper gibt es verschiedene Beschreibungen, die in der Praxis verwendet werden. In der Codierungstheorie benutzt man etwa die Beschreibung durch die Relation

$$\alpha^8 = \alpha^4 + \alpha^3 + \alpha^2 + 1 \tag{C.1}$$

für das Verschlüsselungsverfahren AES dagegen wird mit der sogenannten Rijndael-Relation

$$\alpha^8 = \alpha^4 + \alpha^3 + \alpha + 1 \tag{C.2}$$

gearbeitet. Es handelt sich dabei beide Male um den gleichen Körper, es wird lediglich eine andere Sicht darauf gegeben.

Wir überlassen es dem Leser, sich davon zu überzeugen, dass das sowohl das Polynom

$$f_1(X) = X^8 - X^4 - X^3 - X^2 - 1 = X^8 + X^4 + X^3 + X^2 + 1 \in \mathbb{F}_2[X]$$

als auch das Polynom

$$f_2(X) = X^8 - X^4 - X^3 - X - 1 = X^8 + X^4 + X^3 + X + 1 \in \mathbb{F}_2[X]$$

irreduzibel ist, sich also nicht als Produkt von zwei Polynomen schreiben lässt. Damit definieren in der Tat beide Polynome den Körper \mathbb{F}_{256} .

Eine entscheidende Eigenschaft eines Körpers K ist, dass es zu jedem Element $r \in K \setminus \{0\}$ ein Inverses s bezüglich der Multiplikation gibt, also ein Element $s \in K \setminus \{0\}$ mit $r \cdot s = 1$. Für dieses Element s schreiben wir auch r^{-1} und nennen es das inverse Element zu r . In endlichen Körpern kann dieses inverse Element am einfachsten durch Potenzbilden gefunden werden:

Satz C.5. Ist K ein endlicher Körper mit $q = p^n$ Elementen, und ist $r \in K \setminus \{0\}$, so gilt

$$r^{q-1} = 1$$

Also gilt speziell für das Element $s = r^{q-2}$:

$$r \cdot s = 1$$

d.h.

$$r^{-1} = r^{q-2}$$

Beweis: Da K ein Körper (mit q Elementen) ist, ist $K \setminus \{0\}$ bezüglich der Multiplikation eine Gruppe der Ordnung $q - 1$. Nach den allgemeinen Regeln aus der Gruppentheorie teilt damit die Ordnung eines jeden Elements aus $K \setminus \{0\}$ die Ordnung dieser Gruppe, also $q - 1$, und daraus ergibt sich diese Aussage.

Beispiel C.6. Wir betrachten den Körper $K = \mathbb{F}_8$, gegeben durch die Relation $\alpha^3 = \alpha^2 + 1$ und das Element

$$r = \frac{1}{\alpha + 1} = (0, 1, 1)^{-1}$$

Dann gilt

$$r = (\alpha + 1)^{8-2} = (\alpha + 1)^6 = ((\alpha + 1)^2)^2 \cdot (\alpha + 1)^2$$

wobei

$$\begin{aligned} (\alpha + 1)^2 &= \alpha^2 + 1 \\ ((\alpha + 1)^2)^2 &= (\alpha^2 + 1)^2 = \alpha^4 + 1 \\ &= \alpha^3 + \alpha + 1 = \alpha^2 + 1 + \alpha + 1 \\ &= \alpha^2 + \alpha \end{aligned}$$

also

$$\begin{aligned} r &= (\alpha^2 + \alpha) \cdot (\alpha^2 + 1) = \alpha^4 + \alpha^2 + \alpha^3 + \alpha \\ &= \alpha^3 + \alpha + \alpha^2 + \alpha^2 + 1 + \alpha = \alpha^3 + 1 \\ &= \alpha^2 + 1 + 1 = \alpha^2 \\ &= (1, 0, 0) \end{aligned}$$

Wird die Arithmetik eines endlichen Körpers sehr oft ausgenutzt und sind viele Gleichungssysteme zu lösen, so kann die Berechnung der inversen Elemente durch Ausnutzung von Satz C.4 weiter stark vereinfacht werden. Die Relation $\alpha^3 = \alpha^2 + 1$, die den Körper \mathbb{F}_8 definiert etwa erfüllt

$$\mathbb{F}_8 \setminus \{0\} = \{\alpha, \alpha^2, \dots, \alpha^7 = 1\}$$

und eine einfache Rechnung ergibt

$$\begin{aligned}\alpha &= \alpha \\ \alpha^2 &= \alpha^2 \\ \alpha^3 &= \alpha^2 + 1 \\ \alpha^4 &= \alpha^2 + \alpha + 1 \\ \alpha^5 &= \alpha + 1 \\ \alpha^6 &= \alpha^2 + \alpha \\ \alpha^7 &= 1\end{aligned}$$

Daraus lesen wir nun unmittelbar ab

$$\frac{1}{\alpha+1} = \frac{1}{\alpha^5} = \frac{\alpha^7}{\alpha^5} = \alpha^2$$

Auch Division lassen sich damit leicht durchführen, etwa

$$\frac{\alpha+1}{\alpha^2+1} = \frac{\alpha^5}{\alpha^3} = \alpha^2$$

oder

$$\frac{\alpha^2+\alpha+1}{\alpha^2+\alpha} = \frac{\alpha^4}{\alpha^6} = \frac{1}{\alpha^2} = \frac{\alpha^7}{\alpha^2} = \alpha^5 = \alpha + 1$$

Beispiel C.7. Wir betrachten wieder den Körper $K = \mathbb{F}_8$, diesmal aber gegeben durch die Relation $\alpha^3 = \alpha + 1$. Auch hier gilt

$$\mathbb{F}_8 \setminus \{0\} = \{\alpha, \alpha^2, \dots, \alpha^7 = 1\}$$

und in diesem Fall rechnen wir nach, dass

$$\begin{aligned}\alpha &= \alpha \\ \alpha^2 &= \alpha^2 \\ \alpha^3 &= \alpha + 1 \\ \alpha^4 &= \alpha^2 + \alpha \\ \alpha^5 &= \alpha^2 + \alpha + 1 \\ \alpha^6 &= \alpha^2 + 1 \\ \alpha^7 &= 1\end{aligned}$$

Daher gilt in dieser Situation

$$\frac{1}{\alpha+1} = \frac{1}{\alpha^3} = \frac{\alpha^7}{\alpha^3} = \alpha^4 = \alpha^2 + \alpha$$

und für die Division erhalten wir

$$\frac{\alpha+1}{\alpha^2+1} = \frac{\alpha^3}{\alpha^6} = \frac{1}{\alpha^3} = \frac{\alpha^7}{\alpha^3} = \alpha^4 = \alpha^2 + \alpha$$

oder

$$\frac{\alpha^2 + \alpha + 1}{\alpha^2 + \alpha} = \frac{\alpha^5}{\alpha^4} = \alpha$$

Die explizite Arithmetik eines endlichen Körpers hängt also stark davon ab, welches α bzw. welche Erzeugerrelation wir wählen. Zu beachten ist aber, dass der Körper insgesamt eindeutig ist, dass es also nur einen Körper mit acht Elementen gibt (wenn auch mit unterschiedlichen Beschreibungen). Definieren wir etwa \mathbb{F}_8 durch $\alpha^3 = \alpha + 1$, so gilt für das Element $\tilde{\alpha} = \alpha^3$ in diesem Körper die Relation

$$\tilde{\alpha}^3 = \alpha^9 = \alpha^2 = (\alpha^2 + 1) + 1 = \alpha^6 + 1 = \tilde{\alpha}^2 + 1$$

Damit erfüllt also das Element $\tilde{\alpha} = \alpha^3$ in diesem Körper die Relation $\tilde{\alpha}^3 = \tilde{\alpha}^2 + 1$ und definiert ebenfalls \mathbb{F}_8 .

Beispiel C.8. Der Körper \mathbb{F}_{16} kann beschrieben werden durch die Relation

$$\alpha^4 = \alpha + 1$$

und auch bei dieser Relation ist α ein erzeugendes Element der Einheitengruppe von \mathbb{F}_{16} .

In diesem Fall gilt

$$\begin{array}{ll} \alpha^1 &= \alpha \\ \alpha^2 &= \alpha^2 \\ \alpha^3 &= \alpha^3 \\ \alpha^4 &= \alpha + 1 \\ \alpha^5 &= \alpha^2 + \alpha \\ \alpha^6 &= \alpha^3 + \alpha^2 \\ \alpha^7 &= \alpha^3 + \alpha + 1 \\ \alpha^8 &= \alpha^2 + 1 \end{array} \quad \begin{array}{ll} \alpha^9 &= \alpha^3 + \alpha \\ \alpha^{10} &= \alpha^2 + \alpha + 1 \\ \alpha^{11} &= \alpha^3 + \alpha^2 + \alpha \\ \alpha^{12} &= \alpha^3 + \alpha^2 + \alpha + 1 \\ \alpha^{13} &= \alpha^3 + \alpha^2 + 1 \\ \alpha^{14} &= \alpha^3 + 1 \\ \alpha^{15} &= 1 \end{array}$$

Damit gilt etwa

$$\frac{\alpha^3 + \alpha + 1}{\alpha^3 + \alpha^2 + 1} = \frac{\alpha^7}{\alpha^{13}} = \alpha^{-6} = \alpha^0 = \alpha^3 + \alpha$$

Bemerkung C.6. In den Körpern \mathbb{F}_p mit einer Primzahl p haben wir das inverse Element mithilfe des erweiterten euklidischen Algorithmus berechnet. Dieses Verfahren kann auf Körper \mathbb{F}_q wie folgt verallgemeinert werden:

Wir nehmen an, dass \mathbb{F}_q durch die Relation

$$\alpha^l = r_0 + r_1\alpha + r_2\alpha^2 + \cdots + r_{l-1}\alpha^{l-1}$$

beschrieben wird und bilden daraus das Minimalpolynom

$$f(X) = X^l - r_{l-1} \cdot X^{l-1} - \cdots - r_1 \cdot X - r_0 \in \mathbb{F}_p[X]$$

der Körpererweiterung $\mathbb{F}_q/\mathbb{F}_p$.

Ist nun $a = a_0 + a_1\alpha + \cdots + a_{l-1}\alpha^{l-1}$ ein Element von $\mathbb{F}_q \setminus \{0\}$, so betrachten wir das zugehörige Polynom

$$a(X) = a_{l-1} \cdot X^{l-1} + \cdots + a_1 \cdot X + a_0$$

Da wir im Polynomring, genauso wie in \mathbb{Z} (Polynom-)Division mit Rest durchführen können, können wir den (erweiterten) euklidischen Algorithmus für die Polynome $f(X)$ und $a(X)$ vollständig auf diese Situation übertragen. Auch hier wird nach mehrfachem Durchführen die Division ohne Rest aufgehen und auch hier ist der letzte nicht-verschwindende Rest der größte gemeinsame Teiler. Da nach Voraussetzung aber $f(x)$ keine Teiler hat und $a(X)$ kein Vielfaches von $f(x)$ ist (denn $a \neq 0$), sind $f(X)$ und $a(X)$ teilerfremd und der größte gemeinsame Teiler ist immer 1.

Durch Rückwärtsrechnen finden wir auch hier Polynome $g(X)$ und $h(X)$ mit

$$1 = g(X) \cdot a(X) + h(X) \cdot f(X)$$

Setzen wir nun α für X ein, so wird daraus

$$1 = g(\alpha) \cdot a(\alpha) + h(\alpha) \cdot f(\alpha) = g(\alpha) \cdot a$$

(denn $f(\alpha) = 0$ in \mathbb{F}_q). Damit ist $g(\alpha)$ das multiplikative Inverse von a , dh. ist

$$g(X) = g_0 + g_1 \cdot X + \cdots + g_t \cdot X^t$$

so ist

$$\frac{1}{a} = g_0 + g_1\alpha + \cdots + g_t\alpha^t$$

Beispiel C.9. Wir betrachten den Körper \mathbb{F}_4 mit 4 Elementen, gegeben durch die Relation $\alpha^2 = \alpha + 1$, also das Minimalpolynom $f(X) = X^2 + X + 1$, und das Element $a = \alpha + 1$. Hierfür gilt

$$a(X) = X + 1$$

und der euklidische Algorithmus liefert

1. $f(X) = X \cdot a(X) + 1.$
2. $a(X) = (X + 1) \cdot 1 + 0 \rightarrow \text{STOPP, Division geht ohne Rest auf.}$

Rückwärtsrechnen liefert

$$1 = f(X) - X \cdot a(X) = 1 \cdot f(X) + X \cdot (X + 1)$$

Einsetzen von α für X ergibt in \mathbb{F}_4 :

$$1 = \alpha \cdot (\alpha + 1)$$

also

$$\frac{1}{\alpha + 1} = \alpha$$

Beispiel C.10. Wir betrachten den Körper \mathbb{F}_8 mit 8 Elementen, gegeben durch die Relation $\alpha^3 = \alpha + 1$, also das Minimalpolynom $f(X) = X^3 + X + 1$, und das Element $a = \alpha^2 + \alpha + 1$. Hierfür gilt

$$a(X) = X^2 + X + 1$$

und der euklidische Algorithmus liefert

1. $f(X) = (X + 1) \cdot a(X) + X.$
2. $a(X) = (X + 1) \cdot X + 1.$
3. $X = X \cdot 1 + 0 \rightarrow \text{STOPP, Division geht ohne Rest auf.}$

Rückwärtsrechnen liefert

$$\begin{aligned} 1 &= (X + 1) \cdot X + a(X) \\ &= (X + 1) \cdot f(X) + (X + 1) \cdot (X + 1) \cdot a(X) + a(X) \\ &= (X + 1) \cdot f(X) + X^2 \cdot a(X) \end{aligned}$$

Einsetzen von α für X ergibt in \mathbb{F}_8 :

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

also

$$\frac{1}{\alpha^2 + \alpha + 1} = \alpha^2$$

Wird der Körper \mathbb{F}_{256} durch die Relation

$$\alpha^8 = \alpha^4 + \alpha^3 + \alpha^2 + 1$$

beschrieben, so gilt hierfür

$$\mathbb{F}_{256} \setminus \{0\} = \{\alpha, \alpha^2, \alpha^3, \dots, \alpha^{255} = 1\}$$

und damit können hier inverse Elemente über die Potenzdarstellung berechnet werden:

Ist $x \in \mathbb{F}_{256} \setminus \{0\}$, so können wir $x = \alpha^r$ für $1 \leq r \leq 255$ schreiben, und dann gilt

$$\frac{1}{x} = \alpha^{255-r}$$

Für die Relation

$$\alpha^8 = \alpha^4 + \alpha^3 + \alpha + 1$$

die in der Kryptologie verwendet wird, gilt das jedoch nicht. Hier hat das Element α nur die Ordnung 51. Deshalb ist die Division in diesem Fall komplizierter, und es ist notwendig, hierfür entweder den euklidischen Algorithmus oder eine Multiplikationstabelle zu benutzen.

Für praktische Anwendungen empfiehlt es sich, einmal eine Tabelle mit allen inversen Elementen anzulegen und dann bei Bedarf darauf zuzugreifen.

D. Quadratische Gleichungen in der Charakteristik 2

Die wichtigsten Körper in der Informatik sind Körper der Charakteristik 2, also Körper der Form \mathbb{F}_q mit $q = 2^e$, denn die Regeln zum Rechnen in diesen Körpern entsprechen der Funktionsweise der Prozessoren. Der Umgang mit diesen Körpern ist allerdings ungewohnt und das Arbeiten in ihnen unterscheidet sich teils sehr deutlich vom Arbeiten in anderen endlichen Körpern. In diesem Abschnitt wird das anhand der Behandlung quadratischer Gleichungen

$$x^2 + ax + b = 0 \quad (\text{D.1})$$

mit $a, b \in k$ untersucht werden.

Der Grund, warum quadratische Gleichungen in der Charakteristik 2 anders zu behandeln sind als in allen anderen Charakteristiken, ist die spezielle Form der binomischen Formel in diesem Fall:

Regel D.1 (Binomische Formel in der Charakteristik 2). *Ist k ein Körper mit $\text{char}(k) = 2$, so gilt für $a, b \in k$:*

$$(a + b)^2 = a^2 + b^2$$

Beweis: Diese Formel folgt, wie die klassische binomische Formel, durch ausmultiplizieren:

$$\begin{aligned} (a + b)^2 &= (a + b) \cdot (a + b) \\ &= a^2 + a \cdot b + a \cdot b + b^2 \\ &= a^2 + b^2 \end{aligned}$$

wobei wir ausgenutzt haben, dass $a \cdot b + a \cdot b = 0$ wegen Charakteristik 2.

Für den Rest dieses Abschnittes betrachten wir einen endlichen Körper k der Form $k = \mathbb{F}_q$, wobei $q = 2^e$ für ein $e \in \mathbb{N}$.

Regel D.2. *Für alle $a \in k$ gilt*

$$a^q = a$$

Beweis: Für $a = 0$ ist die Behauptung klar. Wir brauchen daher nur noch den Fall $a \in k^*$ zu betrachten.

Da (k^*, \cdot) eine (zyklische) Gruppe der Ordnung $q - 1$ bildet, gilt für jedes $a \in k^*$ nach dem Satz von Fermat

$$a^{q-1} = 1$$

Damit gilt aber

$$a^q = a \cdot a^{q-1} = a \cdot 1 = a$$

und die Behauptung ist gezeigt.

Folgerung D.3. Ist $b \in k$ beliebig, so ist

$$w = b^{\frac{q}{2}} = b^{2^{l-1}} \in k$$

die eindeutig bestimmte Wurzel aus b , dh. die Gleichung

$$x^2 = b$$

hat die eindeutige Lösung w in k .

Beweis: Da $b^q = b$ gilt

$$w^2 = \left(b^{\frac{q}{2}}\right)^2 = b^{2 \cdot \frac{q}{2}} = b^q = b$$

und damit ist w eine Wurzel aus b . Wäre v noch eine weitere Lösung, so müsste gelten

$$0 = b + b = v^2 + w^2 = (v + w)^2$$

und damit $v + w = 0$, also $v = -w$, ein Widerspruch. Also ist w die eindeutig bestimmte und einzige Wurzel aus b .

Regel D.4. Für $a \in k$ gilt

$$a \in \mathbb{F}_2 \iff a^2 = a$$

Beweis: Die eine Richtung \implies dieser Aussage ist offensichtlich: Der Unterkörper $\mathbb{F}_2 \subseteq \mathbb{F}_q$ besteht aus den beiden Elementen 0 und 1 und hierfür gilt

$$0^2 = 0, \quad 1^2 = 1$$

Für die andere Richtung betrachten wir das Polynom $f(X) = X^2 + X \in k[X]$. Dann gilt für jedes $a \in k$ mit $a^2 = a$:

$$f(a) = a^2 + a = a + a = 2a = 0$$

(da $\text{char}(k) = 2$), dh. jedes $a \in k$ mit $a^2 = a$ ist eine Nullstelle von $f(X)$. Da aber offensichtlich auch

$$f(X) = X \cdot (X + 1)$$

die Zerlegung von $f(X)$ in Linearfaktoren ist, sind die Nullstellen von $f(X)$ die Nullstellen von X oder $X + 1$, als die Zahlen 0 oder 1. Das beweist die Rückrichtung.

Definition D.1. Für $a \in k$ heißt

$$\text{Tr}(a) = \sum_{j=0}^{l-1} a^{2^j}$$

die **Spur** von a .

Beispiel D.1. Für $k = \mathbb{F}_4$ (gegeben durch $\alpha^2 = \alpha + 1$) gilt:

$$\begin{aligned}\text{Tr}(0) &= 0^1 + 0^2 &= 0 \\ \text{Tr}(1) &= 1^1 + 1^2 &= 0 \\ \text{Tr}(\alpha) &= \alpha^1 + \alpha^2 &= \alpha + \alpha + 1 &= 1 \\ \text{Tr}(\alpha + 1) &= (\alpha + 1)^1 + (\alpha + 1)^2 &= \alpha + 1 + \alpha &= 1\end{aligned}$$

Regel D.5.

- a) Für alle $a, b \in k$ gilt: $\text{Tr}(a + b) = \text{Tr}(a) + \text{Tr}(b)$.
- b) Für alle $a \in k$ und alle $\lambda \in \mathbb{F}_2$ gilt: $\text{Tr}(\lambda \cdot a) = \lambda \cdot \text{Tr}(a)$.
- c) Für alle $a \in k$ gilt: $\text{Tr}(a^2) = (\text{Tr}(a))^2 = \text{Tr}(a)$.
- d) Die Spur Tr definiert eine \mathbb{F}_2 -lineare Abbildung

$$\text{Tr} : \mathbb{F}_q \longrightarrow \mathbb{F}_2$$

Beweis:

- a) Wir wissen aus Regel D.1, dass für alle $r, s \in k$ gilt $(r + s)^2 = r^2 + s^2$. Daraus folgt durch iteriertes Anwenden

$$(r + s)^{2^j} = r^{2^j} + s^{2^j}$$

für jedes $j \geq 1$ und (wiederum durch wiederholtes Anwenden)

$$(r_1 + r_2 + \dots + r_n)^{2^j} = r_1^{2^j} + r_2^{2^j} + \dots + r_n^{2^j}$$

für $r_1, \dots, r_n \in k$. Damit gilt

$$\begin{aligned}\text{Tr}(a + b) &= \sum_{j=0}^{l-1} (a + b)^{2^j} \\ &= \sum_{j=0}^{l-1} (a^{2^j} + b^{2^j}) \\ &= \sum_{j=0}^{l-1} a^{2^j} + \sum_{j=0}^{l-1} b^{2^j} \\ &= \text{Tr}(a) + \text{Tr}(b)\end{aligned}$$

- b) Wir wissen aus Regel D.4, dass für alle $\lambda \in \mathbb{F}_2$ die Formel $\lambda^2 = \lambda$ gilt, woraus durch iteriertes Anwenden folgt, dass $\lambda^{2^j} = \lambda$ für alle $j \geq 1$.

Damit erhalten wir

$$\begin{aligned}\text{Tr}(\lambda \cdot a) &= \sum_{j=0}^{l-1} (\lambda \cdot a)^{2^j} \\ &= \sum_{j=0}^{l-1} \lambda \cdot a^{2^j} \\ &= \lambda \cdot \left(\sum_{j=0}^{l-1} a^{2^j} \right) \\ &= \lambda \cdot \text{Tr}(a)\end{aligned}$$

- c) Wir wenden wieder die Eigenschaft

$$(r_1 + r_2 + \cdots + r_n)^{2^j} = r_1^{2^j} + r_2^{2^j} + \cdots + r_n^{2^j}$$

an, die wir schon in Teil a) benutzt haben. Außerdem benutzen wir die Eigenschaft, dass

$$a^{2^l} = a$$

für alle $a \in k$, was wir in Regel D.2 gesehen haben. Damit gilt

$$\begin{aligned}(\text{Tr}(a))^2 &= \left(\sum_{j=0}^{l-1} a^{2^j} \right)^2 \\ &= \sum_{j=0}^{l-1} (a^{2^j})^2 \\ &= \sum_{j=0}^{l-1} a^{2 \cdot 2^j} \\ &= \sum_{j=0}^{l-1} a^{2^{j+1}} \\ &= a^{2^l} + \sum_{j=1}^{l-1} a^{2^j} \\ &= a + \sum_{j=1}^{l-1} a^{2^j} \\ &= \sum_{j=0}^{l-1} a^{2^j} \\ &= \text{Tr}(a)\end{aligned}$$

und vollkommen analog

$$\begin{aligned}
\text{Tr}(a^2) &= \sum_{j=0}^{l-1} (a^2)^{2^j} \\
&= \sum_{j=0}^{l-1} a^{2 \cdot 2^j} \\
&= \sum_{j=0}^{l-1} a^{2^{j+1}} \\
&= a^{2^l} + \sum_{j=1}^{l-1} a^{2^j} \\
&= a + \sum_{j=1}^{l-1} a^{2^j} \\
&= \sum_{j=0}^{l-1} a^{2^j} \\
&= \text{Tr}(a)
\end{aligned}$$

- d) Da nach Teil c) gilt: $\text{Tr}(a^2) = \text{Tr}(a)$ ist nach Regel D.4 notwendig $\text{Tr}(a) \in \mathbb{F}_2$, und wir erhalten eine Abbildung

$$\text{Tr} : \mathbb{F}_q \longrightarrow \mathbb{F}_2$$

die nach den Teilen a) und b) auch \mathbb{F}_2 -linear ist.

Wir kehren nun zurück zur Gleichung (D.1), also zu

$$x^2 + ax + b = 0$$

Für den Fall, dass $a = 0$, wird diese Gleichung zu

$$x^2 = b$$

und wir haben schon in Folgerung D.3 gesehen, dass $w = b^{2^{l-1}}$ die einzige Lösung dieser Gleichung ist.

Wir können daher für die weitere Behandlung von Gleichung (D.1) annehmen, dass $a \neq 0$.

Hilfssatz D.6. Ist $a \neq 0$, so ist $n \in k$ genau dann eine Lösung von (D.1), wenn $\frac{n}{a}$ eine Lösung der Gleichung

$$x^2 + x + \frac{b}{a^2} = 0 \tag{D.2}$$

ist. Es reicht also, Gleichungen von der Bauart

$$x^2 + x + b = 0 \tag{D.3}$$

zu untersuchen.

Beweis: Wir nehmen zunächst an, dass n eine Lösung von Gleichung (D.1) ist. Dann gilt

$$\left(\frac{n}{a}\right)^2 + \frac{n}{a} + \frac{b}{a^2} = \frac{n^2}{a^2} + \frac{n}{a} + \frac{b}{a^2} = \frac{1}{a^2} \cdot (n^2 + a \cdot n + b) = 0$$

und damit löst $\frac{n}{a}$ die Gleichung (D.2).

Ist umgekehrt n' eine Lösung von Gleichung (D.2), so ist zu zeigen, dass $a \cdot n'$ die Gleichung (D.1) löst. Dazu rechnen wir ähnlich

$$(a \cdot n')^2 + a \cdot (a \cdot n') + b = a^2 \cdot (n')^2 + a^2 \cdot n' + a^2 \cdot \frac{b}{a^2} = a^2 \cdot \left((n')^2 + n' + \frac{b}{a^2}\right) = 0$$

Also löst $a \cdot n'$ tatsächlich Gleichung (D.1).

Hilfssatz D.7. Ist n_1 eine Lösung von Gleichung (D.3), so ist $n_2 = n_1 + 1$ eine weitere Lösung von Gleichung (D.3), und n_1 und n_2 sind die einzigen Lösungen.

Beweis: Ist n eine Lösung von Gleichung (D.3), so gilt

$$(n+1)^2 + (n+1) + b = n^2 + 1^2 + n + 1 + b = n^2 + n + b + 2 = n^2 + n + b = 0$$

also ist $n+1$ eine weitere Lösung von Gleichung (D.3) (die auch von n verschieden ist). Damit sind n und $n+1$ Nullstellen des quadratischen Polynoms $f(X) = X^2 + X + b$. Da ein quadratisches Polynom aber höchstens 2 Nullstellen hat, gibt es keine weiteren Lösungen mehr.

Regel D.8. Hat die Gleichung (D.3) eine Lösungen, so ist $\text{Tr}(b) = 0$

Beweis: Wir betrachten eine Lösung n von Gleichung (D.3). Dann gilt

$$n^2 + n + b = 0$$

also auch

$$\text{Tr}(n^2 + n + b) = 0$$

Damit gilt aber (mit Regel D.5)

$$\begin{aligned} \text{Tr}(n^2 + n + b) &= \text{Tr}(n^2) + \text{Tr}(n) + \text{Tr}(b) \\ &= \text{Tr}(n) + \text{Tr}(n) + \text{Tr}(b) \\ &= 2 \cdot \text{Tr}(n) + \text{Tr}(b) \\ &= \text{Tr}(b) \end{aligned}$$

und die Behauptung ist gezeigt.

Beispiel D.2. Wir betrachten den Körper $k = \mathbb{F}_4$ (gegeben durch $\alpha^2 = \alpha + 1$). Dann hat die quadratische Gleichung

$$x^2 + x + \alpha = 0$$

keine Lösung in k , denn nach Beispiel D.1 ist

$$\text{Tr}(\alpha) = 1 \neq 0$$

Wir betrachten nun einen Körper $k = \mathbb{F}_q$ mit $q = 2^l$, wobei l **ungerade** ist.

Definition D.2. Für ein Element $x \in \mathbb{F}_q$ heißt

$$\text{HTr}(x) = \sum_{j=0}^{\frac{l-1}{2}} x^{2^{2j}}$$

die **Halbspur** von x .

Beispiel D.3. Wir betrachten den Körper \mathbb{F}_8 , gegeben durch $\alpha^3 = \alpha + 1$ und das Element

$$x = \alpha^2 + \alpha + 1 = \alpha^5$$

Hier ist $l = 3$, also $\frac{l-1}{2} = 1$ und daher

$$\begin{aligned} \text{HTr}(x) &= x^{2^0} + x^{2^2} \\ &= x + x^4 \\ &= \alpha^5 + \alpha^{20} \\ &= \alpha^5 + \alpha^6 \\ &= \alpha^2 + \alpha + 1 + \alpha^2 + 1 \\ &= \alpha \end{aligned}$$

Regel D.9. Ist $q = 2^l$ mit l ungerade, und ist $b \in \mathbb{F}_q$ eine Element mit $\text{Tr}(b) = 0$, so ist $n = \text{HTr}(b)$ eine Lösung von Gleichung (D.3), also von

$$x^2 + x + b = 0$$

Beweis: Wir überprüfen diese Aussage durch direktes Nachrechnen und setzen dazu $n = \text{HTr}(b)$ und $m = \frac{l-1}{2}$:

$$\begin{aligned}
n^2 + n + b &= \left(\sum_{j=0}^{\frac{l-1}{2}} b^{2^{2j}} \right)^2 + \sum_{j=0}^{\frac{l-1}{2}} b^{2^{2j}} + b \\
&= \sum_{j=0}^{\frac{l-1}{2}} (b^{2^{2j}})^2 + \sum_{j=0}^{\frac{l-1}{2}} b^{2^{2j}} + b \\
&= \sum_{j=0}^{\frac{l-1}{2}} b^{2 \cdot 2^{2j}} + \sum_{j=0}^{\frac{l-1}{2}} b^{2^{2j}} + b \\
&= \sum_{j=0}^{\frac{l-1}{2}} b^{2^{2j+1}} + \sum_{j=0}^{\frac{l-1}{2}} b^{2^{2j}} + b \\
&= \sum_{j=0}^{\frac{l-1}{2}-1} b^{2^{2j+1}} + b^l + \sum_{j=0}^{\frac{l-1}{2}} b^{2^{2j}} + b \\
&= \sum_{j=0}^{\frac{l-1}{2}-1} b^{2^{2j+1}} + b + \sum_{j=0}^{\frac{l-1}{2}} b^{2^{2j}} + b \\
&= \sum_{j=0}^{\frac{l-1}{2}-1} b^{2^{2j+1}} + \sum_{j=0}^{\frac{l-1}{2}} b^{2^{2j}} \\
&= \sum_{j=0}^{l-1} b^{2^j} \\
&= \text{Tr}(b) \\
&= 0
\end{aligned}$$

Beispiel D.4. Wir betrachten den Körper \mathbb{F}_8 , gegeben durch $\alpha^3 = \alpha + 1$ und die quadratische Gleichung

$$x^2 + x + \alpha^2 = 0$$

Hierfür gilt

$$\text{Tr}(\alpha^2) = \alpha^2 + \alpha^4 + \alpha^8 = \alpha^2 + \alpha^2 + \alpha + \alpha = 0$$

Damit hat diese Gleichung Lösungen, und eine davon ist nach Regel D.9

$$\tilde{x}_1 = \text{HTr}(\alpha^2) = \alpha^2 + \alpha^8 = \alpha^2 + \alpha$$

und die zweite ist dann nach Hilfssatz D.7

$$\tilde{x}_2 = \tilde{x}_1 + 1 = \alpha^2 + \alpha + 1$$

Beispiel D.5. Wir betrachten wieder den Körper \mathbb{F}_8 , gegeben durch $\alpha^3 = \alpha + 1$ und die quadratische Gleichung

$$x^2 + \alpha \cdot x + \alpha^2 + \alpha = 0$$

Gemäß Hilfssatz D.6 betrachten wir zunächst die Gleichung

$$x^2 + x + \frac{\alpha^2 + \alpha}{\alpha^2} = 0$$

also

$$x^2 + x + \alpha^2 = 0$$

Diese hat nach Beispiel D.4 die beiden Lösungen

$$\tilde{x}_1 = \alpha^2 + \alpha, \quad \tilde{x}_2 = \alpha^2 + \alpha + 1$$

Damit hat die Ausgangsgleichung

$$x^2 + \alpha \cdot x + \alpha^2 + \alpha = 0$$

die beiden Lösungen

$$x_1 = \alpha \cdot \tilde{x}_1 = \alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1$$

und

$$x_2 = \alpha \cdot \tilde{x}_2 = \alpha^3 + \alpha^2 + \alpha = \alpha^2 + 1$$

Wir betrachten nun einen Körper \mathbb{F}_q der Charakteristik 2 mit $q = 2^l$ Elementen, wobei l eine **gerade** natürliche Zahl ist. Ferner setzen wir für Elemente $b, c \in \mathbb{F}_q$:

$$\sigma(b, c) = \sum_{i=0}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i}$$

Hilfssatz D.10. Für Elemente $b, c \in \mathbb{F}_q$ gilt

$$a) \quad \sigma(b, c)^2 = \sum_{i=1}^{l-1} \left(\sum_{j=i+1}^l c^{2^j} \right) \cdot b^{2^i}.$$

$$b) \quad \sum_{i=1}^{l-1} \left(\sum_{j=i+1}^l c^{2^j} \right) \cdot b^{2^i} = \sum_{i=1}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} + c \cdot \sum_{i=1}^{l-1} b^{2^i}.$$

Beweis:

a) Wir berechnen $\sigma(b, c)^2$ und erhalten

$$\begin{aligned}
\sigma(b, c)^2 &= \left(\sum_{i=0}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} \right)^2 \\
&= \sum_{i=0}^{l-2} \left(\sum_{j=i+1}^{l-1} (c^{2^j})^2 \right) \cdot (b^{2^i})^2 + \sum_{i=0}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} \\
&= \sum_{i=0}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^{j+1}} \right) \cdot b^{2^{i+1}} + \sum_{i=0}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} \\
&= \sum_{i=0}^{l-2} \left(\sum_{j=i+2}^l c^{2^j} \right) \cdot b^{2^{i+1}} \\
&= \sum_{n=1}^{l-1} \left(\sum_{j=n+1}^l c^{2^j} \right) \cdot b^{2^n} \\
&= \sum_{i=1}^{l-1} \left(\sum_{j=i+1}^l c^{2^j} \right) \cdot b^{2^i}
\end{aligned}$$

Beachten Sie dabei, dass wir in der vorletzten Zeile eine Umindizierung $n = i + 1$ durchgeführt haben. Dadurch beginnt die innere Summe von $j = i+2 = (i+1)+1 = n + 1$ zu laufen. Im letzten Schritt wurde der Laufindex dann wieder i genannt.

b) Durch Abspaltung des letzten Summanden der äußeren Summe erhalten wir

$$\begin{aligned}
\sum_{i=1}^{l-1} \left(\sum_{j=i+1}^l c^{2^j} \right) \cdot b^{2^i} &= c^{2^l} \cdot b^{2^{l-1}} + \sum_{i=1}^{l-2} \left(\sum_{j=i+1}^l c^{2^j} \right) \cdot b^{2^i} \\
&= c^{2^l} \cdot b^{2^{l-1}} + \sum_{i=1}^{l-2} \left(c^{2^l} + \sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} \\
&= c^{2^l} \cdot b^{2^{l-1}} + \sum_{i=1}^{l-2} c^{2^l} \cdot b^{2^i} + \sum_{i=1}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} \\
&= c \cdot b^{2^{l-1}} + \sum_{i=1}^{l-2} c \cdot b^{2^i} + \sum_{i=1}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} \\
&= c \cdot \sum_{i=1}^{l-1} b^{2^i} + \sum_{i=1}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i}
\end{aligned}$$

Regel D.11. Wir betrachten einen endlichen Körper \mathbb{F}_q mit $q = 2^l$, wobei $l \in \mathbb{N}$ gerade ist, und eine quadratische Gleichung der Form (D.3), also

$$x^2 + x + b = 0$$

und fixieren ein Element $c \in \mathbb{F}_q$ mit $\text{Tr}(c) = 1$.

Ist $\text{Tr}(b) = 0$, so ist $\sigma(b, c)$ eine Lösung von (D.3).

Beweis: Wir berechnen zunächst $\sigma(b, c)^2 + \sigma(b, c)$ (unter Ausnutzung von Hilfssatz D.10)) und erhalten

$$\begin{aligned}
\sigma(b, c)^2 + \sigma(b, c) &= \left(\sum_{i=0}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} \right)^2 + \sum_{i=0}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} \\
&= \sum_{i=1}^{l-1} \left(\sum_{j=i+1}^l c^{2^j} \right) \cdot b^{2^i} + \sum_{i=0}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} \\
&= c \cdot \sum_{i=1}^{l-1} b^{2^i} + \sum_{i=1}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} + \sum_{i=0}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} \\
&= c \cdot \sum_{i=1}^{l-1} b^{2^i} + \sum_{i=1}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} + \sum_{i=1}^{l-2} \left(\sum_{j=i+1}^{l-1} c^{2^j} \right) \cdot b^{2^i} + \left(\sum_{j=0+1}^{l-1} c^{2^j} \right) \cdot b^{2^0} \\
&= c \cdot \sum_{i=1}^{l-1} b^{2^i} + \left(\sum_{j=1}^{l-1} c^{2^j} \right) \cdot b^{2^0} \\
&= c \cdot \sum_{i=1}^{l-1} b^{2^i} + c \cdot b^{2^0} + c \cdot b^{2^0} \left(\sum_{j=1}^{l-1} c^{2^j} \right) \cdot b + c^{2^0} \cdot + c^{2^0} \cdot b \\
&= c \cdot \sum_{i=0}^{l-1} b^{2^i} + c \cdot b \left(\sum_{j=0}^{l-1} c^{2^j} \right) \cdot b + c \cdot b \\
&= c \cdot \text{Tr}(b) + \text{Tr}(c) \cdot b \\
&= c \cdot 0 + 1 \cdot b \\
&= b
\end{aligned}$$

Damit gilt

$$\sigma(b, c)^2 + \sigma(b, c) + b = b + b = 0$$

und daher ist $\sigma(b, c)$ eine Lösung von Gleichung (D.3).

Beispiel D.6. Wir betrachten nun speziell den Körper \mathbb{F}_{16} mit $16 = 2^4$ Elementen, gegeben durch die Relation $\alpha^4 = \alpha + 1$. Wir sind also in dieser Situation mit $l = 4$.

Für das Element $c = \alpha^3$ gilt:

$$\text{Tr}(\alpha^3) = \alpha^3 + (\alpha^3)^2 + (\alpha^3)^4 + (\alpha^3)^8 = \alpha^3 + \alpha^6 + \alpha^{12} + \alpha^{24}$$

Diese Potenzen berechnen wir zunächst durch (teilweise durch iteriertes Quadrieren)

$$\alpha^6 = \alpha^2 \cdot \alpha^4 = \alpha^2 \cdot (\alpha + 1) = \alpha^3 + \alpha^2$$

und

$$\alpha^{12} = (\alpha^6)^2 = (\alpha^3 + \alpha^2)^2 = \alpha^6 + \alpha^4 = \alpha^3 + \alpha^2 + \alpha + 1$$

und

$$\alpha^{24} = (\alpha^{12})^2 = \alpha^6 + \alpha^4 + \alpha^2 + 1 = \alpha^3 + \alpha^2 + \alpha + 1 + \alpha^2 + 1 = \alpha^3 + \alpha$$

Damit erhalten wir

$$\text{Tr}(\alpha^3) = \alpha^3 + \alpha^3 + \alpha^2 + \alpha^3 + \alpha^2 + \alpha + 1 + \alpha^3 + \alpha = 1$$

wir haben also ein Element mit Spur 1 gefunden.

Damit wollen wir zunächst die quadratische Gleichung

$$x^2 + x + \alpha^2 = 0$$

untersuchen. Hierfür gilt

$$\text{Tr}(\alpha^2) = \alpha^2 + \alpha + 1 + \alpha^2 + 1 + \alpha + 1 + 1 = 0$$

und damit hat die Gleichung Lösungen. Diese können sofort wie in Regel D.11 angegeben bestimmt werden. Das erforderliche Element mit Spur 1 haben wir mit $c = \alpha^3$ schon gefunden, und wir berechnen als erstes für $i = 0, 1, 2$ die Hilfselemente

$$d_i = \sum_{j=i+1}^3 c_1^{2^j}$$

Es gilt (mit den Zwischenrechnungen wie oben)

$$d_0 = \alpha^6 + \alpha^{12} + \alpha^{24} = \alpha^3 + \alpha^2 + \alpha^3 + \alpha^2 + \alpha + 1 + \alpha^3 + \alpha = \alpha^3 + 1$$

und

$$d_1 = \alpha^{12} + \alpha^{24} = \alpha^3 + \alpha^2 + \alpha + 1 + \alpha^3 + \alpha = \alpha^2 + 1$$

und

$$d_2 = \alpha^{24} = \alpha^3 + \alpha$$

Damit erhalten wir (mit $b = \alpha^2$) eine Lösung durch

$$\begin{aligned} x_1 &= d_0 \cdot b^{2^0} + d_1 \cdot b^{2^1} + d_2 \cdot b^{2^2} \\ &= d_0 \cdot b + d_1 \cdot b^2 + d_2 \cdot b^4 \\ &= (\alpha^3 + 1) \cdot \alpha^2 + (\alpha^2 + 1) \cdot \alpha^4 + (\alpha^3 + \alpha) \cdot \alpha^8 \\ &= \alpha^5 + \alpha^2 + \alpha^6 + \alpha^4 + (\alpha^3 + \alpha) \cdot (\alpha^2 + 1) \\ &= \alpha^2 + \alpha + \alpha^2 + \alpha^3 + \alpha^2 + \alpha + 1 + \alpha^5 + \alpha^3 + \alpha^3 + \alpha \\ &= \alpha^3 + \alpha^2 + 1 + \alpha^2 + \alpha + \alpha \\ &= \alpha^3 + 1 \end{aligned}$$

Zur Kontrolle:

$$(\alpha^3 + 1)^2 + (\alpha^3 + 1) + \alpha^2 = \alpha^6 + 1 + \alpha^3 + 1\alpha^2 = \alpha^3 + \alpha^2 + \alpha^3 + \alpha^2 = 0$$

Damit haben wir eine Lösung $x_1 = \alpha^3 + 1$. Die zweite Lösung ist dann $x_2 = x_1 + 1 = \alpha^3$. Nun wollen wir auch noch die Gleichung

$$x^2 + \alpha \cdot x + \alpha^3 + \alpha^2 = 0 \quad (\text{D.4})$$

untersuchen. Zunächst gehen wir dazu vom Typ $x^2 + ax + b = 0$ über zum Typ $x^2 + x + \frac{b}{a^2} = 0$ also hier zu

$$x^2 + x + \frac{\alpha^3 + \alpha}{\alpha^2} = 0$$

bzw. zu

$$x^2 + x + \alpha + 1 = 0 \quad (\text{D.5})$$

Es ist

$$\text{Tr}(\alpha + 1) = \alpha + 1 + \alpha^2 + 1 + \alpha + 1 + 1 + \alpha^2 = 0$$

und damit hat die reduzierte Gleichung eine Lösung.

Um eine Lösung x_1 von Gleichung (D.5) zu bestimmen, benutzen wir wieder die Hilfs-elemente d_0, d_1 und d_2 von oben und erhalten

$$\begin{aligned} x_1 &= d_0 \cdot b^{2^0} + d_1 \cdot b^{2^1} + d_2 \cdot b^{2^2} \\ &= d_0 \cdot b + d_1 \cdot b^2 + d_2 \cdot b^4 \\ &= (\alpha^3 + 1) \cdot (\alpha + 1) + (\alpha^2 + 1) \cdot (\alpha + 1)^2 + (\alpha^3 + \alpha) \cdot (\alpha + 1)^4 \\ &= (\alpha^3 + 1) \cdot (\alpha + 1) + (\alpha^2 + 1) \cdot (\alpha^2 + 1) + (\alpha^3 + \alpha) \cdot (\alpha^4 + 1) \\ &= (\alpha^3 + 1) \cdot (\alpha + 1) + (\alpha^2 + 1) \cdot (\alpha^2 + 1) + (\alpha^3 + \alpha) \cdot \alpha \\ &= \alpha^4 + \alpha^3 + \alpha + 1 + \alpha^4 + 1 + \alpha^4 + \alpha^2 \\ &= \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1 \\ &= \alpha^3 + \alpha^2 \end{aligned}$$

Damit haben wir eine Lösung $x_1 = \alpha^3 + \alpha^2$. Die zweite Lösung ist dann $x_2 = x_1 + 1 = \alpha^3 + \alpha^2 + 1$.

Damit sind $n_1 = \alpha \cdot x_1$ und $n_2 = \alpha \cdot x_2$ die Lösungen von Gleichung (D.4) also

$$n_1 = \alpha^4 + \alpha^3 = \alpha^3 + \alpha + 1$$

und

$$n_2 = \alpha^4 + \alpha^3 + \alpha = \alpha^3 + 1$$

E. Primzahltest und die Suche nach großen Primzahlen

In der Kryptologie wird sehr häufig mit großen Primzahlen gearbeitet. Eine positive Zahl $p \in \mathbb{Z}$, $p \geq 2$ ist eine **Primzahl**, wenn 1 und p ihre einzigen Teiler sind. Andernfalls heißt die Zahl **zusammengesetzt**. Für kleinere Zahlen ist das leicht durch Probiedivision zu überprüfen, je größer die Zahlen werden, desto ineffizienter ist diese Methode jedoch und andere Ansätze sind notwendig.

E.1. Der Sieb des Eratosthenes

Eine Methode zum Auffinden von Primzahlen in einer bestimmten Größenordnung ist es, alle Primzahlen bis zu einer bestimmten Schranke zu ermitteln. Ein klassischer Ansatz hierfür ist der **Sieb des Eratosthenes**. Suchen wir etwa eine hohe zweistellige Primzahl, so bestimmen wir dafür alle Primzahlen bis zur Zahl 100. Dafür schreiben wir zunächst alle Zahlen ≥ 2 bis 100 in eine Tabelle,

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Am Anfang sind alle Zahlen in der Tabelle nicht markiert und die Liste \mathcal{P} der Primzahlen ist leer. Der Algorithmus (ganz allgemein für alle Zahlen bis S) funktioniert nun wie folgt

1. Untersuche die Tabelle, ob sie noch nicht-markierte Einträge enthält.
 - alle Zahlen in der Tabelle sind markiert \rightarrow STOPP.
 - es gibt noch nicht markierte Zahlen: Gehe zu (2).
2. Wähle die kleinste nicht markierte Zahl a aus, markiere sie und füge sie zu \mathcal{P} hinzu.
3. Überprüfe, ob $a^2 \leq S$.

Ist $a^2 \leq S$, so markiere alle Vielfachen von a die mindestens so groß wie a^2 (und noch nicht markiert) sind. Dann gehe zu (1)

Ist $a^2 > S$, so füge alle noch nicht markierten Zahlen zur Liste \mathcal{P} hinzu und markiere sie, → STOPP.

Ist das STOPP-Kriterium erreicht, so enthält \mathcal{P} alle Primzahlen bis zur vorgegebenen Schranke.

In unserem Beispiel bedeutet das also:

Füge 2 zur Liste der Primzahlen hinzu, $\mathcal{P} = \{2\}$, und markieren sie. Da $2^2 = 4 \leq 100$, markieren wir alle Vielfachen von 2, die größer oder gleich 4 sind: .

Markiere 2 und alle Vielfachen von 2, die größer oder gleich 4 sind:

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Es gibt noch nicht markierte Zahlen. Die kleinste davon ist die 3. Wir fügen 3 zur Liste der Primzahlen hinzu, $\mathcal{P} = \{2, 3\}$, und markieren sie. Da $3^2 = 9 \leq 100$, markieren wir alle (noch nicht markierten) Vielfachen von 3, die größer oder gleich 9 sind:

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Es gibt noch nicht markierte Zahlen. Die kleinste davon ist die 5. Wir fügen 5 also zur Liste der Primzahlen hinzu, $\mathcal{P} = \{2, 3, 5\}$, und markieren sie. Da $5^2 = 25 \leq 100$, markieren wir alle (noch nicht markierten) Vielfachen von 5, die größer oder gleich 25 sind:

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Es gibt noch nicht markierte Zahlen. Die kleinste davon ist die 7. Wir fügen 7 also zur Liste der Primzahlen hinzu, $\mathcal{P} = \{2, 3, 5, 7\}$, und markieren sie. Da $7^2 = 49 \leq 100$, markieren wir alle (noch nicht markierten) Vielfachen von 7, die größer oder gleich 49 sind:

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Es gibt noch nicht markierte Zahlen. Die kleinste davon ist die 11. Wir fügen 11 also zur Liste der Primzahlen hinzu, $\mathcal{P} = \{2, 3, 5, 7, 11\}$, und markieren sie. Da $11^2 = 121 > 100$, fügen wir auch alle anderen noch nicht markierten Zahlen zur Liste der Primzahlen hinzu, markieren Sie und beenden den Algorithmus.

Insgesamt erhalten wir

$$\mathcal{P} = \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97\}$$

und das ist nun eine vollständige Liste aller Primzahlen bis 100.

E.2. Probdivision

Jede positive ganze Zahl n hat eine Primfaktorzerlegung

$$n = p_1^{a_1} \cdot p_2^{a_2} \cdots p_t^{a_t} \quad (\text{E.1})$$

Fordern wir dabei, dass $a_i \geq 1$ und $p_1 < p_2 < \dots < p_t$, so ist diese Zerlegung sogar eindeutig, und n ist genau dann eine Primzahl, wenn $t = 1$ und $a_1 = 1$. Umgekehrt bedeutet das, dass entweder $t \geq 2$ oder $a_2 \geq 2$ gilt, wenn n zusammengesetzt ist. In beiden Fällen ist also $p_1^2 \leq n$. Daraus erhalten wir

Regel E.1. Eine positive Zahl $n \geq 2$ ist genau dann zusammengesetzt, wenn es eine Primzahl $p \leq \sqrt{n}$ gibt, die n teilt.

Um nachzuweisen, dass eine Zahl n eine Primzahl ist, reicht es also, zu zeigen, dass n von keiner Primzahl p mit $p^2 \leq n$ geteilt wird. Daher kann die Primalität von n geprüft werden, indem wir n durch alle Primzahlen p mit $p^2 \leq n$ teilen. Dieses Verfahren nennt man **Probdivision**

Beispiel E.1. Wir wollen überprüfen, ob $n = 9239$ eine Primzahl ist oder nicht.

Die Liste der Primzahlen $\leq \sqrt{9239}$ ist

$$\mathcal{P} = \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89\}$$

(vergleichen Sie dazu die Liste aus dem Abschnitt E.1 und beachten Sie dabei, dass $97^2 = 9409 > n$).

Hierfür gilt

$9239 \div 2$	=	4610	Rest	1
$9239 \div 3$	=	3079	Rest	2
$9239 \div 5$	=	1847	Rest	4
$9239 \div 7$	=	1319	Rest	6
$9239 \div 11$	=	839	Rest	10
$9239 \div 13$	=	710	Rest	9
$9239 \div 17$	=	354	Rest	8
$9239 \div 19$	=	486	Rest	5
$9239 \div 23$	=	401	Rest	16
$9239 \div 29$	=	318	Rest	17
$9239 \div 31$	=	298	Rest	1
$9239 \div 37$	=	249	Rest	26
$9239 \div 41$	=	225	Rest	14
$9239 \div 43$	=	214	Rest	37
$9239 \div 47$	=	196	Rest	27
$9239 \div 53$	=	174	Rest	17
$9239 \div 59$	=	156	Rest	35
$9239 \div 61$	=	151	Rest	28
$9239 \div 67$	=	137	Rest	60
$9239 \div 71$	=	130	Rest	9
$9239 \div 73$	=	126	Rest	41
$9239 \div 79$	=	116	Rest	75
$9239 \div 83$	=	111	Rest	26
$9239 \div 89$	=	103	Rest	72

Bei allen Divisionen bleibt ein Rest, und daher ist n eine Primzahl.

Beispiel E.2. Wir wollen überprüfen, ob $n = 9487$ eine Primzahl ist oder nicht.

Die Liste der Primzahlen $\leq \sqrt{9487}$ ist

$$\mathcal{P} = \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97\}$$

(vergleichen Sie dazu die Liste aus dem Abschnitt E.1 und beachten Sie dabei, dass jetzt $97^2 = 9409 \leq n$).

Hierfür gilt

$$\begin{array}{rcl}
 9847 \div 2 & = & 4743 \quad \text{Rest} \quad 1 \\
 9487 \div 3 & = & 3162 \quad \text{Rest} \quad 1 \\
 9487 \div 5 & = & 1897 \quad \text{Rest} \quad 2 \\
 9487 \div 7 & = & 1355 \quad \text{Rest} \quad 2 \\
 9487 \div 11 & = & 862 \quad \text{Rest} \quad 5 \\
 9487 \div 13 & = & 729 \quad \text{Rest} \quad 10 \\
 9487 \div 17 & = & 558 \quad \text{Rest} \quad 1 \\
 9487 \div 19 & = & 499 \quad \text{Rest} \quad 6 \\
 9487 \div 23 & = & 412 \quad \text{Rest} \quad 11 \\
 9487 \div 29 & = & 327 \quad \text{Rest} \quad 4 \\
 9487 \div 31 & = & 306 \quad \text{Rest} \quad 1 \\
 9487 \div 37 & = & 256 \quad \text{Rest} \quad 15 \\
 9487 \div 41 & = & 231 \quad \text{Rest} \quad 16 \\
 9487 \div 43 & = & 220 \quad \text{Rest} \quad 27 \\
 9487 \div 47 & = & 201 \quad \text{Rest} \quad 40 \\
 9487 \div 53 & = & 179 \quad \text{Rest} \quad 0
 \end{array}$$

Damit geht die Division durch 53 ohne Rest auf und n ist eine zusammengesetzte Zahl.

In der Tat ist

$$n = 53 \cdot 179$$

hier auch schon die Primfaktorzerlegung von n .

E.3. Der Fermat–Test

Das Verfahren der Probedivision setzt voraus, dass eine vollständige Liste aller Primzahlen bis \sqrt{n} vorliegt. Da in der Kryptographie sehr hohe Primzahlen gesucht werden (aktuell mit mehr als 154 Stellen, wenn man wirklich auf der sicheren Seite sein will), ist das in der Praxis nicht gegeben. Beachten Sie dabei auch, dass es „sehr viele“ Primzahlen gibt:

Für eine positive ganze Zahl x bezeichnen wir mit $\pi(x)$ die Anzahl der Primzahlen p mit $p \leq x$. Dann gilt

Satz E.2 (Primzahlsatz).

1. Für $x \geq 17$ ist $\pi(x) > \frac{x}{\ln(x)}$.
2. Für $x > 1$ ist $\pi(x) < 1.25506 \cdot \frac{x}{\ln(x)}$.
3. Es gilt $\lim_{x \rightarrow \infty} \frac{\pi(x)}{\frac{x}{\ln(x)}} = 1$.

Damit gibt es also größtenordnungsmäßig etwa $\frac{a}{\ln(a)}$ viele Primzahlen p mit $p \leq a$, also etwa mehr als 10^{74} viele Primzahlen bis $10^{77} (= \sqrt{10^{154}})$. Diese Listen wäre also unwahrscheinlich lang. Das Verfahren der Probdivision ist daher nicht geeignet, kryptographisch relevante Primzahlen zu finden.

Eine Technik, zu untersuchen, ob eine Zahl zusammengesetzt ist oder nicht, beruht auf der Tatsache, dass für eine Primzahl p der Ring $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ schon ein Körper ist, und dass die Einheitengruppe $E(\mathbb{F}_p)$ daher $p - 1$ Elemente hat. Daher sagt der Satz von Fermat

Regel E.3. *Ist n eine Primzahl und a eine zu n teilerfremde Zahl, so gilt*

$$a^{n-1} = 1 \quad \text{mod } n$$

Ist umgekehrt $n \geq 2$ eine positive ganze Zahl und a eine zu n teilerfremde Zahl mit

$$a^{n-1} \neq 1 \quad \text{mod } n$$

so ist n eine zusammengesetzte Zahl.

Aus dieser Regel lässt sich der Fermat–Test ableiten, der helfen kann, zu entscheiden, ob eine Zahl n zusammengesetzt ist oder nicht. Gegeben sei dafür eine ganze Zahl n und eine Zahl a mit $1 < a < n - 1$.

Der Fermat–Test:

- Überprüfe, ob a und n teilerfremd sind.

Sind a und n nicht teilerfremd, so ist n zusammengesetzt → STOPP.

Sind a und n teilerfremd, gehe zu (2).

- Berechen $a^{n-1} \bmod n$.

Ist $a^{n-1} \neq 1 \bmod n$, so ist n zusammengesetzt → STOPP.

Ist $a^{n-1} = 1 \bmod n$, so ist keine Aussage möglich → STOPP.

Der Fermat–Test kann also allenfalls die Aussage liefern, dass eine Zahl zusammengesetzt ist. Das kann er allerdings sehr häufig feststellen.

Beispiel E.3. Wählen wir $a = 2$, so gilt für ausgewählte (zu a teilerfremde) n :

n	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39
$2^{n-1} \bmod n$	1	1	1	4	1	1	4	1	1	4	1	16	13	1	1	4	9	1	4

Damit hat der Test tatsächlich alle ungeraden zusammengesetzten Zahlen bis 39 erkannt. Es gilt sogar genauer, dass der Fermattest mit $a = 2$ alle zusammengesetzten ungeraden

Zahlen bis $n = 340$ erkennt und nur für Primzahlen $p \leq 340$ gilt, dass $2^{p-1} = 1 \pmod p$ ist. Allerdings gilt für $n = 341$:

$$2^{340} = 1 \pmod{341}$$

obwohl $n = 11 \cdot 31$ eine zusammengesetzte Zahl ist.

Definition E.1. Eine zusammengesetzte Zahl n heißt **Pseudo–Primzahl zur Basis a** , wenn

$$a^{n-1} = 1 \pmod{n}$$

Die Zahl 341 ist also eine Pseudoprimzahl zu Basis 2.

Die Wahrscheinlichkeit, eine Zahl als zusammengesetzt zu erkennen, kann dadurch erhöht werden, dass man den Fermat–Test mit verschiedenen a durchführt.

Beispiel E.4. Wie wir oben gesehen haben, ist $2^{340} = 1 \pmod{341}$. Allerdings ist

$$3^{340} = 56 \pmod{341}$$

dh. der Fermat–Test zur Basis $a = 3$ erkennt, dass 341 zusammengesetzt ist.

Es gibt allerdings zusammengesetzte Zahlen n , die Pseudoprimzahlen zu jeder Basis a sind, die teilerfremd zu n ist. Diese Zahlen werden **Carmichael–Zahlen** genannt. Die kleinste Carmichael Zahl ist $n = 561 = 3 \cdot 11 \cdot 17$.

Satz E.4. Eine ungerade zusammengesetzte Zahl $n \geq 3$ ist genau dann eine Carmichael–Zahl, wenn n keinen quadratischen Teiler hat und wenn für jeden Primfaktor p von n die Zahl $p - 1$ ein Teiler von $n - 1$ ist.

Da es unendlich viele Carmichael–Zahlen gibt, wird der Fermat–Test heute in der Praxis kaum noch verwendet.

E.4. Der Miller–Rabin–Test

Ein häufig verwendeter Primzahltest beruht auf einer Verschärfung des Satzes von Fermat. Dazu betrachten wir eine ungerade Zahl n . Dann ist $n - 1$ gerade, und wir können schreiben

$$n - 1 = 2^t \cdot u$$

mit einer ungeraden Zahl u und einem $t \geq 1$.

Satz E.5 (Miller). Ist n eine Primzahl, $n - 1 = 2^t \cdot u$ wie oben, und a zu n teilerfremd, so gilt entweder

$$a^u \equiv 1 \pmod{n}$$

oder es gibt ein s , $0 \leq s \leq t - 1$, mit

$$a^{2^s \cdot u} \equiv -1 \pmod{n}$$

Beweis: Ist n eine Primzahl, so hat die Einheitengruppe $E(\mathbb{Z}_n)$ die Ordnung $n - 1$, und daher ist die Ordnung von a in $E(\mathbb{Z}_n)$ ein Teiler von $n - 1$. Also ist die Ordnung von a^u ein Teiler von $\frac{n-1}{u} = 2^t$.

Falls also nicht schon gilt

$$a^u \equiv 1 \pmod{n}$$

so gibt es ein $r \in \{1, \dots, t\}$ mit

$$(a^u)^{2^r} \equiv 1 \pmod{n}, \quad (a^u)^{2^{r-1}} \not\equiv 1 \pmod{n}$$

Damit ist $(a^u)^{2^{r-1}}$ in \mathbb{F}_n eine von 1 verschiedene Quadratwurzel aus 1, also -1 , da \mathbb{F}_n ein Körper ist und 1 daher nur die beiden Wurzeln 1 und -1 hat, dh.

$$(a^u)^{2^{r-1}} = a^{2^{r-1} \cdot u} \equiv -1 \pmod{n}$$

und wir können $s = r - 1$ setzen.

Bemerkung E.1. Der Satz von Miller enthält die Aussage des Satzes von Fermat. Ist nämlich $a^u \equiv 1 \pmod{n}$, so gilt sicherlich

$$a^{n-1} = (a^u)^{2^t} = 1^{2^t} = 1 \pmod{n}$$

und ist $a^{2^s \cdot u} \equiv -1 \pmod{n}$ mit $s < t$, so ist

$$a^{n-1} = (a^{2^s \cdot u})^{2^{t-s}} = (-1)^{2^{t-s}} = 1 \pmod{n}$$

Aus dieser Regel lässt sich der Miller–Rabin–Test ableiten. Gegeben ist dafür eine ungerade ganze Zahl $n \geq 3$, die Darstellung $n - 1 = 2^t \cdot u$ und eine Zahl a mit $1 < a < n - 1$.

Der Miller–Rabin–Test:

1. Überprüfe, ob a und n teilerfremd sind.

Sind a und n nicht teilerfremd, so ist n zusammengesetzt → STOPP.

Sind a und n teilerfremd, gehe zu (2).

2. Berechen $a_s = a^{2^{s \cdot u}} \pmod{n}$ ($s = 0, \dots, n-1$).

Ist $a_0 \neq 1 \pmod{n}$ und $a_s \neq -1 \pmod{n}$ für alle $s = 0, \dots, t-1$, so ist n zusammengesetzt → STOPP.

Ist $a_0 = 1 \pmod{n}$ oder $a_s = -1 \pmod{n}$ für ein $s \in \{0, \dots, t-1\}$, so ist keine Aussage möglich → STOPP.

Die Aussage des Miller–Rabin–Tests ist also wieder entweder „ n ist zusammengesetzt“ oder „es ist keine Aussage möglich“. Auch hier gibt es zusammengesetzte Zahlen n und zu n teilerfremde Zahlen, für die (mit den Bezeichnungen aus dem Algorithmus) $a_0 = 1 \pmod{n}$ oder $a_s = -1 \pmod{n}$ für ein $s \in \{0, \dots, t-1\}$ gilt. Eine solche Zahl n wird **starke Pseudoprimzahlen zur Basis a** genannt.

Gilt dagegen $a_0 \neq 1 \pmod{n}$ und $a_s \neq -1 \pmod{n}$ für alle $s = 0, \dots, t-1$ so heißt a **Zeuge gegen die Primalität von n** .

Beispiel E.5. Wir betrachten die Zahl $n = 561 = 3 \cdot 11 \cdot 17$. Der Fermat–Test kann nicht nachweisen, dass n zusammengesetzt ist, da n ein Carmichael–Zahl ist. Wir wollen daher den Miller–Rabin–Test mit Basis $a = 2$ anwenden:

Es ist $n = 2^4 \cdot 35$, also $u = 35$ und $t = 4$. Wir berechnen

$$\begin{aligned} a_0 &= 2^{35} \pmod{561} = 263 \pmod{561} \\ a_1 &= 2^{2 \cdot 35} \pmod{561} = 166 \pmod{561} \\ a_2 &= 2^{2^2 \cdot 35} \pmod{561} = 67 \pmod{561} \\ a_3 &= 2^{32^3 \cdot 5} \pmod{561} = 1 \pmod{561} \end{aligned}$$

Also ist $a_0 \neq 1$ und $a_s \neq -1$ für alle $s \in \{0, 1, 2, 3\}$ und damit ist $a = 2$ ein Zeuge gegen die Primalität von 561. Also ist $n = 561$ keine Primzahl.

Beachten Sie dabei, dass die Tatsache, dass $a_3 = 1 \pmod{561}$ ist, hier kein Problem darstellt.

Dieses Beispiel zeigt bereits, dass der Miller–Rabin–Test stärker ist als der Fermat–Test. Seine praktische Relevanz gewinnt es aus

Satz E.6. Ist $n \geq 3$ eine ungerade zusammengesetzte Zahl, so gibt es unter den Zahlen $\{1, \dots, n-1\}$ höchstens $\frac{n-1}{4}$ viele, die zu n teilerfremd sind und keine Zeugen gegen die Primalität von n sind.

Bemerkung E.2. Der Satz kann auch so gelesen werden:

Ist $n \geq 3$ eine ungerade Zahl und wählen wir mindestens $\frac{n-1}{4} + 1$ Zahlen aus der Menge $\{1, \dots, n-1\}$ aus, so ist n entweder eine Primzahl oder eine der gewählten Zahlen ist nicht teilerfremd zu n oder ein Zeuge gegen die Primalität von n .

Beispiel E.6. Wir wollen untersuchen, wie viele zu $n = 21$ teilerfremde Zahlen a aus $\{1, \dots, 20\}$ es gibt, die keine Zeugen gegen die Primärlität von n sind. Es ist

$$n - 1 = 20 = 2^2 \cdot 5$$

also $u = 5$ und $t = 2$, und eine zu n teilerfremde Zahl a ist kein Zeuge gegen die Primärlität von n , wenn

$$a^5 \equiv 1 \pmod{21} \quad \text{oder} \quad a^5 \equiv 20 \pmod{21} \quad \text{oder} \quad a^{10} \equiv 20 \pmod{21}$$

Es gilt

a	1	2	4	5	8	10	11	13	16	17	19	20
$a^5 \pmod{21}$	1	11	16	17	8	19	2	13	4	5	10	20
$a^{10} \pmod{21}$	1	16	4	16	1	4	4	1	16	4	16	1

Damit haben wir zwei Zahlen gefunden, die keine Zeugen gegen die Primärlität von $n = 21$ sind, nämlich 1 und 20. Offensichtlich ist $2 \leq \frac{n-1}{4} = 5$.

Der Nachweis von Satz E.6 erfordert einiges an Gruppentheorie und soll hier nicht geführt werden. Etwas einfacher zu sehen ist die folgende (schwächere) Aussage

Satz E.7. Ist $n \geq 3$ eine ungerade zusammengesetzte Zahl, so gibt es eine zu n teilerfremde Zahl a mit $1 < a < n$, die Zeuge gegen die Primärlität von n ist.

Beweis: Wir unterscheiden hier zwei Fälle:

1. Fall: n hat mindestens zwei Primteiler

Wir wählen zwei voneinander verschiedene Primteiler p und q (die dann beide ungerade sein müssen) von n und eine Zahl $g \in \mathbb{Z}$, deren Restklasse $g \in \mathbb{F}_p$ die Einheitengruppe von \mathbb{F}_p erzeugt (dh. $g^{p-1} = 1$ in \mathbb{F}_p , aber $g^l \neq 1$ in \mathbb{F}_p für $1 \leq l \leq p-2$), und wir wählen eine zu n teilerfremde Zahl a mit

$$a \equiv g \pmod{p}, \quad a \equiv 1 \pmod{q}$$

Ein solches a existiert immer nach dem chinesischen Restsatz und kann aus der Menge $\{1, 2, \dots, n-1\}$ gewählt werden. Damit gilt

$$a^u \pmod{p} = g^u \pmod{p} \neq 1 \pmod{p}$$

(denn u ist ungerade, und daher ist (die gerade Zahl) $p-1$ kein Teiler von u), also auch

$$a^u \neq 1 \pmod{n}$$

Ferner gilt für jedes $s \in \{0, \dots, t-1\}$:

$$a^{2^s \cdot u} \bmod q = 1^{2^s \cdot u} \bmod q = 1 \bmod q$$

Da q ungerade ist, ist also

$$a^{2^s \cdot u} \neq -1 \bmod q$$

und da q ein Teiler von n ist damit auch

$$a^{2^s \cdot u} \neq -1 \bmod n$$

Also ist in diesem Fall a ein Zeuge gegen die Primalität von n .

2. Fall: n hat nur einen Primteiler

In diesem Fall hat n notwendigerweise die Form $n = p^k$ mit einem $k \geq 2$ (denn n ist zusammengesetzt). Wir wählen eine Zahl $a < p^k$, deren Restklasse in $\mathbb{Z}/p^k\mathbb{Z}$ ein Erzeuger der Einheitengruppe $E(\mathbb{Z}/p^k\mathbb{Z})$ ist, die also dort die Ordnung $\varphi(p^k) = (p-1) \cdot p^{k-1}$ hat (vergleiche Beispiel A.9). Notwendig ist dann a teilerfremd zu $n = p^k$ (da p kein Teiler von a sein kann). Wir behaupten, dass dann

$$a^{n-1} \neq 1 \bmod n$$

Wäre nämlich $a^{n-1} = 1 \bmod n$, so müsste $\varphi(p^k) = p^{k-1} \cdot (p-1)$ ein Teiler von $p^k - 1 = (p^{k-1} + p^{k-2} + \dots + p + 1) \cdot (p-1)$ sein, also auch p^{k-1} ein Teiler von $b = p^{k-1} + p^{k-2} + \dots + p + 1$, was offensichtlich nicht der Fall sein kann, da nicht einmal p ein Teiler von b ist. Damit ist aber a nach Bemerkung E.1 eine Zeuge gegen die Primalität von n .

Unter einer einschränkenden Voraussetzung gilt sogar eine wesentlich schärfere Formulierung:

Satz E.8 (Miller). *Ist $n \geq 3$ ein ungerade zusammengesetzte Zahl und gilt die verallgemeinerte Riemannsche Vermutung, so gibt es eine zu n teilerfremde Zahl $a < 2 \cdot (\ln(n))^2$ die Zeuge gegen die Primalität von n ist.*

Für viele kryptographische Verfahren ist es notwendig, Primzahlen einer bestimmten festen Bitlänge zu erzeugen. Um eine solche zu finden, kann man wie folgt vorgehen:

1. Belege das erste und das letzte der k Bits mit einer 1 (um eine ungerade Zahl zu bekommen und um auch tatsächlich die richtige Größenordnung zu haben).
2. Fülle die Bits dazwischen nach dem Zufallsprinzip mit 0 und 1 auf und nenne die so gefundene Zahl n .

3. Führe eine Probbedivision mit allen Primzahlen bis zu einer vorgegebenen Schranke S durch. Häufig wird $S = 10^6$ gewählt.

Geht eine der Divisionen ohne Rest auf \rightarrow STOPP, n ist keine Primzahl.

4. Wähle zufällig eine Zahl $a \in \{2, 3, \dots, n-1\}$ aus und führe den Miller–Rabin–Test mit n und a durch.

ist a eine Zeuge gegen die Primalität von n , \rightarrow STOPP, n ist keine Primzahl.

5. Wiederhole Schritt (4) bis das Verfahren einen Zeugen liefert oder eine Schranke R erreicht ist.

Die Schranke R hängt davon ab, wie sicher man sich sein möchte, eine Primzahl zu finden. Ist die Zahl n zusammengesetzt, so ist die Wahrscheinlichkeit dafür, dass ein zufällig gewähltes a kein Zeuge gegen die Primalität von n ist, ist höchstens $\frac{1}{4}$ (meistens sogar sehr viel kleiner). Bei 10 unabhängigen Versuchen ist die Wahrscheinlichkeit, keinen Zeugen zu finden also höchstens $\frac{1}{4^{10}} \approx 10^{-6}$. Zusammen mit den Probbedivisionen ist es also sehr unwahrscheinlich, bei $R = 10$ nicht zu entdecken, dass eine Zahl zusammengesetzt ist.

Bemerkung E.3. Aus dem Primzahlsatz folgt, dass die Dichte der Primzahlen unter den Zahlen in der Größenordnung einer vorgegebenen Zahl n etwa $\frac{1}{\ln(n)}$ ist. Damit ist die Wahrscheinlichkeit, dass eine zufällig ausgewählte Zahl n eine Primzahl ist also $p = \frac{1}{\ln(n)}$. Da wir nur ungerade Zahlen betrachten, steigt hierfür die Wahrscheinlichkeit sogar auf $p = \frac{2}{\ln(n)}$.

Damit erhalten wir, dass eine zufällig ausgewählte ungerade 1024–Bitzahl mit einer Wahrscheinlichkeit von etwa

$$p = \frac{2}{\ln(2^{1024})} = \frac{2}{1024 \cdot \ln(2)} \approx \frac{1}{355}$$

eine Primzahl ist, und bei einer zufällig ausgewählten ungeraden 2048–Bitzahl liegt diese Wahrscheinlichkeit immer noch bei etwa $\frac{1}{710}$.

Die Suche nach großen Primzahlen ist also mühsam, es gibt aber genügend viele davon, sodass in einem überschaubaren Zeitrahmen eine gefunden werden kann.

F. Faktorisierung großer Zahlen

Die Sicherheit des RSA–Verfahrens oder der Rabin–Verschlüsselung hängt eng mit der Frage zusammen, wie schnell große Zahlen in ihre Primfaktoren zerlegt werden können. In den letzten Jahrzehnten wurden immer effizientere Faktorisierungsmethoden gefunden, bei der richtigen Wahl der Parameter sind sie aber immer noch weit davon entfernt, die Sicherheit des RSA–Verfahrens oder der Rabin–Verschlüsselung zu gefährden. Allerdings ist nicht auszuschließen, dass schon bald Algorithmen zur schnellen Faktorisierung großer Zahlen gefunden werden. Besondere Gefahr droht hier durch Quantencomputer, für die bereits theoretische Ansätze zu einer schnellen Faktorisierung vorhanden sind. Daher ist es immer wichtig, kryptographische Systeme und Protokolle so aufzusetzen, dass die grundlegenden Verfahren leicht ersetzt werden können.

F.1. Probdivision

Der offensichtliche Ansatz zur Zerlegung einer Zahl n in ihre Primfaktoren ist, jede Primzahl p auszuprobieren und p so oft aus n herauszudividieren, wie das möglich ist. Dazu müssen nur Primzahlen $p \leq \sqrt{n}$ betrachtet werden, denn hat n keinen Primfaktor $p \leq \sqrt{n}$, so ist n schon eine Primzahl. Dafür ist es notwendig, dass alle Primzahlen bis \sqrt{n} als Liste vorliegen,

$$\mathcal{P} = \{p_1, p_2, \dots, p_t\}$$

Ferner setzen wir $i = 1$, $R = n$ und $\mathcal{F} = \emptyset$. Der Algorithmus lässt sich dann wie folgt beschreiben:

Faktorisierung durch Probdivision:

Solange $i \leq t$ oder $R > 1$ führe folgende Schritte durch:

1. Berechne $r_i = R \bmod p_i$.
2. Falls $r_i = 0$ führe folgende Schritte durch:
 - setze $a_i = 1$.
 - Solange $r_i = 0$ setze $R = \frac{R}{p_i}$, $a_i = a_i + 1$ und $r_i = R \bmod p_i$.
 - Setze $\mathcal{F} = \mathcal{F} \cup \{p_i^{a_i}\}$.
3. Setze $i = i + 1$.

Ist $\mathcal{F} = \emptyset$, so ist n eine Primzahl, andernfalls enthält \mathcal{F} alle Primfaktoren von n (bereits zu den passenden Potenzen).

Für kleinere Zahlen ist dieses Vorgehen recht gut geeignet, für große Zahlen mit großen Primteilern ist es allerdings zu aufwendig und unpraktikabel. Es kann allerdings auch bei

großen Zahlen noch leidlich effizient benutzt werden, um kleine Primfaktoren zu finden und abzuspalten.

Beispiel F.1. Wir betrachten die Zahl $n = 882\,078\,485\,121$ und wollen zunächst untersuchen, welche Primteiler $p \leq 100$ diese Zahl hat. Die Liste der Primzahlen bis 100 ist und bereits bekannt,

$$\mathcal{P} = \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97\}$$

(vergleiche etwa Abschnitt E.1).

Probdivision mit den Primzahlen bis 100 liefert:

- $a_3 = 4$ und $R = \frac{n}{3^4} = 10\,889\,857\,841$.
- $a_{17} = 2$ und $R = \frac{R}{17^2} = 37\,681\,169$.
- $a_{47} = 1$ und $R = \frac{R}{47} = 801\,727$.

Weitere Primteiler $p \leq 100$ von n gibt es nicht mehr, und es bleibt ein noch zu untersuchender Rest von $R = 801727$. Da

$$2^{801726} = 418812 \bmod 801727$$

besagt der Fermat–Test, dass dieser Rest noch weiter zerlegt werden kann. Für eine vollständige Analyse von R müssten jetzt aber alle Primzahlen $p \leq 895 = \lfloor \sqrt{R} \rfloor$ betrachtet werden, also eine ziemlich lange Liste.

F.2. Fermat–Faktorisierung

Dieses klassische Verfahren wird erstmals von Fermat 1643 in einem Brief erwähnt. Darin betrachtet er eine ungerade Zahl n und stellt sie als Differenz von zwei Quadraten positiver ganzer Zahlen dar,

$$n = r^2 - s^2 \quad \text{mit } r, s \in \mathbb{N}, r > s + 1$$

Daraus erhält man durch Anwendung der dritten binomischen Formel

$$n = r^2 - s^2 = (r + s) \cdot (r - s)$$

also eine Faktorisierung von n (da $r - s > 1$).

Zunächst scheint diese Methode recht weit hergeholt zu sein, denn warum sollte sich eine zusammengesetzte Zahl n als Differenz von Quadraten schreiben lassen? Tatsächlich ist das aber recht naheliegend. Ist nämlich n eine ungerade zusammengesetzte Zahl,

$$n = a \cdot b$$

mit $a \geq b$, so sind notwendig a und b auch ungerade, also $a+b$ und $a-b$ gerade. Damit ist

$$r = \frac{a+b}{2} \in \mathbb{N} \quad \text{und} \quad s = \frac{a-b}{2} \in \mathbb{N}$$

und

$$a = r + s, \quad b = r - s$$

Also gilt

$$n = a \cdot b = (r+s) \cdot (r-s) = r^2 - s^2$$

und damit lässt sich auf diese Art und Weise jede Zerlegung einer ungeraden Zahl in zwei Faktoren finden. Für die Primfaktorzerlegung können wir dann mit den einzelnen Faktoren weiterarbeiten.

Ist n gerade, so schreiben wir zunächst $n = 2^t \cdot u$ mit einer ungeraden Zahl u und zerlegen dann u wie oben angegeben.

Mit einigen Überlegungen und Abschätzungen erhält man ferner, dass man sich bei der Suche nach Darstellungen

$$n = r^2 - s^2$$

auf Zahlen r mit $\sqrt{n} \leq r \leq \frac{n}{6} + 2$ beschränken kann. Damit lässt sich aus Fermats Beobachtung der folgende Algorithmus ableiten:

Des Fermat–Verfahren:

Setze $r_0 = \lceil \sqrt{n} \rceil$ (die Aufrundung von \sqrt{n}) und $r_1 = \lfloor \frac{n}{6} + 2 \rfloor$ (die Abrundung von $\frac{n}{6} + 2$). Für r von r_0 bis r_1 führe folgende Schritte durch

1. Berechne $z = r^2 - n$.
2. Falls $z = s^2$ für eine ganze Zahl s , → STOPP, $a = r + s$ ist ein Faktor von n .

Beispiel F.2. Wir betrachten die Zahl $n = 231\,377$. Hierfür gilt

$$r_0 = \lceil \sqrt{n} \rceil = 482, \quad r_1 = 38\,565$$

und der Algorithmus läuft wie folgt:

r	$z = r^2 - n$	Primfaktorzerlegung von z	z ein Quadrat?
482	947	947	nein
483	1912	$2^3 \cdot 239$	nein
484	2879	2879	nein
485	3848	$2^3 \cdot 13 \cdot 37$	nein
486	4819	$61 \cdot 79$	nein
487	5792	$2^5 \cdot 181$	nein
488	6767	$67 \cdot 101$	nein
489	7744	$2^6 \cdot 11^2$	ja, $z = 88^2$

Der Algorithmus liefert also den Faktor $a = 489 + 88 = 577$ zurück und es gilt

$$n = 401 \cdot 577$$

Das ist auch schon die Primfaktorzerlegung von n .

Beispiel F.3. Im Beispiel F.1 hatten wir $n = 882\,078\,485\,121$ untersucht und festgestellt dass

$$n = 3^4 \cdot 17^2 \cdot 47 \cdot 801\,727$$

gilt. Den Rest $R = 801\,727$ wollen wir jetzt mit der Fermat–Methode untersuchen. Es ist

$$r_0 = 896, \quad r_1 = 133\,624$$

Damit erhalten wir

r	$z = r^2 - n$	Primfaktorzerlegung von z	z ein Quadrat?
896	1089	$3^2 \cdot 11^2$	ja, $z = 33^2$

Der Algorithmus liefert also bereits im ersten Schritt den Faktor $a = 896 + 33 = 929$ zurück und es gilt

$$n = 861 \cdot 929$$

Das ist auch schon die Primfaktorzerlegung von n .

Bemerkung F.1. Falls n zwei Faktoren von ähnlicher Größe hat, liefert der Algorithmus die Faktoren in wenigen Schritten. Im allgemeinen kann die Laufzeit aber sehr lange sein, im Extremfall ca. $\frac{n}{6}$ Schritte, und daher wird dieses Verfahren in der Praxis nur selten verwendet. Allerdings ist es trotzdem wichtig, beim RSA–Verfahren zwei Primzahlen zu wählen, die nicht zu nahe beieinander liegen.

F.3. Die $p - 1$ -Methode von Pollard

Grundlage für diese Methode ist wieder der Satz von Fermat, der besagt, dass für jede Primzahl p und jede zu p teilerfremde Zahl a gilt

$$a^{p-1} = 1 \mod p$$

Daraus folgt natürlich für jedes $k \geq 1$:

$$a^{k \cdot (p-1)} = (a^{p-1})^k = 1^k = 1 \mod p$$

Ist also m ein Vielfaches von $p - 1$, so ist

$$a^m - 1 = 0 \mod p \quad (\text{F.1})$$

und damit ist p ein Teiler von $a^m - 1$. Diese Idee wurde von Pollard ausgenutzt, um eine Technik zur Faktorisierung zu entwickeln.

Dazu betrachten wir zunächst eine positive ganze Zahl z mit Primfaktorzerlegung

$$z = p_1^{a_1} \cdot p_2^{a_2} \cdots p_t^{a_t}$$

und eine positive ganze Zahl B .

Definition F.1. Die Zahl z heißt **B -glatt**, wenn $p_i \leq B$ für alle i ist, und z heißt **B -potenzglatt**, wenn $p_i^{a_i} \leq B$ für alle i gilt.

Beispiel F.4. Die Zahl $z = 1008 = 2^4 \cdot 3^2 \cdot 7$ ist 7-glatt und 16-potenzglatt, aber nicht 7-potenzglatt.

Ferner definieren wir für eine Zahl B und eine Primzahl $p \leq B$ die Zahl $m_p(B)$ dadurch, dass für sie gilt

$$p^{m_p(B)} \leq B, \quad p^{m_p(B)+1} > B$$

Die Idee von Pollard war nun die folgende:

Ist n eine zusammengesetzte Zahl und gilt wir für einen Primteiler q von n , dass die Zahl $q - 1$ eine B -potenzglatte Zahl ist, so gilt für

$$m = \prod_{p:p \leq B} p^{m_p(B)}$$

(wobei das Produkt über alle Primzahlen $p \leq B$ läuft), dass $q - 1$ ein Teiler von m ist. Damit gilt aber wegen Beziehung F.1, dass q ein Teiler von $a^m - 1$ ist. Da q auch ein Teiler von n ist, gilt also

$$q | \text{ggT}(a^m - 1, n)$$

Daher kann man Informationen über die Teiler von n aus der Untersuchung der größten gemeinsamen Teilers von $a^m - 1$ und n ziehen.

Bemerkung F.2. Da q ja nicht bekannt ist, ist es zunächst nicht klar, wie ein zu q teilerfremdes a gefunden werden kann. Da aber q eine Teiler von n ist, reicht es, eine zu n teilerfremde Zahl zu wählen. Da wir nur ungerade Zahlen n betrachten müssen, können wir sogar immer $a = 2$ wählen (was in der Regel auch getan wird).

Bemerkung F.3. Die Zahlen m werden sehr schnell sehr groß (und damit natürlich auch die Zahlen a^m). Da wir aber nur am größten gemeinsamen Teiler von $a^m - 1$ und n interessiert sind, reicht es die Potenzen a^m modulo n zu berechnen. Das kann mit der Methode des iterierten Quadrierens effizient implementiert werden.

Der von Pollard aus diesen Beobachtungen abgeleitete Algorithmus lässt sich wie folgt beschreiben:

Das $p - 1$ -Verfahren von Pollard:

Gegeben ist eine (zusammengesetzte) Zahl n .

1. Wähle eine Glattheitsschranke B und eine zu n teilerfremde Zahl a .
2. Berechne

$$m = \prod_{p:p \leq B} p^{m_p(B)}$$

3. Berechne $b = a^m - 1 \bmod n$.

4. Berechne $g = \text{ggT}(b, n)$. Falls

$g = 1$, → STOPP mit Fehler, das gewählte B war zu klein.

$g = n$, → STOPP mit Fehler, das gewählte B war zu groß.

$1 < g < n$, → STOPP mit Erfolg, g ist ein Teiler von n .

Falls der Algorithmus mit $g = 1$ abbricht, so war B zu klein und es ist für keinen Primteiler p von n die Zahl $p - 1$ eine B -potenzglatte Zahl. In diesem Fall sollte der Algorithmus mit einem größeren B wiederholt werden.

Falls der Algorithmus mit $g = n$ abbricht (also falls $b = 0$), so war B zu groß gewählt, und es gilt für jeden Primteiler p von n , dass $p - 1$ eine B -potenzglatte Zahl ist. In diesem Fall sollte der Algorithmus mit einem kleineren B wiederholt werden.

Beispiel F.5. Im Beispiel F.2 haben wir die Zahl $n = 231\,377$ mit der Fermat-Methode zerlegt. Hier wollen wir sie mit der Methode von Pollard betrachten.

Als zu n teilerfremde Zahl wählen wir $a = 2$.

Zunächst starten wir mit $B = 15$. Dann ist

$$m_{15} = \prod_{p:p \leq 15} p^{m_p(15)} = 2^3 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13 = 360\,360$$

Mit der Methode des iterierten Quadrierens erhalten wir

$$a_{15} = a^{m_{15}} = 2^{360360} = 99\,820 \mod n$$

Mit dem euklidischen Algorithmus erhalten wir

$$g = \text{ggT}(a_{15} - 1, n) = 1$$

Die Glattheitsschranke $B = 15$ war also zu niedrig.

Wir wiederholen den Prozess mit $B = 20$. Hierfür ist

$$m_{20} = \prod_{p:p \leq 20} p^{m_p(20)} = 232\,792\,560$$

Es ist aber hier einfacher

$$m_{20} = m_{15} \cdot 2 \cdot 17 \cdot 19$$

zu schreiben. Wir haben nämlich schon $a_{15} = a^{m_{15}} \mod n$ berechnet und erhalten daraus

$$a_{20} = a^{m_{20}} = a_{15}^{2 \cdot 17 \cdot 19} = 99\,620^{646} = 123\,479 \mod n$$

Mit dem euklidischen Algorithmus erhalten wir

$$g = \text{ggT}(a_{20} - 1, n) = 577$$

Damit haben wir also einen Teiler $g = 577$ von n gefunden. Dabei ist 577 auch eine Primzahl, und der Kofaktor dazu ist 401, ebenfalls eine Primzahl. Damit ist

$$n = 401 \cdot 577$$

Beispiel F.6. Im Beispiel F.1 haben wir die Zahl $n = 801\,727$ mit der Fermat–Methode zerlegt. Auch diese Zahl wollen wir mit der Methode von Pollard untersuchen.

Als zu n teilerfremde Zahl wählen wir $a = 2$.

Wie oben starten wir zunächst mit $B = 15$ und $m_{15} = 360\,360$. Mit der Methode des iterierten Quadrierens erhalten wir

$$a_{15} = a^{m_{15}} = 2^{360360} = 408\,041 \mod n$$

Mit dem euklidischen Algorithmus erhalten wir

$$g = \text{ggT}(a_{15} - 1, n) = 1$$

Die Glattheitsschranke $B = 15$ war also auch hier zu niedrig.

Wir wiederholen den Prozess mit $B = 20$ und nutzen wie oben aus, dass

$$m_{20} = m_{15} \cdot 2 \cdot 17 \cdot 19$$

Damit gilt wieder

$$a_{20} = a_{15}^{2 \cdot 17 \cdot 19} = 408\,041^{646} = 517\,974 \bmod n$$

Mit dem euklidischen Algorithmus erhalten wir

$$g = \text{ggT}(a_{20} - 1, n) = 1$$

Die Glattheitsschranke $B = 20$ war also immer noch zu niedrig.

Wir wiederholen den Prozess mit $B = 30$ und nutzen dabei aus, dass

$$m_{30} = m_{20} \cdot 3 \cdot 23 \cdot 29$$

Damit gilt

$$a_{30} = a_{20}^{3 \cdot 23 \cdot 29} = 517\,974^{2001} = 767\,355 \bmod n$$

Mit dem euklidischen Algorithmus erhalten wir

$$g = \text{ggT}(a_{30} - 1, n) = 929$$

Damit haben wir also einen Teiler $g = 929$ von n gefunden. Dabei ist 929 auch eine Primzahl, und der Kofaktor dazu ist 863, ebenfalls eine Primzahl. Damit erhalten wir auch hier wieder die Primfaktorzerlegung

$$n = 863 \cdot 929$$

Bemerkung F.4. Die Komplexität von Pollards $p - 1$ -Verfahren zur Faktorisierung einer Zahl n wächst mit $\sqrt[3]{n}$. Damit ist es zwar für große Zahlen wesentlich schneller als der Fermat-Ansatz, wächst aber immer noch exponentiell mit der Länge von n .

F.4. Das Quadratische Sieb

Die effizientesten bekannten Verfahren zur Faktorisierung großer Zahlen sind die sogenannten Siebmethoden. Das Quadratische Sieb soll hier kurz beschrieben werden.

Die Grundidee ist ähnlich zur Idee des Fermatverfahrens. Wieder soll hier die Zahl n mit einer Quadratdifferenz $r^2 - s^2$ in Verbindung gebracht werden. Allerdings wollen wir hier nicht unbedingt eine Darstellung $r^2 - s^2 = n$, wir wollen hier nur $r^2 - s^2 = k \cdot n$ für ein $k \in \mathbb{Z}$ erreichen, also

$$r^2 = s^2 \bmod n$$

allerdings mit

$$r \neq s \pmod{n} \quad \text{und} \quad r \neq -s \pmod{n}$$

Das bedeutet nämlich dann, dass n ein Teiler von $r^2 - s^2$ ist, aber weder von $r - s$ noch von $r + s$. Da aber $(r - s) \cdot (r + s) = r^2 - s^2$, bedeutet das

$$1 < \text{ggT}(r - s, n) < n \quad \text{und} \quad 1 < \text{ggT}(r + s, n) < n$$

und daher haben wir zwei Teiler und damit eine Faktorisierung von n gefunden.

Beispiel F.7. Im Beispiel F.2 haben wir die Zahl $n = 231\,377$ mit der Fermat–Methode zerlegt. Dazu haben wir die für Zahlen r „nahe“ bei $m = \lceil \sqrt{n} \rceil = 482$ den Ausdruck $r^2 - n$ untersucht und versucht, ihn als Quadrat zu schreiben. Diese Zahlen $r^2 - n$ wollen wir auch hier wieder betrachten:

r	$z = r^2 - n$	Primfaktorzerlegung von z
479	-1936	$-2^4 \cdot 11^2$
480	-977	-977
481	-16	-2^4
482	947	947
483	1912	$2^3 \cdot 239$
484	2879	2879
485	3848	$2^3 \cdot 13 \cdot 37$

Keine der hier betrachteten Differenzen $r^2 - n$ ist ein Quadrat (beachten Sie dabei, dass auch -16 und -1936 aufgrund der Vorzeichen keine Quadrate sind), aber

$$(479^2 - n) \cdot (481^2 - n) = (-2^4) \cdot (-2^4 \cdot 11^2) = 176^2$$

also

$$(479 \cdot 481)^2 = 176^2 \pmod{n}$$

mit $r = 481 \cdot 479 = 230\,399$ und $s = 176$ gilt also

$$r + s = 230\,575, \quad r - s = 230\,223$$

und

$$n \mid (r^2 - s^2) \quad \text{aber} \quad n \nmid (r + s), \quad n \nmid (r - s)$$

Der euklidische Algorithmus liefert nun

$$g_1 = \text{ggT}(r + s, n) = 401, \quad g_2 = \text{ggT}(r - s, n) = 577$$

was uns in der Tat in diesem Fall sogar schon die Primfaktorzerlegung

$$n = 401 \cdot 577$$

liefert.

Beispiel F.8. Der hier skizzierte Ansatz ist allgemeiner als die Methode von Fermat. Wenn wir also wie in Beispiel F.1 die Zahl $n = 801\,727$, erhalten wir auch hier schon im ersten Schritt die gewünschte Faktorisierung.

Beispiel F.7 enthält schon die wesentlichen Punkte für die Vorgehensweise zum Finden der Zahlen r und s :

Zum Faktorisieren einer ganzen Zahl n setzen wir $m = \lceil \sqrt{n} \rceil$ und definieren eine Funktion

$$f : \mathbb{Z} \longrightarrow \mathbb{Z}$$

durch $f(t) = (m+t)^2 - n$. Mithilfe der Werte für t und $f(t)$ sollen die Zahlen r und s konstruiert werden. Dazu betrachten wir ausgewählte t_1, \dots, t_τ , für die die Primfaktorzerlegung von $f(t_i)$ nur Primzahlen aus einer fest vorgegebenen Menge

$$\mathcal{F} = \{p_1, p_2, \dots, p_\rho\}$$

ist. Damit können wir (für $i = 1, \dots, \tau$) schreiben

$$f(t_i) = (-1)^{\lambda_{i,0}} \cdot p_1^{\lambda_{i,1}} \cdot p_2^{\lambda_{i,2}} \cdots p_\rho^{\lambda_{i,\rho}} \quad (\text{F.2})$$

(wobei $\lambda_{i,0} = 0$ oder $\lambda_{i,0} = 1$, je nachdem, ob $f(t_i)$ positiv oder negativ ist, und $\lambda_{i,j} \geq 0$). Wir setzen

$$v_i = (\lambda_{i,0}, \lambda_{i,1}, \dots, \lambda_{i,\rho}) \quad (\text{F.3})$$

und suchen i_1, \dots, i_ν , sodass

$$v_{i_1} + \cdots + v_{i_\nu} = (0, \dots, 0) \mod 2$$

Dann ist $f(t_{i_1}) \cdot f(t_{i_2}) \cdots f(t_{i_\nu}) = r^2$ ein Quadrat. Setzen wir noch

$$s = (m + t_{i_1}) \cdots (m + t_{i_\nu})$$

so haben wir zwei Zahlen mit $s^2 = r^2 \mod n$ gefunden. Gilt jetzt auch noch $s \neq r \mod n$ und $s \neq -r \mod n$, so ist das Problem gelöst.

Im Beispiel F.7 waren die relevanten t -Werte $t_1 = -1$ und $t_2 = -3$, die Primzahlen von Interesse waren $\mathcal{F} = \{2, 11\}$ und die zugehörigen Vektoren waren

$$v_1 = (1, 4, 0), \quad v_2 = (1, 4, 2)$$

Hierfür gilt

$$v_1 + v_2 = (2, 8, 2) = (0, 0, 0) \mod 2$$

und $f(t_1) \cdot f(t_2) = 176^2$ ist das Quadrat von $r = 2^4 \cdot 11$, das zusammen mit der Zahl $s = (m + t_1) \cdot (m + t_2)$ die Bedingung erfüllt.

Die Menge \mathcal{F} der relevanten Primzahlen wird dabei wie folgt gewählt: Wir geben uns eine Schranke B vor (üblicherweise in der Großenordnung von $\sqrt{\exp(\sqrt{\ln(n) \cdot \ln(\ln(n))})}$) und betrachtet nur Primzahlen $p \leq B$ (also nur t , für die $f(t)$ eine B -glatte Zahl ist). Zu beachten ist dabei, dass p nur dann ein Teiler von $f(t)$ ist, wenn

$$(m + t)^2 - n = 0 \mod p \quad \text{also} \quad (m + t)^2 = n \mod p$$

Das bedeutet auf jeden Fall, dass n ein Quadrat in \mathbb{F}_p sein muss. Das kann (speziell für kleine p) sehr leicht überprüft werden (vergleiche etwa Folgerung B.3). Alle p , die diese Bedingung nicht erfüllen, müssen also nicht in \mathcal{F} aufgenommen werden. Dafür nimmt man üblicherweise die Zahl -1 in die Menge \mathcal{F} mit auf (da $f(t)$ auch negativ sein darf).

Definition F.2. Zu gegebenem n und B heißt

$$\mathcal{F} = \mathcal{F}(B) = \{-1\} \cup \{p \mid p \text{ ist prim}, p \leq B \text{ und } n \text{ ist eine Quadrat in } \mathbb{F}_p\}$$

die **Faktorbasis zur Schranke B** .

Es bleibt nun noch zu klären, aus welchem Bereich die t gewählt werden, für die $f(t)$ berechnet wird. Dazu setzen wir $C = \exp(\sqrt{\ln(n) \cdot \ln(\ln(n))})$ ($= B^2$ mit B aus der Faktorbasis) und bezeichnen mit

$$S = S(B) = \{t \in \mathbb{Z} \mid |t| < C\} = \{-\lfloor C \rfloor, -\lfloor C \rfloor + 1, \dots, \lfloor C \rfloor - 1, \lfloor C \rfloor\}$$

das **Siebintervall**. Für alle t aus dem Siebintervall wird dann $f(t)$ untersucht und es werden diejenigen t ausgewählt, für die $f(t)$ über der Faktorbasis in Primfaktoren zerfällt (unter Berücksichtigung des Vorzeichens).

Bemerkung F.5. Für ein $t \in S$ ist $p \in \mathcal{F}$ genau dann ein Primfaktor von $f(t)$, wenn $(m + t)^2 = n \mod p$, wenn also $m + t \mod p$ eine Quadratwurzel aus $n \mod p$ ist. Dieses Problem kann (vor allem für kleine p) effizient gelöst werden. Dadurch lassen sich alle $t \in \{0, \dots, p-1\}$ bestimmen, für die $m + t \mod p$ eine Quadratwurzel aus $n \mod p$ ist. Ferner gilt (für jedes $k \in \mathbb{Z}$)

$$\begin{aligned} f(t + k \cdot p) &= (m + t + k \cdot p)^2 - n = (m + t)^2 - n + 2k \cdot (m + t) \cdot p + (k \cdot p)^2 \\ &= (m + t)^2 - n + p \cdot (2k \cdot (m + t) + k^2 \cdot p) \end{aligned}$$

und daraus folgt

$$p \mid f(t) \iff p \mid f(t + k \cdot p) \quad \text{für alle } k \in \mathbb{Z}$$

Damit lassen sich also aus den schon bestimmten $t \in \{0, \dots, p-1\}$ alle t aus dem Siebintervall ableiten, für die $f(t)$ durch p teilbar ist. Das erspart unnötige Probdivisionen.

Der Algorithmus zur Bestimmung einer Faktorisierung der Zahl n lässt sich dann wie folgt formulieren:

Das Quadratische Sieb:

1. Setze $B = \sqrt{\exp(\sqrt{\ln(n) \cdot \ln(\ln(n))})}$.
2. Bestimme die Faktorbasis $\mathcal{F}(B)$.
3. Bestimme das Siebintervall $S(B)$.
4. Für $t \in S$ berechne $f(t)$ und bestimme diejenigen t_1, \dots, t_r , für die $f(t)$ über der Faktorbasis \mathcal{F} vollständig zerfällt.
5. Für alle $i \in \{1, \dots, r\}$ bestimme v_i wie in (F.2) und (F.3).
6. Bestimme $i_1, \dots, i_\rho \in \{1, \dots, r\}$, sodass $v_{i_1} + \dots + v_{i_\rho} = 0 \pmod{2}$.
Falls das nicht möglich ist, → STOPP mit Fehler.
Andernfalls ist $f(t_{i_1}) \cdots f(t_{i_\rho})$ ein Quadrat. Bestimme r mit $r^2 = f(t_{i_1}) \cdots f(t_{i_\rho})$ und setze $s = (m + t_{i_1}) \cdots (m + t_{i_\rho})$.
7. Überprüfe ob $s \neq \pm r \pmod{n}$.
 - falls ja, → STOPP mit Erfolg, $g_1 = \text{ggT}(r - s, n)$ und $g_2 = \text{ggT}(r + s, n)$ sind Teiler von n .
 - falls nein, → STOPP mit Fehler.

Bemerkung F.6. Falls keine $i_1, \dots, i_\rho \in \{1, \dots, r\}$ mit $v_{i_1} + \dots + v_{i_\rho} = 0 \pmod{2}$ gefunden werden können, so kann der Algorithmus mit einem größeren B wiederholt werden.

Wenn der Algorithmus mit $s = r \pmod{n}$ oder $s = -r \pmod{n}$ abbricht, so können andere i_1, \dots, i_ρ mit $v_{i_1} + \dots + v_{i_\rho} = 0 \pmod{2}$ untersucht werden. Falls das nicht möglich ist, kann der Algorithmus mit einem größeren B wiederholt werden.

Bemerkung F.7. Falls bei den Berechnungen für ein t ein v mit $v = 0 \pmod{2}$ ermittelt wird, so kann an dieser Stelle sofort abgebrochen werden, denn dann ist bereits $f(t)$ ein Quadrat, und wir können den Schluss aus der Fermat–Methode anwenden.

Beispiel F.9. Wir untersuchen die Zahl $n = 515\,113$.

Es ist $m = \lceil \sqrt{n} \rceil = 718$ und $B = \sqrt{\exp(\sqrt{\ln(n) \cdot \ln(\ln(n))})} = 18.37$. Wir wählen daher zunächst alle Primzahlen $p \leq 19$ (im Zweifelsfall empfiehlt es sich, hier aufzurunden, speziell wenn B nicht sehr groß ist). Für die Primzahlen $p \leq 19$ ist lediglich für $p = 5$ die Zahl $n \bmod 5$ kein Quadrat, daher erhalten wir als Faktorbasis

$$\mathcal{F} = \{-1, 2, 3, 7, 11, 13, 17, 19\}$$

Für das Siebintervall ist $C = B^2 = 337.412$ relevant, wir betrachten hier jedoch nur

$$S = \{-25, -24, \dots, 24, 25\}$$

um die Darstellung übersichtlicher zu halten. In der Tabelle listen wir auch nur die Werte von t auf, für die $f(t)$ über der Faktorbasis zerfällt:

t	λ_{-1}	λ_2	λ_3	λ_7	λ_{11}	λ_{13}	λ_{17}	λ_{19}	$f(t)$
-17	1	5	1	0	0	1	0	1	-23 712
-13	1	3	0	1	0	0	1	1	-18 088
-3	1	4	5	0	0	0	0	0	-3888
-2	1	0	3	1	0	1	0	0	-2457
-1	1	10	0	0	0	0	0	0	-1024
1	0	3	1	1	1	0	0	0	1848
4	0	0	1	0	2	0	1	0	6171
5	0	6	0	1	0	0	1	0	7616
12	0	0	1	2	2	0	0	0	17 787
15	0	5	2	1	1	0	0	0	22 176
21	0	5	1	0	0	0	1	1	31 008
22	0	0	1	2	0	1	1	0	32 487
25	0	3	5	0	0	0	0	1	36 936

Wir suchen nun eine Linearkombination der v_i , die modulo 2 den Wert 0 ergibt, dh. wir suchen eine Lösung des linearen Gleichungssystems

$$\sum x_i \cdot v_i = 0$$

über \mathbb{F}_2 . Ob solche Lösungen existieren und wie sie aussehen, kann mit Mitteln der linearen Algebra bestimmt werden. In unserem Fall sehen wir, dass

$$v_{-3} + v_{-1} + v_{12} = 0 \quad \bmod 2$$

und wir erhalten, dass

$$f(-3) \cdot f(-1) \cdot f(12) = 70\,815\,596\,544 = 266\,112^2$$

also $r = 255\,112$ und

$$s = (m - 3) \cdot (m - 1) \cdot (m + 12) = 374\,238\,150$$

Allerdings stellt sich hierfür heraus, dass

$$g_1 = \text{ggT}(s - r, n) = 515\,113 = n, \quad g_2 = \text{ggT}(s + r, n) = 1$$

Diese Lösung liefert also keine Faktorisierung von n .

Eine weitere Relation modulo 2 ergibt sich als

$$v_{-3} + v_{-1} + v_1 + v_{15} = 0 \pmod{2}$$

und wir erhalten, dass

$$f(-3) \cdot f(-1) \cdot f(1) \cdot v(15) = 163\,159\,134\,437\,376 = 12\,773\,376^2$$

also $r = 12\,773\,376$ und

$$s = (m - 3) \cdot (m - 1) \cdot (m + 1) \cdot (m + 15) = 270\,183\,026\,685$$

Hierfür gilt nun

$$g_1 = \text{ggT}(s - r, n) = 373 = n, \quad g_2 = \text{ggT}(s + r, n) = 1381$$

Damit haben wir zwei Teiler von n gefunden, und es gilt sogar, dass

$$n = 373 \cdot 1381$$

eine Primfaktorzerlegung von n ist.

Bemerkung F.8. Bei der Untersuchung von $f(t)$ gehen wir am besten wie folgt vor:

Wir setzen $\mathcal{F} = \{-1, p_1, p_2, \dots, p_l\}$ und $R = f(t)$.

Für $i = 1, \dots, l$ führen wir dann folgende Schritte durch:

Wir überprüfen zunächst, ob $m + t \pmod{p_i}$ eine Quadratwurzel von $n \pmod{p_i}$ ist. Ob t diese Eigenschaft erfüllt, haben wir in Bemerkung F.5 schon untersucht. Ist das der Fall, so teilt die Primzahl p_i die Zahl $f(t)$ (und damit auch R), und solange $R \pmod{p_i} = 0$ setzen wir $R = \frac{R}{p_i}$.

Der verbleibende Rest R wird an die nächste Primzahl p_{i+1} aus \mathcal{F} weitergegeben.

Erhalten wir am Schluss $R = \pm 1$, so zerfällt $f(t)$ über der Faktorbasis \mathcal{F} , ist das nicht der Fall, so wird t aus der Liste der relevanten Siebwerte gestrichen.

Bemerkung F.9. Das Quadratische Sieb hat eine Laufzeit in der Größenordnung $e^{\sqrt{\ln(n) \cdot \ln(\ln(n))}}$ und gehört damit zu den schnellsten bekannten Faktorisierungsverfahren.

Bemerkung F.10. Die Grundidee des Quadratischen Siebs kann durch einige Modifikationen weiter beschleunigt werden (partielle Relationen, mehrfache Polynome).

Bemerkung F.11. Der schnellste bekannte klassische Algorithmus zum Faktorisieren großer Zahlen ist das Zahlkörpersieb–Verfahren, das allerdings immer noch nicht polynomiale Laufzeit hat.

Es gibt jedoch einen Quantenalgorithmus von Shor zum Faktorisieren großer Zahlen in polynomialem Laufzeit. Die aktuelle existierenden Quantencomputer sind allerdings noch weit davon entfernt, diesen Algorithmus implementieren zu können.

G. Das diskrete Logarithmus–Problem

In den reellen Zahlen ist der Logarithmus einer positiven Zahl a zu einer Basis $b > 0$ die Zahl $e \in \mathbb{R}$ für die gilt

$$a = b^e$$

Diese Zahl $e := \log_b(a)$ kann in den reellen Zahlen sehr gut effizient algorithmisch berechnet werden.

Anders sieht es mit dem diskreten Logarithmus in endlichen zyklischen Gruppen aus. Dazu betrachten wir eine endliche zyklische Gruppe (G, \circ) mit einem Erzeuger g . Für ein beliebiges Element $a \in G$ und ein $t \in \mathbb{N}$ schreiben wir wie üblich

$$a^t = \underbrace{a \circ a \circ \cdots \circ a}_{t\text{-mal}}$$

Da g ein Erzeuger von G ist, gibt es zu jedem $a \in G$ eine Zahl $n \in \mathbb{N}$, sodass

$$a = g^n$$

Diese Zahl n ist eindeutig, wenn wir zusätzlich noch fordern, dass $0 \leq n < \text{ord}(G)$.

Bezeichnung:

Das Element n mit $0 \leq n < \text{ord}(G)$ und $a = g^n$ heißt der **diskrete Logarithmus** von a (bezüglich g) und wird auch mit $\log_g(a)$ bezeichnet.

Diskretes Logarithmus–Problem:

Zu einem $a \in G$ bestimme die Zahl $n \in \mathbb{N}$ mit $1 \leq n \leq \text{ord}(G)$, sodass

$$a = g^n$$

Beispiel G.1. Die Gruppe $G = (\mathbb{Z}_N, +) = (\mathbb{Z}/N\mathbb{Z}, +)$ ist für jedes $N \in \mathbb{N}$ zyklisch und 1 ist ein erzeugendes Element dieser Gruppe. In diesem Fall schreiben wir $n \cdot a$ anstelle von a^n .

Ein beliebiges Element $a \in \mathbb{Z}_N$, beschrieben durch seinen Standardrepräsentanten aus der Menge $\{0, \dots, N-1\}$ schreibt sich dann auch also

$$a = a \cdot 1$$

und damit ist a auch der diskrete Logarithmus von a (bezüglich 1). Das diskrete Logarithmusproblem in \mathbb{Z}_N ist also einfach.

Beispiel G.2. Die multiplikative Gruppe $E(\mathbb{F}_{17}) = \mathbb{F}_{17} \setminus \{0\}$ des Körpers mit 17 Elementen ist zyklisch und $g = 3$ ist ein Erzeuger davon. Hierfür gilt

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\log_3(a)$	0	2	1	12	5	15	11	10	2	3	7	13	4	9	6	8

Eine Struktur in den diskreten Logarithmen ist hier nicht zu erkennen, sie müssen durch ausprobieren ermittelt werden.

Ganz allgemein ist die multiplikative Gruppe $E(\mathbb{F}_p) = \mathbb{F}_p \setminus \{0\}$ zyklisch (allerdings ohne einen offensichtlichen Erzeuger). Das diskrete Logarithmus–Problem in $E(\mathbb{F}_p)$ (zu einem gegebenen Erzeuger g) gilt als sehr schwierig. Bis heute ist keine klassischer Algorithmus bekannt, der das Problem in polynomialer Zeit lösen würde. Es gibt allerdings (technisch noch nicht realisierbare) Quantenalgorithmen für dieses Problem mit polynomialer Laufzeit.

Ein erster Ansatz zur Bestimmung des diskreten Logarithmus von a in $E(\mathbb{F}_p)$ zur Basis g wäre ein vollsändige Berechnung aller Elemente g^n , solange bis wir $g^n = a$ erhalten. Für kleine Gruppenordnungen ist das möglich.

Beispiel G.3. Für die Primzahl $p = 2027$ ist $g = 7$ ein Erzeuger von $E(\mathbb{F}_{2027})$. Betrachten wir die Zahl $a = 1133$, so erhalten wir durch Ausprobieren und vollständige Berechnung, dass

$$7^{1417} = 1133$$

also $\log_7(1133) = 1417$.

Bemerkung G.1. In der Kryptographie werden aktuell Primzahl p verwendet, deren binäre Länge mindestens 1024 ist. In diesem Fall ist die vollständige Auflistung keine Alternative zur Berechnung des diskreten Logarithmus.

G.1. Endliche zyklische Gruppen

Viele der Verfahren zur Bestimmung des diskreten Logarithmus beziehen sich auf allgemeine zyklische Gruppe. Daher stellen wir in diesem Abschnitt kurz einige grundlegende Eigenschaften (endlicher, zyklischer) Gruppen aus der linearen Algebra zusammen.

Definition G.1. Eine **Gruppe** (G, \circ) ist eine nicht–leere Menge G zusammen mit einer Verknüpfung

$$\circ : G \times G \longrightarrow G, \quad (a, b) \longmapsto a \circ b$$

sodass die folgenden Eigenschaften erfüllt sind:

Für alle $a, b, c \in G$ gilt $a \circ (b \circ c) = (a \circ b) \circ c$ (**Assoziativität**).

Es gibt ein Element $e \in G$ mit $a \circ a = a \circ e = a$ für alle $a \in G$ (**neutrales Element**).

Zu jedem $a \in G$ gibt es ein Element $b \in G$ mit $a \circ b = e = b \circ a$ (**inverses Element**).

Gilt zusätzlich

$$a \circ b = b \circ a \quad \text{für alle } a, b \in G$$

so heißt die Gruppe **kommutativ**.

Beispiel G.4. Die Mengen $(\mathbb{R}, +)$, $(\mathbb{Q}, +)$ und $(\mathbb{Z}, +)$ (mit der üblichen Addition) sind Gruppen, aber (\mathbb{R}, \cdot) , (\mathbb{Q}, \cdot) sind keine Gruppen, da es bezüglich \cdot kein inverses Element zu 0 gibt.

Die Menge $(\mathbb{Z}/n\mathbb{Z}, +)$ ist eine Gruppe.

$(\mathbb{R} \setminus \{0\}, \cdot)$, $(\mathbb{Q} \setminus \{0\}, \cdot)$ sind Gruppen, da es zu jeder von Null verschiedenen reellen oder rationalen Zahl ein multiplikatives Inverses in \mathbb{R} oder \mathbb{Q} gibt, aber $(\mathbb{Z} \setminus \{0\}, \cdot)$ ist keine Gruppe, denn 2 hat kein multiplikatives Inverses in \mathbb{Z} .

Für ein Element $g \in G$ schreiben wir g^{-1} für das inverse Element zu g . Ferner setzen wir für $n \geq 1$:

$$g^n = \underbrace{g \circ g \circ \cdots \circ g}_{n-\text{mal}}$$

und

$$g^{-n} = (g^{-1})^n \quad (= (g^n)^{-1})$$

Definition G.2. Eine Gruppe G heißt **zyklisch**, wenn es eine Element $g \in G$ gibt, sodass für jedes $a \in G$ ein $k \in \mathbb{Z}$ existiert mit

$$a = g^k$$

In diesem Fall heißt g **Erzeuger** von G .

Beispiel G.5. Die Gruppen $(\mathbb{Z}/n\mathbb{Z}, +)$ sind zyklisch. Für jede Zahl $z \in \mathbb{Z}$, die teilerfremd zu n ist, ist die Restklasse $z \in \mathbb{Z}/n\mathbb{Z}$ ein Erzeuger von $\mathbb{Z}/n\mathbb{Z}$.

Bemerkung G.2. Jede zyklische Gruppe ist kommutativ.

Definition G.3. Eine Gruppe (G, \circ) heißt **endlich**, wenn $|G| < \infty$, also wenn G nur endlich viele Elemente hat.

In diesem Fall heißt $\text{ord}(G) = |G|$ die **Ordnung** der Gruppe G .

Beispiel G.6. Die Gruppen $(\mathbb{Z}/n\mathbb{Z}, +)$ sind endlich mit $\text{ord}(G) = n$.

Satz G.1 (Fermat). Ist G eine endliche Gruppe der Ordnung n , so gilt für jedes Element $g \in G$:

$$g^n = e$$

(wobei e das neutrale Element von G ist.

Definition G.4. Ist G eine endliche Gruppe und $g \in G$, so heißt

$$\text{ord}(g) = \min\{n \in \mathbb{N} \mid g^n = e\}$$

die **Ordnung** von g .

Bemerkung G.3. Ist $g \in G$ ein Element der Ordnung n , so ist $g^i \neq g^j$ für alle Zahlen $1 \leq i < j \leq n$.

Wäre nämlich $g^i = g^j$, so würde gelten

$$g^{i-j} = g^i \circ g^{-j} = g^i \circ (g^j)^{-1} = g^i \circ (g^i)^{-1} = e$$

im Widerspruch zur Definition der Ordnung, denn $1 \leq i - j < n$.

Definition G.5. Eine **Untergruppe** einer Gruppe G ist eine nichtleere Teilmenge U von G mit der Eigenschaft, dass $a \circ b^{-1} \in U$ für alle $a, b \in U$.

Ist G endlich, so heißt in diesem Fall $\text{ord}(U) = |U|$ die **Ordnung** der Untergruppe U .

Bemerkung G.4. Ist G eine endliche Gruppe und $g \in G$ ein Element der Ordnung n , so ist die Teilmenge

$$U_g = \langle g \rangle = \{g, g^2, \dots, g^n = e\} \subseteq G$$

eine Untergruppe der Ordnung n von G .

Die Untergruppe U_g heißt die von g erzeugte (zyklische) Untergruppe von G . Hierfür gilt

$$\text{ord}(U_g) = \text{ord}(g)$$

Bemerkung G.5. Eine endliche Gruppe der Ordnung n ist genau dann zyklisch, wenn sie ein Element der Ordnung n enthält.

Satz G.2 (Lagrange). Ist G eine endliche Gruppe und $U \subseteq G$ eine Untergruppe, so gilt

$$\text{ord}(U) \mid \text{ord}(G)$$

Speziell gilt für jedes Element $g \in G$.

$$\text{ord}(g) \mid \text{ord}(G)$$

Ist p ein Primteiler der Ordnung $\text{ord}(G)$ von G , so gibt es immer ein Element $g \in G$ mit $\text{ord}(g) = p$. Insbesondere gibt es also zu jedem Primteiler p von G immer eine Untergruppe $U \subseteq G$ mit p Elementen.

Folgerung G.3. Ist G eine endliche Gruppe der Ordnung p , wobei p eine Primzahl ist, so ist G schon zyklisch.

Folgerung G.4. Ist G eine endliche kommutative Gruppe der Ordnung $n = 2 \cdot p$ mit einer Primzahl $p \geq 3$, so ist G schon zyklisch, und ein Element $g \in G$ ist genau dann ein Erzeuger von G , wenn $g^2 \neq e$ und $g^p \neq e$.

Beweis: Nach dem Satz G.2 von Lagrange gilt für ein Element $g \in G$:

$$\text{ord}(g) = 1 \quad \text{oder} \quad \text{ord}(g) = 2 \quad \text{oder} \quad \text{ord}(g) = p \quad \text{oder} \quad \text{ord}(g) = 2 \cdot p$$

und nach Bemerkung G.5 ist G genau dann zyklisch, wenn es ein Element der Ordnung $2 \cdot p$ gibt.

Außerdem gibt es nach dem Satz von Lagrange ein Element g_1 der Ordnung 2 und ein Element g_2 der Ordnung p . Dann gilt für das Element $g = g_1 \cdot g_2$:

$$g^2 = (g_1 \cdot g_2)^2 = g_1^2 \circ g_2^2 = g_2^2 \neq e$$

(den $\text{ord}(g_2) \geq 3$) und

$$g^p = g_1^p \cdot g_2^p = g_1^p = g_1^{p-1} \circ g_1 = g_1$$

(denn $p-1$ ist gerade, also $g_1^{p-1} = e$.) Offensichtlich ist damit auch $g \neq e$, also $\text{ord}(g) \neq 1$, und daher muss gelten, dass $\text{ord}(g) = 2 \cdot p$. Folglich ist G zyklisch.

Umgekehrt muss offensichtlich auch jedes Element mit $g^2 \neq e$ und $g^p \neq e$ die Ordnung $2 \cdot p$ haben.

Folgerung G.5. Ist G eine zyklische Gruppe der Ordnung n mit Erzeuger g , und ist m ein Teiler von n , so gibt es genau eine Untergruppe U_m von G der Ordnung m und diese Untergruppe U wird erzeugt von dem Element $g^{\frac{n}{m}}$.

Beweis: Zunächst ist klar, dass $g_m := g^{\frac{n}{m}}$ ein Element der Ordnung m ist, denn g ist ein Element der Ordnung n . Also erzeugt g_m eine Untergruppe $U_m = \langle g_m \rangle$ der Ordnung m .

Ist umgekehrt U eine Untergruppe der Ordnung m von G , so setze

$$i = \min\{j \in \mathbb{N} \mid g^j \in U\}$$

Wir zeigen zunächst, dass U zyklisch ist und von $h := g^i$ erzeugt wird. Dazu betrachten wir eine beliebiges Element $a \in U$. Notwendig ist $a = g^j$ für ein $j \in \mathbb{N}$. Falls $j = k \cdot i$ für ein $k \in \mathbb{N}$, so ist

$$a = g^{k \cdot i} = (g^i)^k = h^k$$

in der von h erzeugten Untergruppe. Falls dagegen j kein Vielfaches von i ist, so gilt

$$u := \text{ggT}(i, j) < i$$

Nach dem Lemma von Bezout (vergleiche Satz A.1) gibt es dann $r, s \in \mathbb{Z}$ mit

$$u = r \cdot i + s \cdot j$$

also

$$g^u = g^{r \cdot i + s \cdot j} = (g^i)^r \circ (g^j)^s \in U$$

Da aber $u < i$ ist das ein Widerspruch zur Definition von i , und damit erzeugt $h = g^i$ die Untergruppe U . Notwendig ist damit $\text{ord}(g^i) = m$, also

$$(g^i)^m = g^{i \cdot m} = e$$

woraus folgt, dass $i \cdot m = k \cdot n$, also $i = k \cdot \frac{n}{m}$. Dabei muss notwendig k teilerfremd zu m sein, den falls $k = r \cdot k'$ und $m = r \cdot m'$, so würde schon $i = k' \cdot \frac{n}{m'}$, und damit hätte g^i die Ordnung $m' < m$. Also gibt es nach dem Lemma von Bezout ganze Zahlen r, s mit

$$1 = r \cdot k + s \cdot m$$

Damit gilt

$$g^n m = g^{(r \cdot k + s \cdot m) \cdot \frac{n}{m}} = g^{r \cdot k \cdot \frac{n}{m}} \circ g^{s \cdot n} = (g^i)^r \circ (g^n)^s = (g^i)^r$$

also liegt auch $g^{\frac{n}{m}}$ in der von g^i erzeugten Untergruppe, und daher muss nach Definition von i schon $k = 1$ gelten und $U = U_m$.

G.2. Babystep–Giantstep–Algorithmus

Eine erste Methode zur schnelleren Berechnung des diskreten Logarithmus ist der Babystep–Giantstep–Algorithmus von Daniel Shanks. Dazu betrachten wir ganz allgemein eine zyklische Gruppe G der Ordnung N , einen Erzeuger g von G und ein beliebiges Element a von G . Ferner setzen wir

$$m = \lceil \sqrt{N} \rceil$$

(die Aufrundung von \sqrt{N}). Ist dann $a = g^n$, so schreiben wir

$$n = q \cdot m + r$$

mit $0 \leq r \leq m - 1$ und $0 \leq q \leq m - 1$. Damit erhalten wir

$$a = g^{q \cdot m + r} = g^{q \cdot m} \cdot g^r = (g^m)^q \cdot g^r$$

also

$$g^r = a \cdot (g^{-m})^q$$

Diese Darstellung nutzen wir jetzt aus und führen folgenden Algorithmus durch:

Babystep–Giantstep–Algorithmus

1. *babystep*: Für r von 0 bis $m - 1$ berechne g^r und lege eine Tabelle \mathcal{B} mit den Werten (r, g^r) an.
2. *giantstep*: Für q von 0 bis $m - 1$ berechne $a \cdot (g^{-m})^q$.
 - berechne $x_q = a \cdot (g^{-m})^q$.
 - falls für ein $0 \leq r \leq m - 1$ der Wert $(r, x_q) \in \mathcal{B}$, → STOPP, gebe $n = q \cdot m + r$ zurück

In der Tat gilt, falls $(r, x) \in \mathcal{B}$, dass

$$x = a \cdot (g^{-m})^q = g^r$$

also

$$a = (g^m)^q \cdot g^r = g^{m \cdot q + r} = g^n$$

und das Problem ist gelöst.

Beispiel G.7. Wir greifen Beispiel G.3 wieder auf und wollen $\log_7(1133)$ in $E(\mathbb{F}_{2027})$ mit dem Babystep–Giantstep–Algorithmus berechnen. Wir setzen $a = 1133$. Es ist

$$m = \lceil \sqrt{2026} \rceil = 46$$

1. Als Tabelle \mathcal{B} erhalten wir

(0, 1)	(1, 7)	(2, 49)	(3, 343)	(4, 374)	(5, 591)	(6, 83)
(7, 581)	(8, 13)	(9, 91)	(10, 637)	(11, 405)	(12, 808)	(13, 1602)
(14, 1079)	(15, 1472)	(16, 169)	(17, 1183)	(18, 173)	(19, 1211)	(20, 369)
(21, 556)	(22, 1865)	(23, 893)	(24, 170)	(25, 1190)	(26, 222)	(27, 1554)
(28, 743)	(29, 1147)	(30, 1948)	(31, 1474)	(32, 183)	(33, 1281)	(34, 859)
(35, 1959)	(36, 1551)	(37, 722)	(38, 1000)	(39, 919)	(40, 352)	(41, 437)
(42, 1032)	(43, 1143)	(44, 1920)	(45, 1278)			

2. Wir erhalten

$$g^{-m} = g^{-46} = g^{2026-46} = g^{1980} = 1565$$

in \mathbb{F}_{2027} (wobei wir $g^{2026} = 1$ ausgenutzt haben). Daher berechnen sich die Werte x_q als

$$x_q = a \cdot (g^{-m})^q = 1133 \cdot 1565^q$$

und wir erhalten

$$\begin{array}{llllll}
 x_0 = 1133 & x_1 = 1547 & x_2 = 817 & x_3 = 1595 & x_4 = 938 & x_5 = 422 \\
 x_6 = 1655 & x_7 = 1596 & x_8 = 476 & x_9 = 1031 & x_{10} = 23 & x_{11} = 1536 \\
 x_{12} = 1845 & x_{13} = 977 & x_{14} = 647 & x_{15} = 1082 & x_{16} = 785 & x_{17} = 163 \\
 x_{18} = 1720 & x_{19} = 1971 & x_{20} = 1548 & x_{21} = 355 & x_{22} = 177 & x_{23} = 1333 \\
 x_{24} = 362 & x_{25} = 997 & x_{26} = 1452 & x_{27} = 1100 & x_{28} = 577 & x_{29} = 990 \\
 x_{30} = 722 & \rightarrow & & & & \text{STOPP}
 \end{array}$$

Wir haben in der Tabelle \mathcal{B} den Wert $(37, 722) = (37, x_{30})$, also erhalten wir

$$n = 30 \cdot 46 + 37 = 1417$$

und haben damit $\log_7(1133)$ gefunden.

Bemerkung G.6. Die Laufzeit des Babystep–Giantstep Algorithmus ist $\mathcal{O}(\sqrt{N})$, also proportional zu \sqrt{N} (im Gegensatz zur vollständigen Auflistung, deren Laufzeit proportional zu N ist). Der Algorithmus erfordert allerdings die Berechnung einer Nachschlagetabelle mit \sqrt{N} Einträgen, ist also sehr speicherintensiv.

G.3. Der Pollard– ρ –Algorithmus

Eine von der Komplexität her mit dem Babystep–Giantstep–Verfahren vergleichbare Methode ist Pollards ρ –Algorithmus. Auch hier wird für eine beliebige zyklische Gruppe G der Ordnung N mit Erzeuger g zu einem $a \in G$ ein n ermittelt mit $g^n = a$. Für diesen Algorithmus wird zunächst die Gruppe G in drei disjunkte, etwa gleich große Teilmengen G_1 , G_2 und G_3 zerlegt. Nun definieren wir eine Funktion

$$f : G \longrightarrow G$$

durch

$$f(b) = \begin{cases} g \cdot b & \text{falls } b \in G_1 \\ b^2 & \text{falls } b \in G_2 \\ a \cdot b & \text{falls } b \in G_3 \end{cases}$$

Ferner definieren wir für einen beliebigen Startwert $b_0 \in G$ induktiv

$$b_{i+1} = f(b_i) \quad \text{für } i \geq 0$$

und wir definieren zwei Hilfsfunktionen

$$g, h : G \times \mathbb{Z} \longrightarrow G$$

durch

$$g(b, n) = \begin{cases} n + 1 \bmod p - 1 & \text{falls } b \in G_1 \\ 2n \bmod p - 1 & \text{falls } b \in G_2 \\ n \bmod p - 1 & \text{falls } b \in G_3 \end{cases}$$

und

$$h(b, n) = \begin{cases} n \bmod N & \text{falls } b \in G_1 \\ 2n \bmod N & \text{falls } b \in G_2 \\ n + 1 \bmod N & \text{falls } b \in G_3 \end{cases}$$

Damit lässt sich der Algorithmus wie folgt formulieren:

Pollards- ρ -Algorithmus

- Wähle zufällig ein $x_0 \in \{1, \dots, p - 1\}$ und setze

$$y_0 = 0, \quad b_0 = g^{x_0}$$

- Für $i \geq 0$ führe folgende Schritte durch

- Berechne

$$b_i = f(b_{i-1}), \quad b_{2i} = f(b_{2i-2})$$

- Berechne

$$x_i = g(b_{i-1}, x_{i-1}), \quad y_i = h(b_{i-1}, y_{i-1})$$

- Berechne

$$x_{2i} = g(f(b_{2i-2}), g(b_{2i-2}, x_{2i-2})), \quad y_{2i} = h(f(b_{2i-2}), h(b_{2i-2}, y_{2i-2}))$$

- falls $b_i = b_{2i}$, → STOPP

Der entscheidende Punkt bei dem Algorithmus ist, dass

$$b_i = g^{x_i} \cdot a^{y_i}, \quad b_{2i} = g^{x_{2i}} \cdot a^{y_{2i}}$$

(wie man induktiv leicht aus der Definition von f , g und h bekommt). Wenn also $b_i = b_{2i}$, so erhalten wir daraus

$$g^{x_i} \cdot a^{y_i} = g^{x_{2i}} \cdot a^{y_{2i}}$$

also durch Umstellen

$$a^{y_i - y_{2i}} = g^{x_{2i} - x_i}$$

Ist nun $a = g^n$, so bedeutet das

$$g^{n \cdot (y_i - y_{2i})} = g^{x_{2i} - x_i}$$

und damit

$$n \cdot (y_i - y_{2i}) = x_{2i} - x_i \mod N \quad (\text{G.1})$$

Im Bereich $\{0, \dots, p-1\}$ kann es mehrere Werte von n geben, die diese Gleichung lösen. Unter den Lösungen ist aber auch der gesuchte Wert von $\log_g(a)$.

Beispiel G.8. Wir betrachten wieder $p = 2027$, den Erzeuger $g = 7$ von $E(\mathbb{F}_{2027})$ und $a = 1133$ und führen Pollards Algorithmus mit der Zahl $x_0 = 17$ durch. Der Algorithmus wird nach 36 Schritten beendet und liefert $b_{36} = b_{72} = 1684$ sowie die Zahlen

$$x_{36} = 88, x_{72} = 1453, y_{36} = 1928, y_{72} = 533$$

Zur Lösung unseres Problem suchen wir also die Lösungen n von

$$n \cdot (1928 - 533) = 1453 - 88 \mod 2026$$

also von

$$n \cdot 1395 = 1365 \mod 2026 \quad (\text{G.2})$$

Da $\text{ggT}(1395, 2026) = 1$, ist 1395 invertierbar modulo 2026 und der euklidische Algorithmus liefert als Inverses den Wert

$$r = 1371$$

Damit hat Gleichung (G.2) die eindeutige Lösung

$$n = 1371 \cdot 1365 \mod 2026 = 1417$$

und wir haben $\log_7(1133)$ gefunden.

Hätten wir den Algorithmus mit dem zufälligen Wert $x_0 = 7$ gestartet, so hätte der Algorithmus (ebenfalls nach 36 Schritten) das Ergebnis $b_{36} = b_{72} = 1285$ sowie die Zahlen

$$x_{36} = 1122, x_{72} = 130, y_{36} = 333, y_{72} = 857$$

geliefert. Zu lösen ist nun die Gleichung

$$n \cdot (333 - 857) = 130 - 1122 \mod 2026$$

also von

$$n \cdot 524 = 992 \mod 2026 \quad (\text{G.3})$$

Da $\text{ggT}(524, 2026) = 2$, ist 524 nicht invertierbar modulo 2026, und wir erhalten daher in diesem Fall mehrere Lösungen (genau zwei Stück), nämlich

$$n_1 = 442, n_2 = 1417$$

In diesem Fall ist durch Ausprobieren zu verifizieren, dass $n = n_2 = 1417$ die gesuchte Lösung ist.

Bemerkung G.7. Der Pollard- ρ -Algorithmus hat eine Laufzeit von $\mathcal{O}(\sqrt{p})$, ist also von der Komplexität her vergleichbar mit dem Babystep-Giantstep-Algorithmus. Allerdings hat er einen sehr viel geringeren Speicherbedarf.

Bemerkung G.8. Je mehr Teiler die Gruppenordnung N der zyklischen Gruppe G hat desto mehr Lösungen für die Gleichung (G.1), also für

$$n \cdot (y_i - y_{2i}) = x_{2i} - x_i \mod N$$

kann es geben. Unter Umständen kann es dann einfacher sein, den Algorithmus mit einem neuen Zufallswert x_0 zu wiederholen.

Am besten funktioniert der Algorithmus für den Fall, dass N eine Primzahl ist, denn dann hat die Gleichung (G.1) nur dann eine eindeutige Lösung, wenn $y_i = y_{2i}$ ist. Allerdings ist $p - 1$ immer durch 2 teilbar, und daher ist die Ordnung von $E(\mathbb{F}_p)$ nur im uninteressanten Fall $p = 3$ eine Primzahl.

G.4. Der Pohlig–Hellman–Algorithmus

Der Ansatz von Pohlig und Hellman ist weniger ein komplett neuer und eigenständiger Algorithmus zur Bestimmung des diskreten Logarithmus als vielmehr ein Verfahren, Algorithmen, die für Gruppen vergleichsweise kleiner Ordnung gut funktionieren, auf Gruppen größerer Ordnung auszudehnen.

Zunächst betrachten wir eine zyklische Gruppe G der Ordnung p^e mit einer Primzahl p und einen Erzeuger $g \in G$. Schreiben wir ein Elemente $a \in G$ in der Form

$$a = g^n \quad \text{für ein } n \in \{0, 1, \dots, p^e - 1\}$$

so können wir n auch p -adisch darstellen als

$$n = n_0 + n_1 \cdot p + n_2 \cdot p^2 + \dots + n_{e-1} \cdot p^{e-1}$$

mit $0 \leq n_i \leq p - 1$. Daraus erhalten wir

$$p^{e-1} \cdot n = n_0 \cdot p^{e-1} + n_1 \cdot p^e + \dots + n_{e-1} \cdot p^{2e-2} = n_0 \cdot p^{e-1} + p^e \cdot (n_1 + n_2 \cdot p + \dots + n_{e-1} \cdot p^{e-2})$$

Setzen wir jetzt $n' = n_1 + n_2 \cdot p + \dots + n_{e-1} \cdot p^{e-2}$, so erhalten wir

$$\begin{aligned} a^{p^{e-1}} &= g^{n \cdot p^{e-1}} = g^{n_0 \cdot p^{e-1} + p^e \cdot n'} \\ &= g^{n_0 \cdot p^{e-1}} \cdot g^{p^e \cdot n'} = (g^{p^{e-1}})^{n_0} \cdot (g^{p^e})^{n'} \\ &= (g^{p^{e-1}})^{n_0} \end{aligned}$$

(wobei wir ausgenutzt haben, dass nach dem Satz von Fermat $g^{p^e} = 1$). Setzen wir nun noch

$$a' = g^{-n_0} \cdot a$$

so erhalten wir

$$\begin{aligned} a' &= g^{-n_0} \cdot g^n \\ &= g^{-n_0} \cdot g^{n_0 + n_1 \cdot p + n_2 \cdot p^2 + \dots + n_{e-1} \cdot p^{e-1}} \\ &= g^{n_1 \cdot p + n_2 \cdot p^2 + \dots + n_{e-1} \cdot p^{e-1}} \\ &= g^{p \cdot n'} \\ &= (g^p)^{n'} \end{aligned}$$

Damit haben wir das Problem der Bestimmung von n in zwei Teilproblem aufgespaltet, nämlich

1. Bestimme $n_0 \in \{0, \dots, p-1\}$ so, dass $a^{p^{e-1}} = (g^{p^{e-1}})^{n_0}$.
2. Bestimme $n' \in \{0, \dots, p^{e-1}-1\}$ so, dass $g^{-n_0} \cdot a = (g^p)^{n'}$.

Das Problem (1) ist wieder ein diskretes Logarithmusproblem, aber jetzt in der (im allgemeinen) sehr viel kleineren zyklischen Gruppe G_1 , die von $g^{p^{e-1}}$ erzeugt wird. Diese Gruppe hat die Ordnung p (also Primzahlordnung), und da kann das Problem in der Regel recht effektiv mit dem Pollard- ρ -Algorithmus behandelt werden.

Das Problem (2) ist auch ein diskretes Logarithmusproblem, aber ebenfalls in einer kleineren zyklischen Gruppe, nämlich der von g^p erzeugten zyklischen Gruppe, die die Ordnung p^{e-1} hat. Falls $e-1 > 1$, kann das Problem (2) ferner nach dem gleichen Prinzip weiter aufgespalten werden in die Berechnung von n_1 und einen Rest n'' . Insgesamt kann man das solange reduzieren, bis nur noch Gruppen von Primzahlordnung zu untersuchen sind. Daraus leitet sich folgender Algorithmus ab:

Pohlig–Hellman–Algorithmus für Primzahlpotenzen Gegeben ist eine zyklische Gruppe G der Ordnung p^e mit Erzeuger g und ein $a \in G$. Gesucht ist ein n mit $a = g^n$.

1. Setze $x_0 = 0$.
2. Berechne $h = g^{p^{e-1}}$ (und beachte, dass h die Ordnung p hat).
3. Für $k = 0, \dots, e-1$ führe die folgenden Schritte durch:
 - berechne $a_k = (g^{-x_k} \cdot a)^{p^{e-1-k}}$ und beachte, dass dieses Element die Ordnung 1 oder p hat (also in der von h erzeugten Gruppe G_1 liegt).
 - Berechne (z.B. mit Pollards- ρ -Algorithmus oder dem Babystep–Giantstep–Algorithmus) das Element n_k mit $a_k = h^{n_k}$.
 - Setze $x_{k+1} = x_k + n_k \cdot p^k$.

4. Setze $n = x_e$.

Die Vorüberlegungen stellen sicher, dass n tatsächlich der diskrete Logarithmus von a bezüglich g ist.

Bemerkung G.9. Die Komplexität des Pohlig–Hellman–Algorithmus für Primzahlpotenzen p^e (bei Anwendung von Pollards– ρ –Algorithmus in Schritt (2)) ist $\mathcal{O}(e \cdot \sqrt{p})$.

Für das diskrete Logarithmusproblem in $E(\mathbb{F}_p)$ ist dieser Teil alleine noch nicht sehr hilfreich, denn $p - 1$ ist immer gerade. Da es wenige Primzahlen der Form $p = 2^e + 1$ gibt, kann dieser Ansatz also selten zu Anwendung kommen und wir benötigen noch eine Ausdehnung auf beliebige zyklische Gruppen der Ordnung N . Dazu betrachten wir die eindeutige Primfaktorzerlegung

$$N = p_1^{e_1} \cdot p_2^{e_2} \cdots p_t^{e_t}$$

mit $p_1 < p_2 < \dots < p_t$ und $e_i > 0$. Wir entwickeln nun den allgemeinen Algorithmus durch Induktion nach t , der Anzahl der Primteiler von N .

Wir setzen $m = p_2^{e_2} \cdots p_t^{e_t}$, sodass also

$$N = p_1^{e_1} \cdot m$$

Ist nun g ein Erzeuger von G und $a \in G$ gegeben, so setzen wir

$$g_1 = g^m, \quad g_2 = g^{p_1^{e_1}}$$

und

$$a_1 = a^m, \quad a_2 = a^{p_1^{e_1}}$$

Ist dann $a = g^n$, so gilt offensichtlich auch $a_1 = g_1^n$ und $a_2 = g_2^n$, dh. a_i ist in der von g_i erzeugten zyklischen Gruppe G_i .

Die Gruppe G_1 hat die Primzahlpotenzordnung $p_1^{e_1}$, denn

$$g_1^{e_1} = (g^m)^{p_1^{e_1}} = g^{m \cdot p_1^{e_1}} = g^N = 1$$

und daher können wir nach dem Pohlig–Hellman Algorithmus für Primzahlpotenzen ein $n_1 \in \{0, \dots, p^e - 1\}$ bestimmen mit $a_1 = g_1^{n_1}$. Die Ordnung m der Gruppe G_2 ist nicht unbedingt eine Primzahlpotenz, aber in der Primfaktorzerlegung von m kommt eine Primzahl weniger vor also in der Primfaktorzerlegung von N . Wir nehmen jetzt induktiv an, dass wir für $t - 1$ verschiedene Primteiler das Problem schon gelöst haben und also ein $n_2 \in \{0, \dots, m - 1\}$ finden können mit $a_2 = g_2^{n_2}$.

Da $p_1^{e_1}$ und m teilerfremd sind, gibt es nach dem chinesischen Restsatz genau ein $n \in \{0, \dots, N-1\}$ mit

$$n = n_1 \bmod p_1^{e_1}, \quad n = n_2 \bmod m$$

Hierfür gilt nun

$$(g^{-n} \cdot a)^{p_1^{e_1}} = (g^{p_1^{e_1}})^{-n} \cdot a^{p_1^{e_1}} = g_2^{-n} \cdot a_2 = g_2^{-n_2} \cdot a_2 = 1$$

Beachten Sie dabei, dass $g_2^m = 1$, sodass für alle a, b mit $a = b \bmod m$ gilt $g^a = g^b$, speziell also auch $g_2^{-n} = g_2^{-n_2}$. Entsprechend gilt

$$(g^{-n} \cdot a)^m = 1$$

Wählen wir daher mit dem euklidischen Algorithmus ganze Zahlen r, s mit $1 = r \cdot p_1^{e_1} + s \cdot m$, so gilt

$$\begin{aligned} g^{-n} \cdot a &= (g^{-n} \cdot a)^{r \cdot p_1^{e_1} + s \cdot m} \\ &= (g^{-n} \cdot a)^{r \cdot p_1^{e_1}} \cdot (g^{-n} \cdot a)^{s \cdot m} \\ &= ((g^{-n} \cdot a)^{p_1^{e_1}})^r \cdot ((g^{-n} \cdot a)^m)^s \\ &= 1 \end{aligned}$$

woraus folgt

$$a = g^n$$

Beispiel G.9. Wir betrachten wieder $p = 2027$, den Erzeuger $g = 7$ von $E(\mathbb{F}_{2027})$ und $a = 1133$. In diesem Fall ist die Primfaktorzerlegung

$$p - 1 = 2026 = 2 \cdot 1013$$

In diesem Fall ist

$$g_1 = 7^{1013} = 2026, \quad a_1 = 1133^{1013} = 2026$$

so dass ersichtlich $a_1 = g_1^1$ (also $n_1 = 1$). Ferner ist

$$g_2 = 7^2 = 49, \quad a_2 = 1133^2 = 598$$

Anwendung von Pollards- ρ -Algorithmus oder des Babystep-Giantstep-Algorithmus liefert

$$a_2 = g_2^{404}$$

also $n_2 = 404$. Euklid liefert

$$1 = 1 \cdot 1013 - 506 \cdot 2$$

Daher finden wir das gesuchte n , indem wir zunächst

$$n' = n_1 \cdot 1 \cdot 1013 - n_2 \cdot 506 \cdot 2 = 1015 - 404 \cdot 506 \cdot 2 = -407835$$

berechnen und dann

$$n = n' \bmod 2026 = 1417$$

ermitteln.

Bemerkung G.10. Mit einer Variante des chinesischen Restsatzes, die eine Zerlegung von N in mehr als zwei teilerfremde Faktoren berücksichtigt, lässt sich der allgemeine Fall schnell direkt auf die einzelnen Primteiler zurückführen.

Beispiel G.10. Wir betrachten die Primzahl $p = 2017$. Die Einheitengruppe $E(\mathbb{F}_{2017})$ wird erzeugt von $g = 19$. In diesem Fall gilt

$$p - 1 = 2016 = 2^5 \cdot 3^2 \cdot 7$$

dh. $p - 1$ hat drei Primteiler.

Wir betrachten die Zahl $a = 1133$ und suchen $\log_g(a)$, also die Zahl n mit $19^n = 1133$.

Passend zu den drei Primteilern setzen wir

$$\begin{aligned} m_1 &= \frac{2018}{2^5} = 63, & g_1 = g^{m_1} &= 500, & a_1 = a^{m_1} &= 528 \\ m_2 &= \frac{2018}{3^2} = 224, & g_2 = g^{m_2} &= 24, & a_2 = a^{m_2} &= 1005 \\ m_3 &= \frac{2018}{7} = 288, & g_3 = g^{m_3} &= 79, & a_3 = a^{m_3} &= 1879 \end{aligned}$$

Dann erzeugen die Elemente g_i Gruppen G_i der Ordnung m_i in $E(\mathbb{F}_{2017})$ und a_i liegt in dieser Gruppe. Diese Gruppen G_i haben alle Primzahlpotenzordnung und daher können wir den Pohlig–Hellman–Algorithmus für Primzahlpotenzen darauf anwenden. Wir führen das für G_2 aus:

Die Gruppe G_2 hat die Ordnung $9 = 3^2$, also ist $p = 3$ und $e = 2$. Der Erzeuger ist das Element $g_2 = 24$ und der diskrete Logarithmus wird berechnet für $\alpha = a_2 = 1005$. Wir berechnen daher

$$h = g^{p^{e-1}} = 24^3 = 1722$$

und setzen $x_0 = 0$.

$k = 0$: Setze $\alpha_0 = (g_2^{-0} \cdot \alpha)^{p^{e-1}} = \alpha^3 = 1722$. Es ist offensichtlich, dass

$$\alpha_0 = h^1$$

sodass also $n_0 = 1$ und

$$x_1 = x_0 + n_0 \cdot p^0 = 1 \cdot 3^0 = 1$$

$k = 1$: Setze $\alpha_1 = (g_2^{-1} \cdot \alpha)^{p^{2-2}} = g_2^{-1} \cdot \alpha = 294$. Durch Ausprobieren erhalten wir hier $\alpha_2 = h^2$, also $n_1 = 2$ und damit

$$n = x_2 = x_1 + n_1 \cdot p = 1 + 2 \cdot 3 = 7$$

In der Tat überprüft man leicht, dass $a_2 = g_2^7$.

Entsprechend erhalten wir $a_1 = g_1^{26}$ und $a_3 = g_3^5$.

Aus der allgemeinen Form des chinesischen Restsatzes erhalten wir die Zahl $n = 250$, die die folgenden simultanen Kongruenzen erfüllt:

$$250 \equiv 7 \pmod{3^2}, \quad 250 \equiv 26 \pmod{2^5}, \quad 250 \equiv 5 \pmod{7}$$

und hierfür gilt

$$a = g^{250} \quad \text{in } \mathbb{F}_{2017}$$

G.5. Index–Calculus

Alle bis jetzt vorgestellten Algorithmen sind ganz allgemein für beliebige zyklische Gruppen anwendbar und keiner nimmt Bezug auf die Gruppe $E(\mathbb{F}_p)$, für die das diskrete Logarithmus–Problem am interessantesten ist. Das nun vorgestellte Verfahren, der **Index–Calculus** geht dagegen auf Struktur und Darstellung von $E(\mathbb{F}_p)$ ein und bringt das diskrete Logarithmus–Problem mit der Faktorzerlegung in Verbindung. Es handelt sich dabei weniger um einen konkreten Algorithmus als vielmehr um eine allgemeine Vorgehensweise, die in verschiedenen Algorithmen konkretisiert und weiterentwickelt wurde. Diese konkrete Verfahren übersteigen jedoch den Rahmen dessen, was in dieser Vorlesung behandelt werden kann. Der grundsätzliche Ansatz ist dabei, zunächst die diskreten Logarithmen für bestimmte Hilfsprimzahlen mithilfe linearer Algebra zu bestimmen und daraus den gesuchten diskreten Logarithmus abzuleiten.

Index–Calculus

Gegeben ist eine Primzahl p und ein Erzeuger g von $E(\mathbb{F}_p)$ sowie eine Zahl $a \in E(\mathbb{F}_p)$. Gesucht ist ein $n \in \mathbb{N}$ mit $a = g^n$.

1. Bestimme eine Faktorbasis $\mathcal{F} = \{q_1 = 2, q_2 = 3, \dots, q_t\}$ bestehend aus den r ersten Primzahlen.
2. Benutze lineare Algebra, um für $i = 1, \dots, t$ Zahlen x_i so zu bestimmen, dass $q_i = g^{x_i}$ (in $E(\mathbb{F}_p)$).
3. Für $s = 0, 1, 2, \dots$ führe die folgenden Rechnungen durch
 - Berechne $a_s = g^s \cdot a$.
 - Falls $a_s = q_1^{m_1} \cdot q_2^{m_2} \cdots q_t^{m_t} \pmod{p}$, setze

$$n = m_1 \cdot x_1 + m_2 \cdot x_2 + \cdots + m_t \cdot x_t - s \pmod{p-1}$$

STOPP, n ist die gesuchte Zahl.

Zunächst ist klar, dass n tatsächlich die gesuchte Zahl ist, denn nach Definition der x_i folgt

$$q_1^{m_1} \cdot q_2^{m_2} \cdots q_t^{m_t} = (g^{x_1})^{m_1} \cdots (g^{x_t})^{m_t} = g^{m_1 \cdot x_1 + \cdots + m_t \cdot x_t}$$

sodass aus

$$g^s \cdot a = q_1^{m_1} \cdot q_2^{m_2} \cdots q_t^{m_t}$$

folgt, dass

$$a = g^{m_1 \cdot x_1 + \cdots + m_t \cdot x_t - s}$$

Lineare Algebra zur Bestimmung der x_i kann dabei wie folgt eingesetzt werden:

Für $k \geq 1$ berechne g^k und untersuche, ob g^k in $E(\mathbb{F}_p)$ mit der Faktorbasis geschrieben werden kann,

$$g^k = q_1^{r_{k,1}} \cdot q_2^{r_{k,2}} \cdots q_t^{r_{k,t}} \mod p$$

Ist das der Fall, so heißtt

$$r_k = (r_{k,1}, r_{k,2}, \dots, r_{k,t})$$

eine **Relation** für g^k . Ist nämlich $p_i = g^{x_i}$, so folgt daraus

$$g^k = (g^{x_1})^{r_{k,1}} \cdots (g^{x_t})^{r_{k,t}} = g^{r_{k,1} \cdot x_1 + \cdots + r_{k,t} \cdot x_t}$$

und daraus folgt, dass die Zahlen x_1, \dots, x_t die folgende Gleichung erfüllen

$$k = r_{k,1} \cdot x_1 + \cdots + r_{k,t} \cdot x_t \mod p - 1 \tag{G.4}$$

Ziel ist es nun, t Gleichungen (G.4) zu finden, dass das resultierende Gleichungssystem modulo $p - 1$ eine eindeutige Lösung hat. Bei dieser Lösung muss es sich dann um die gesuchten diskreten Logarithmen der q_i handeln.

Beispiel G.11. Wir greifen wieder das Beispiel G.7 mit $p = 2027$, dem Erzeuger $g = 7$ von $E(\mathbb{F}_{2027})$ und $a = 1133$ auf. Als Faktorbasis wählen wir die ersten acht Primzahlen, also

$$\mathcal{F} = \{2, 3, 5, 7, 11, 13, 17, 19\}$$

und berechnen dazu einige Relationen

k	r_k
1	(0, 0, 0, 1, 0, 0, 0, 0)
4	(1, 0, 0, 0, 1, 0, 1, 0)
8	(0, 0, 0, 0, 0, 1, 0, 0)
24	(1, 0, 1, 0, 0, 0, 1, 0)
37	(1, 0, 0, 0, 0, 0, 0, 2)
40	(5, 0, 0, 0, 1, 0, 0, 0)
44	(7, 1, 1, 0, 0, 0, 0, 0)
50	(1, 1, 0, 0, 1, 0, 0, 1)

Beachten Sie dabei, dass wir hier einige Relationen weggelassen haben. So erhalten wir z.B. für $k = 2$ die Beziehung $g^2 = 49 = 7^2$, also die Relation $r_2 = (0, 0, 0, 2, 0, 0, 0, 0)$. Diese liefert die Gleichung

$$2 \cdot x_4 = 2$$

die aber äquivalent zur Gleichung

$$x_4 = 1$$

ist, die wir schon aus Relation r_1 erhalten. Alle Relationen, die also Gleichungen liefern, die sich schon aus den vorherigen Relationen ergeben, werden daher weggelassen. Natürlich werden auch die Zahlen k , für die sich g^k nicht mit der Faktorbasis schreiben lässt, weggelassen.

Bezeichnen wir mit R die 8×8 -Matrix, die diese Relationen r_1, r_4, \dots, r_{50} als Zeilen hat, und setzen wir

$$\vec{b} = (1, 4, 8, 24, 37, 40, 44, 50)^\top$$

so erhalten wir $\det(R) = -13$. Da -13 teilerfremd zu 2026 ist, hat das Gleichungssystem

$$R \cdot \vec{x} = \vec{b}$$

modulo 2026 eine eindeutige Lösung und mithilfe des Eliminationsalgorithmus (der sich bei entsprechender Sorgfalt auch modulo 2026 anwenden lässt) erhalten wir diese Lösung als

$$\vec{x} = (1869, 298, 845, 1, 825, 8, 1362, 1110)^\top$$

und damit für unsere Faktorbasiselemente q_i die folgenden diskreten Logarithmen

$$\begin{aligned}\log_7(2) &= 1869 \\ \log_7(3) &= 298 \\ \log_7(5) &= 845 \\ \log_7(7) &= 1 \\ \log_7(11) &= 825 \\ \log_7(13) &= 8 \\ \log_7(17) &= 1362 \\ \log_7(19) &= 1110\end{aligned}$$

Bei der Berechnung der Werte $g_2 = g^s \cdot a$ in \mathbb{F}_{2027} ergibt sich

$$g_4 = 99 = 3^2 \cdot 11$$

als der erste Wert, der über der Faktorbasis zerfällt, woraus wir erhalten

$$n = 2 \cdot \log_7(3) + \log_7(11) - 4 = 2 \cdot 298 + 825 - 4 = 1417 \bmod 2026$$

Wir haben also auch hier wieder $n = 1417$ erhalten.

Bemerkung G.11. Der Index–Calculus–Algorithmus kann mit subexponentieller Laufzeit in der Größenordnung von

$$T = \exp \left((c + o(1)) \cdot \sqrt{\ln(N)} \cdot \sqrt{\ln(\ln(N))} \right)$$

(mit $N = p-1 = |E(\mathbb{F}_p)|$) mit einer Konstanten c , die von der Implementierung abhängt. Modifikationen und Weiterentwicklungen der Index–Calculus–Methode (**Zahlkörpersieb** und **Funktionenkörpersieb**) lösen das Problem des diskreten Logarithmus über allgemeinen endlichen Körpern sogar in einer Laufzeit

$$T = \exp \left((c + o(1)) \cdot \sqrt[3]{\ln(N)} \cdot \sqrt[3]{\ln(\ln(N))^2} \right)$$

Das sind die schnellsten bekannten (lauffähigen) Algorithmen zur Lösung des diskreten Logarithmus–Problems.

Für allgemeine Primzahlen der binären Größenordnung 2048 ist das diskrete–Logarithmus–Problem damit noch nicht in überschaubarer Zeit lösbar.

Quantenalgorithmen zur Lösung des Problems, die eine polynomiale Laufzeit haben, sind aktuell und in absehbarer Zukunft noch nicht technisch realisierbar.

H. Polynome und Kurven

In diesem Abschnitt betrachten wir einen beliebigen Körper k , z.B. $k = \mathbb{R}$, $k = \mathbb{C}$, $k = F_p$ für eine Primzahl p oder $k = F_q$ für eine Primzahlpotenz $q = p^e$ (in der Regel $q = 2^e$).

H.1. Polynome in zwei Variablen

Definition H.1. Ein **Polynom** $F(X, Y)$ in zwei Variablen X und Y mit Koeffizienten aus k ist ein Ausdruck

$$F(X, Y) = \sum_{i=0}^n \sum_{j=0}^m a_{i,j} \cdot X^i \cdot Y^j$$

mit $m, n \in \mathbb{N}$ und $a_{i,j} \in k$.

Zwei Polynome

$$F(X, Y) = \sum_{i=0}^n \sum_{j=0}^m a_{i,j} \cdot X^i \cdot Y^j, \quad G(X, Y) = \sum_{i=0}^{n'} \sum_{j=0}^{m'} b_{i,j} \cdot X^i \cdot Y^j$$

heißen **gleich**, wenn $n = n'$, $m = m'$ und wenn

$$a_{i,j} = b_{i,j} \quad \text{für alle } i \in \{0, \dots, n\}, j \in \{0, \dots, m\}$$

Für ein Polynom $F(X, Y) = \sum_{i=0}^n \sum_{j=0}^m a_{i,j} \cdot X^i \cdot Y^j$ heißt

$$\deg(F) = \max\{i + j \mid a_{i,j} \neq 0\}$$

der **Grad** von $F(X, Y)$.

Beispiel H.1.

- a) $F(X, Y) = 2 + \frac{1}{3} \cdot X - \frac{7}{3} \cdot X \cdot Y^2 + \pi \cdot X^4 \cdot Y^3 + Y^4$ ist ein Polynom vom Grad 7 mit Koeffizienten aus \mathbb{R} .
- b) $F(X, Y) = 1 + X + X^2Y + XY^4 + X^5Y^6$ ist ein Polynom vom Grad 11 über \mathbb{F}_2 .
- c) $F(X, Y) = 0$ ist ein Polynom über jedem Körper (das **Nullpolynom**). Für das Nullpolynom ist der Grad zunächst nicht definiert. Üblicherweise benutzt man die Konvention, dass das Nullpolynom jeden Grad haben kann.
- d) $F(X, Y) = 1$ ist ein Polynom über jedem Körper (das **Einspolynom**). Es hat den Grad 0.

Definition H.2. Für einen Körper k heißt

$$k[X, Y] = \{F(X, Y) \mid F(X, Y) \text{ ist ein Polynom mit Koeffizienten aus } k\}$$

der **Polynomring in zwei Variablen** über k .

Bemerkung H.1. $k[X, Y]$ ist ein kommutativer Ring mit 1.

Sind $r, s \in k$ Elemente und $F(X, Y) \in k[X, Y]$ ein Polynom, so können r, s in $F(X, Y)$ eingesetzt werden und wir erhalten eine Zahl

$$F(r, s) = \sum_{i=0}^n \sum_{j=0}^m a_{i,j} \cdot r^i \cdot s^j \in k$$

die *Auswertung von $F(X, Y)$ bei (r, s)* genannt wird.

Beispiel H.2.

a) Für das Polynom $F(X, Y) = 2 + \frac{1}{3} \cdot X - \frac{7}{3} \cdot X \cdot Y^2 + \pi \cdot X^4 \cdot Y^3 + Y^4 \in \mathbb{R}[X, Y]$ gilt

$$F(0, 0) = 2, \quad F(1, 0) = \frac{7}{3}, \quad F(0, 1) = 3, \quad F(1, 1) = 1 + \pi$$

b) Für das Polynom $F(X, Y) = 1 + X + X^2Y + XY^4 + X^5Y^6 \in \mathbb{F}_2[X, Y]$ gilt

$$F(0, 0) = 1, \quad F(1, 0) = 0, \quad F(0, 1) = 1, \quad F(1, 1) = 1$$

d) Für das Polynom $F(X, Y) = 2 + XY + X^2 + 2Y^4 \in \mathbb{F}_3[X, Y]$ gilt

$$\begin{array}{lll} F(0, 0) = 2 & F(1, 0) = 0 & F(2, 0) = 0 \\ F(0, 1) = 1 & F(0, 2) = 1 & F(1, 1) = 0 \\ F(1, 2) = 1 & F(2, 1) = 1 & F(2, 2) = 0 \end{array}$$

d) Für das Polynom $F(X, Y) = X^3Y^3 + 2XY^3 + 2X^3Y + XY \in \mathbb{F}_3[X, Y]$ gilt

$$\begin{array}{lll} F(0, 0) = 0 & F(1, 0) = 0 & F(2, 0) = 0 \\ F(0, 1) = 0 & F(0, 2) = 0 & F(1, 1) = 0 \\ F(1, 2) = 0 & F(2, 1) = 0 & F(2, 2) = 0 \end{array}$$

Das Polynom $F(X, Y)$ ist also nicht das Nullpolynom, trotzdem erhalten wir für alle $r, s \in \mathbb{F}_3$, dass $F(r, s) = 0$, dh. die Auswertung von $F(X, Y)$ verschwindet in allen Punkten $(r, s) \in \mathbb{F}_3^2$.

Definition H.3. Ein Punkt $(r, s) \in K^2$ heißt **Nullstelle** des Polynoms $F(X, Y) \in k[X, Y]$, wenn $F(r, s) = 0$.

Beispiel H.3.

- a) Der Punkt $(3, 4) \in \mathbb{R}^2$ ist Nullstelle des Polynoms $F(X, Y) = X^2 + Y^2 - 25 \in \mathbb{R}[X, Y]$.
- b) Der Punkt $(1, 2) \in \mathbb{F}_3^2$ ist Nullstelle des Polynoms $F(X, Y) = 1 + X + 2Y + 2X^2Y + XY^3 \in \mathbb{F}_3[X, Y]$.
- c) Der Punkt $(\alpha, \alpha+1) \in \mathbb{F}_4^2$ ist Nullstelle des Polynoms $F(X, Y) = XY+1 \in \mathbb{F}_4[X, Y]$, wobei wir den Körper \mathbb{F}_4 über die Relation $\alpha^2 = \alpha + 1$ definieren.

H.2. Ebene Kurven

Die Menge aller Nullstellen eines Polynoms ist für uns von besonderem Interesse.

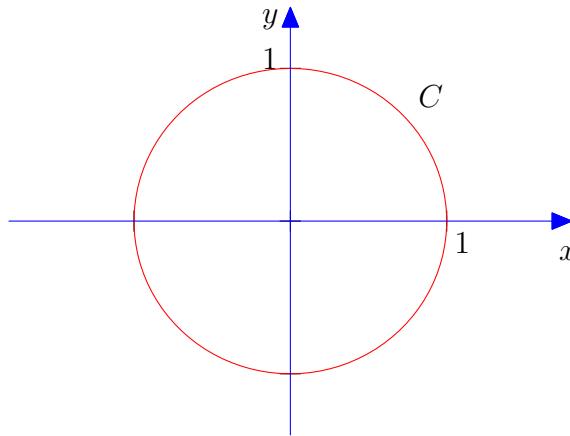
Definition H.4. Die **ebene (algebraische) Kurve** über k die durch das Polynom $F(X, Y) \in k[X, Y]$ gegeben ist, ist die Menge

$$C = \{(r, s) \in k^2 \mid F(r, s) = 0\}$$

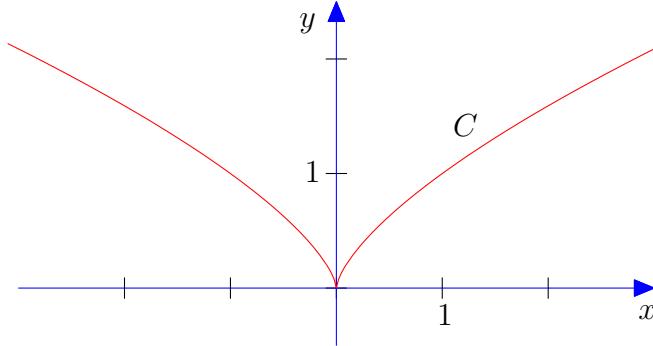
Bemerkung H.2. Die Kurve C , die durch ein Polynom $F(X, Y) \in k[X, Y]$ definiert wird, besteht also aus allen Punkten $(r, s) \in k^2$, die Nullstellen von $F(X, Y)$ sind, also aus allen Punkten $(r, s) \in k^2$ mit

$$F(r, s) = 0$$

Beispiel H.4. Die Kurve C , die durch das Polynom $F(X, Y) = X^2 + Y^2 - 1 \in \mathbb{R}[X, Y]$ definiert wird, ist der Einheitskreis:



Beispiel H.5. Die Kurve C , die durch das Polynom $F(X, Y) = Y^3 - X^2 \in \mathbb{R}[X, Y]$ definiert wird, hat die folgende Gestalt



Beispiel H.6. Die Kurve, die durch das Polynom

$$F(X, Y) = X^2Y + XY^2 + X^2 + X + Y + 1 \in \mathbb{F}_2[X, Y]$$

definiert wird, ist gegeben durch

$$C = \{(0, 1), (1, 1)\} \subseteq \mathbb{F}_2^2$$

Von dieser Kurve können wir uns kein Bild mehr machen.

Beispiel H.7. Wir betrachten wieder den Körper \mathbb{F}_4 gegeben durch $\alpha^2 = \alpha + 1$.

Die Kurve, die durch das Polynom $F(X, Y) = XY + 1 \in \mathbb{F}_4[X, Y]$ definiert wird, ist gegeben durch

$$C = \{(1, 1), (\alpha, \alpha + 1), (\alpha + 1, \alpha)\} \subseteq \mathbb{F}_4^2$$

Auch von dieser Kurve können wir uns kein Bild mehr machen.

Ist k ein endlicher Körper, so kann die Kurve C , die gegeben ist durch ein Polynom $F(X, Y) \in k[X, Y]$, durch einsetzen der (endlich vielen) Punkte $(r, s) \in k^2$ in $F(X; Y)$ bestimmt werden.

Beispiel H.8. Wir betrachten die Kurve C über \mathbb{F}_3 , die gegeben ist durch das Polynom

$$F(X, Y) = X^4Y + 2X^2Y^3 + XY + Y + 1 \in \mathbb{F}_2[X, Y]$$

Hierfür gilt

$$\begin{array}{lll} F(0,0) = 1 & F(0,1) = 1 & F(0,2) = 1 \\ F(1,0) = 1 & F(1,1) = 0 & F(1,2) = 2 \\ F(2,0) = 1 & F(2,1) = 1 & F(2,2) = 1 \end{array}$$

Die Kurve ist als

$$C = \{(1,1)\} \subseteq \mathbb{F}_3^2$$

Beispiel H.9. Wir betrachten den Körper \mathbb{F}_4 , gegeben durch die Relation $\alpha^2 = \alpha + 1$ und die Kurve C über \mathbb{F}_3 , die gegeben ist durch das Polynom

$$F(X, Y) = X^2Y + X + Y + \alpha \in \mathbb{F}_4[X, Y]$$

Hierfür gilt

$$\begin{array}{lll} F(0,0) = \alpha & F(0,1) = \alpha + 1 \\ F(0,\alpha) = 0 & F(0,\alpha+1) = 1 \\ F(1,0) = \alpha + 1 & F(1,1) = \alpha + 1 \\ F(1,\alpha) = \alpha + 1 & F(1,\alpha+1) = \alpha + 1 \\ F(\alpha,0) = 0 & F(\alpha,1) = \alpha \\ F(\alpha,\alpha) = \alpha + 1 & F(\alpha,\alpha+1) = 1 \\ F(\alpha+1,0) = 1 & F(\alpha+1,1) = \alpha + 1 \\ F(\alpha+1,\alpha) = 0 & F(\alpha+1,\alpha+1) = \alpha + 1 \end{array}$$

Die Kurve ist als

$$C = \{(0,\alpha), (\alpha,0), (\alpha+1,\alpha+1)\} \subseteq \mathbb{F}_4^2$$

Beispiel H.10. In Beispiel H.2 hatten wir schon das Polynom $F(X, Y) = X^3Y^3 + 2XY^3 + 2X^3Y + XY \in \mathbb{F}_3[X, Y]$ betrachtet. Für die hierdurch definierte Kurve C gilt

$$C = \mathbb{F}_3^2$$

denn jeder Punkt aus \mathbb{F}_3^2 ist Nullstelle dieses Polynoms.

Bemerkung H.3. Für endliche Körper mit vielen Elementen (wie sie in der Kryptologie üblicherweise verwendet werden) ist die Einsetzungsmethode nicht praktikabel.

Index

Advanced Encryption Standard, 74
AES, 74
Bytesubstitution, 78
Dechiffrierung, 86
Diffusion, 83
MixColumns, 84
Schlüsselfahrplan, 80
Shift–Row, 83
Verschlüsselung, 85
asymmetrische Verschlüsselung, 41

Babystep–Giantstep–Algorithmus, 335
Blockchiffren, 27
brute force–Angriff, 12

Caesar–Verschlüsselung, 8
Carmichael–Zahl, 308
CBC–Mode, 31
Charakteristik, 276
chinesischer Restsatz, 262
chosen–ciphertext–Angriff, 44
chosen–message–Angriff, 117
chosen–plaintext–Angriff, 43
ciphertext–only–Angriff, 43

Data Encryption Standard, 49
DES, 49
 S –Box, 50
Dechiffrierung, 67
erfolgreiche Angriffe, 72
Rundenfahrplan, 61
Schlüsselfahrplan, 54

DHKE, 106
Schlüsselaustausch, 106
Schlüsselerzeugung, 106

Diffie–Hellman–Merkle–Verfahren, 106
Diffie–Hellman–Problem, 108
Diffie–Hellman–Schlüsselaustausch, 106
mit elliptischen Kurven, 206
Diffie–Hellman–Verfahren, 106
Diffie–Hellman–Entscheidungsproblem, 108
Digital Signature Algorithm, 143
diskreter Logarithmus, 329
diskretes Logarithmus–Problem, 329
DSA, 143

EBC–Mode, 29
ECDH, 206
ECDSA, 221
Schlüsselerzeugung, 221
Signatur, 222
Verifikation, 223

Einheit, 254
Einwegfunktion, 89
Falltür, 91

ElGamal–Signatur, 118
existentielle Fälschung, 121
Schlüsselerzeugung, 118
Schulbuch, 121
Signaturerzeugung, 118
Verifikation, 119

ElGamal–Verschlüsselung, 110
Entschlüsselung, 112
Schlüsselerzeugung, 111
Verschlüsselung, 111

elliptische Kryptosysteme, 199
digitale Signatur, 221
empfohlene Kurven, 229
Schlüsselaustausch, 207
unsichere Kurven, 228

elliptische Kurve, 157
 j -Invariante, 157
 Addition, 176
 Diskriminante, 157
 Ordnung, 182
 Primkörper-anomal, 228
 Primordnungsuntergruppe, 228
 projektiv, 175
 Vielfachenbildung, 199
 euklidischer Algorithmus, 256
 existentielle Fälschung, 117

 Faktorbasis, 324
 Faktorisierung
 Fermat–Verfahren, 316
 Pollards $p - 1$ –Verfahren, 319
 Probedivision, 314
 Quadratisches Sieb, 325
 Feistel–Netzwerk, 49
 Fermat–Test, 307

 Gitter, 230
 Basis, 230
 duales Gitter, 233
 Fundamentalmasche, 232
 Gruppe, 330
 endliche Gruppe, 331
 kommutativ, 331
 Ordnung einer Gruppe, 331
 zyklisch, 331

 Halbspur, 294
 Hash–Funktion, 125
 kryptographische, 129
 MAC, 150
 schwache Kollisionsresistenz, 127
 starke Kollisionsresistenz, 128
 Urbildresistenz, 127

 Hermite–Normalform, 246
 Index–Calculus, 344
 Relation, 345

 Kerkhoffsches Prinzip, 42
 known–plaintext–Angriff, 43
 Kompressionsfunktion, 129
 Kompressionsrate, 129
 Konkatenation, 28
 Kryptoanalyse, 26, 42
 differentiell, 72
 linear, 73
 Kryptographie, 26
 asymmetrische Verfahren, 88
 symmetrische Verfahren, 26
 Kurve
 ebene, 350
 elliptische, 157
 Körper, 266
 endlich, 266
 Körpererweiterung
 Minimalpolynom, 280
 Relation, 277

 Las–Vegas–Algorithmen, 90
 Lemma von Bezout, 257
 LFSR, 39
 linear–rückgekoppelte Schieberegister, 39
 LSFR
 Erzeugerpolynom, 40

 MAC, 148
 CBC–MAC, 153
 CMAC, 154
 Hash–Based Message Authentication
 Code, 152
 Hash–Funktion, 150

- HMAC, 152
- kryptographisch sicher, 148
- secret prefix, 150
- secret suffix, 150
- Symmetrieeigenschaft, 148
- Man–In–The–Middle, 110
- Merkle–Damgård–Metakonstruktion, 129
- Message Authentication Code, 148
- Miller–Rabin–Test, 309
- monoalphabetische Substitution, 12
- Monte–Carlo–Algorithmen, 90
- NIST, 74
- Nullteiler, 254
- One–Time–Pad, 44
- Orthogonalitätsdefekt, 239
- Padding
 - OAEP, 100
- Pohlig–Hellman–Algorithmus, 339, 340
- Pollard– ρ –Algorithmus, 336
- Pollards– ρ –Algorithmus, 337
- Polynom, 348
 - Auswertung, 349
 - Einspolynom, 348
 - Grad, 348
 - Nullpolynom, 348
 - Nullstelle, 349
 - Polynomring, 349
- Primkörper, 266
- Primzahl, 301
- Pseudo–Primzahl, 308
 - starke, 310
- Pseudozufallszahlengeneratoren, 38
- Public–Key Cryptography Standards, 147
- Quadratwurzel, 268
- Rabin–Verfahren, 102
- Relation
 - irreduzibel, 279
- Rijndael–Körper, 74, 281
- Inversentabelle, 77
- RSA, 93
 - chinesischer Restsatz, 97
 - Entschlüsselung, 96
 - Exponenten, 96
 - Formbarkeit, 99
 - Modul, 94
 - OAEP–Padding, 100
 - Schlüsselerzeugung, 94
 - Verschlüsselung, 95
 - RSA–PSS, 147
 - RSA–Signatur, 115
 - existentielle Fälschung, 117
 - Schlüsselerzeugung, 115
 - Schulbuch, 117
 - Sicherheit, 117
 - Signaturerzeugung, 115
 - Signaturverifikation, 116
 - RSA–Verfahren, 93
 - Schulbuch, 99
 - Schwamm–Konstruktion, 135
 - Aufsaugphase, 137
 - Auspressphase, 137
 - Keccak, 139
 - Secure Hash Algorithm 1, 132
 - Secure Hash Algorithm 3, 139
 - Siebintervall, 324
 - Skytale, 7
 - Spur, 289
 - Stromchiffren, 37
 - Untergruppe, 332

Vigenère–Verschlüsselung, 16

Kasiski–Angriff, 20

zusammengesetzte Zahl, 301

zyklische Gruppe, 331

Erzeuger, 331

Äquivalenzklasse, 253

Repräsentant, 253

Repräsentantensystem, 253

Äquivalenzrelation, 253