MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Kryptographie durch die Jahrhunderte: Mathematische Analysen historischer und moderner Verschlüsselungsverfahren“

verfasst von / submitted by

Katharina Becker BEd

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Education (MEd)

Wien, 2024 / Vienna, 2024

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 199 514 520 02
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Lehramt Sek (AB) UF Informatik UF Mathematik
Betreut von / Supervisor:	ao. Univ.-Prof. Mag. Dr. Peter Raith

Abstract

Bei der Übermittlung von Nachrichten und Informationen gibt es immer Personen, die an Informationen kommen wollen, die ursprünglich nicht für sie gedacht waren. Damit Dritte nicht an geheime Informationen kommen, werden schon seit vielen Jahren Nachrichten verschlüsselt. Dazu gibt es verschiedene Methoden, die auf verschiedenen mathematischen Prinzipien aufbauen. Diese Arbeit beschäftigt sich damit, welche mathematischen Voraussetzungen für die verschiedenen Verschlüsselungsmethoden benötigt werden. Auch die Sicherheit der einzelnen Verschlüsselungen wird beleuchtet. Anhand einer Literaturrecherche wurden die Informationen zusammengetragen.

Abstract - Englisch

The safety and discretion of transmitting information has for centuries been a driving force in mathematically based encryption. In the digital age it has become even more crucial to prevent individuals from accessing information that was not originally intended for them. Therefore, encrypting messages has played an important role in preventing third parties from accessing confidential information. The purpose of this paper is to explore different mathematical principles on which various methods of encryption are based. The main focus lies on the mathematical prerequisites for different encryption methods as well as examining the security of each encryption technique, of which the latter was achieved by reviewing publications and literature on this subject.

Inhaltsverzeichnis

1 Einleitung

Schon im antiken Griechenland war es wichtig Nachrichten vor Feinden zu schützen. Daher haben die Spartaner im 5. Jahrhundert vor Christus eine Methode gefunden, wie sie Nachrichten und Informationen übermitteln können, ohne dass Dritte diese direkt lesen können, falls die Nachricht in ihre Hände fällt. Auch Julius Cäsar war es im 1. Jahrhundert vor Christus wichtig, dass er seine Feldstrategien mithilfe von Boten sicher versenden konnte. Daher entwickelte er die Cäsar-Verschlüsselung. Auch die Vingère-Verschlüsselung zählt zu den historischen Verschlüsselungsmethoden und bietet eine weitere Möglichkeit Nachrichten vor Dritten zu schützen.

Im heutigen digitalen Zeitalter ist es umso wichtiger geworden, Nachrichten und Informationen zu verschlüsseln, damit sie nicht an Fremde geraten. Sensible Daten sollen somit geschützt werden. Daher haben sich seit dem 20. Jahrhundert viele verschiedene Verschlüsselungstechnologien entwickelt. Da sich nicht nur Verschlüsselungsmethoden weiter entwickeln, sondern auch Strategien, um dennoch an die Informationen zu gelangen, ist der Sicherheitsaspekt der verschiedenen Verschlüsselungen sehr wichtig.

Grob gesagt gibt es zwei verschiedene Arten von Verschlüsselungsmehtoden. Zum einen die symmetrischen Verschlüsselungen und zum anderen die asymmetrischen verschlüsselungen. Zu den symmetrischen Verschlüsselungen zählen vorallem die historischen Verfahren, da die für Entschlüsselung lediglich die Umkehrung der Verschlüsselung benötigt wird. Es gibt allerdings auch moderne symmetrische Verschlüsselungen. Komplexer sind jedoch asymmetrische Verschlüsselungen. Sie verwenden Einwegfunktionen, damit die Umkehrung nicht einfach vorzunehmen ist. Als Einwegfunktion wird eine mathematische Funktion beschrieben, die leicht in eine Richtung zu berechnen ist, jedoch schwierig in die Rückrichtung berechnet werden kann.

Durch eine umfassende Literaturrecherche werden in dieser Arbeit verschiedene historische und moderne Verschlüsselungsverfahren zusammengetragen. Dabei werden ihre Stärken und Schwächen beleuchtet. Die Arbeit beinhaltet zunächst alle mathematischen Voraussetzungen, die für die Verschlüsselungsmethoden notwendig sind. Dabei spielt die Gruppentheorie eine wichtige Rolle. Für benötigte mathematische Sätze werden Beweise geliefert.

Da Nachrichten aus Buchstaben nicht oder nur schwer verschlüsselt werden können, werden diese meist in Zahlen umgewandelt. Dazu gibt es verschiedene Möglichkeiten. Daher werden in dieser Arbeit verschiedene Zahlensysteme vorgestellt und die Umrechnung erläutert.

Das nächste Kapitel bildet die symmetrische Verschlüsselung. Diese unterteilt sich weiderrum in Transpositions- und Substitutionsverfahren. Die Arbeit erläutert die Unterschiede dieser beiden Verfahren und gibt Überblick über historische Verfahren wie die Skytale, Fleissner-Schablone, Cäsar-Verschlüsselung, Vignère-Verschlüsselung, Feistel-Chiffre, DES-Algorithmus und dem AES-Algorithmus.
Das Kapitel der asymmetrischen Verschlüsselung konzentriert sich auf das Diffie-Hellman-Verfahren, welches als Grundvoraussetzung der El-Gamal-Verschlüsselung dient, welche ebenfalls erläutert wird. Weiters werden die RSA-Verschlüsselung und das Rabin-Verfahren beleuchtet. Diese Technologien, die auf mathematischen Prinzipien wie der Schwierigkeit der Faktorisierung großer Zahlen basieren, ermöglichen sichere Kommunikation, ohne dass Absender und Empfänger denselben geheimen Schlüssel teilen müssen.

Zuletzt wird der Fokus auf die Sicherheit gelegt. Daher werden verschiedene Angriffstypen erläutert und welche Schlüsselgröße Sicherheit bietet. In dieser Arbeit sind viele Beispiele vorhanden, um die verschiedenen Verfahren anschaulich zu präsentieren.

2 Mathematische Voraussetzungen der Kryptografie

2.1 Definitionen, Sätze und Beweise

Verschlüsselungssysteme beruhen auf verschiedenenen mathematischen Gegebenheiten welche sie nutzen, damit Nachrichten für Dritte unleserlich werden. In diesem Kapitel werden zunächst Begriffe und Sätze definiert, die für verschiedene Verschlüsselungen benötigt werden.

Definition. Sei G eine nichtleere Menge und $\cdot$ eine Abbildung $G \times G \to G$. Dann nennt man $(G, \cdot)$ ein Gruppoid.

Definition. Sei (G, $\cdot$) ein Gruppoid. Ist die Verknüpfung assoziativ so wird (G, $\cdot$) Halbgruppe genannt. Es gilt also

$$\forall a, b, c \in G : (a \cdot b) \cdot c = a \cdot (b \cdot c)$$

Wenn das Assoziativgesetz gilt, ist das Setzen der Klammern nicht unbedingt notwendig. Statt $(a \cdot b) \cdot c$ kann also $a \cdot b \cdot c$ geschrieben werden.

Definition. Als neutrales Element oder Einselement wird ein Element e eines Gruppoids genannt für das folgendes gilt:

$$\forall a \in G : a \cdot e = e \cdot a = a$$

Für die Gruppoide (**N**, +), (**Z**, +), und (**R**, +) ist 0 das neutrale Element. In diesem Fall wird das neutrale Element nicht Einselement sondern Nullelement genannt.

Satz 2.1.1. *Eindeutigkeit des neutralen Elements*
Sei (G, $\cdot$) ein Gruppoid mit Einselement e bzw. neutralem Element, so ist dieses eindeutig bestimmt.

Beweis. Sei (G, $\cdot$) ein Gruppoid und $\exists e \in G$ so gilt

$$\forall a \in G : a \cdot e = e \cdot a = a$$

Angenommen, es existiere ein weiteres neutrales Element $\tilde{e}$, dann gilt:

$$\forall a \in G : a \cdot \tilde{e} = \tilde{e} \cdot a = a$$

Daher gilt $\tilde{e} = \tilde{e}e = e$. □

Definition. Sei (G, ·) eine Halbgruppe mit Einselement e, so wird G auch Monoid genannt und kann (G, ·, e) geschrieben werden.

Einige Beispiele für Monoide sind (**N**, +), (**N**, ·), (**Z**, +), (**Z**, ·), (**R**,+) und (**R**, ·).

Definition. Gegeben sei ein Gruppoid $(G, \cdot, e)$ mit Einselement. Sei $a \in G$, a' wird Inverses Element von a genannt falls

$$a' \cdot a = a \cdot a' = e$$

a' wird dann Inverses zu a genannt. Das Inverse kann auch als a^{-1} geschrieben werden. Falls die gegebene Verknüpfung ein + ist, so schreibt man das Inverse von a als $-a$.

[18]

Satz 2.1.2. *Eindeutigkeit des Inversen*
Gegeben sei ein Monoid $(G, \cdot, e)$ mit $a \in G$. Falls a ein Inverses a' besitzt, so muss dieses eindeutig bestimmt sein.

Beweis. Da a' inverse zu a ist, gilt $a \cdot a' = a' \cdot a = e$. Angenommen $\tilde{a}$ sei ein weiteres Inverses zu a, es gelte also $a \cdot \tilde{a} = \tilde{a} \cdot a = e$. Dann gilt

$$\tilde{a} = \tilde{a} \cdot e = \tilde{a} \cdot (a \cdot a') = (\tilde{a} \cdot a) \cdot a' = e \cdot a' = a'$$

Und somit gilt $\tilde{a} = a'$ □

Definition. Sei (G, ·) ein Gruppoid, dann wird (G, ·) Gruppe genannt wenn folgende Eigenschaften zutreffen

(1) Assoziativgesetz:

$$\forall a, b, c \in G : (a \cdot b) \cdot c = a \cdot (b \cdot c)$$

(2) Einselement:

$$\exists 1 \in G : \forall a \in G : 1 \cdot a = a \cdot 1 = a$$

(3) Inverse:

$$\forall a \in G : \exists a^{-1} \in G : a^{-1} \cdot a = a \cdot a^{-1} = e$$

Gilt darüber hinaus das Kommutativgesetz, also

(4)Kommutativgesetz:

$$\forall a, b \in G : a \cdot b = b \cdot a$$

Dann wird die Gruppe abelsche Gruppe genannt.

[18] Die Eigenschaften 1 bis 3 der obigen Definition werden Gruppenaxiome genannt. Sie besagen, dass eine Gruppe ein Monoid mit der weiteren Eigenschaft ist, dass jedes $a \in G$ ein Inverses besitzt.
Beispiele für Gruppen sind $(\mathbf{Z}, +)$ und $(\mathbf{R}, +)$. Diese Gruppen sind abelsche Gruppen.

Lemma 2.1.3. $\mathbf{Z}_2$ *ist eine zweielementige Gruppe mit* $\mathbf{Z}_2 := (\{0, 1\}, +)$. *Sie ist die einzige zweielementige Gruppe die exisitiert. Für ihre Verknüpfung gilt*

$0 + 0 = 0$
$1 + 0 = 1$
$0 + 1 = 1$
$1 + 1 = 0$

Die Gruppe $\mathbf{Z}_2$ beschreibt die Addition der geraden und ungeraden Zahlen, wobei 0 die Äquivalenzklasse der geraden Zahlen und 1 die Äquivalenzklasse der ungeraden Zahlen ist.

Lemma 2.1.4. *Sei* $(G, \cdot)$ *eine Gruppe so gilt für jedes* $a \in G$

$$(a^{-1})^{-1} = a.$$

Beweis. Für $(a^{-1})^{-1}$ gilt, dass es das Inverse von a^{-1} ist. Jedoch gilt auch $a \cdot a^{-1} = a^{-1} \cdot a = e$. Da a ebenfalls das Inverse von a^{-1} ist, folgt wegen der Eindeutigkeit des Inverses $a = (a^{-1})^{-1}$ □

[18]

Lemma 2.1.5. *Rechenregeln für Gruppen*
Sei $(G, \cdot)$ *eine Gruppe und* $a, b, c \in G$, *dann gelten folgende Rechenregeln.*

(i). $(a \cdot b)^{-1} = b^{-1} \cdot a^{-1}$

(ii). $c \cdot a = c \cdot b \implies a = b$

(iii). Die Gleichung $a \cdot x = b$ *hat in* G *die eindeutige Lösung*

$$x = a^{-1} \cdot b$$

Sowohl bei (ii) *also auch bei* (iii) *gelten die jeweils symmetrischen Versionen.*

Beweis. (i). Es gilt

$$(a \cdot b) \cdot (b^{-1} \cdot a^{-1}) = a \cdot (b \cdot b^{-1}) \cdot a^{-1} = a \cdot a^{-1} = e.$$

Analog folgt

$$(b^{-1} \cdot a^{-1}) \cdot (a \cdot b) = b^{-1} \cdot (a^{-1} \cdot a) \cdot b = b^{-1} \cdot b = e.$$

Wegen der Eindeutigkeit des Inversen folgt die Aussage.

(ii). Es gilt

$$\begin{gathered} c \cdot a = c \cdot b \implies \\ c^{-1} \cdot (c \cdot a) = c^{-1} \cdot (c \cdot b) \implies \\ (c^{-1} \cdot c) \cdot a = (c^{-1} \cdot c) \cdot b \implies \\ a = e \cdot a = e \cdot b = b \end{gathered}$$

(iii). Es gilt $x = a^{-1} \cdot b$ ist eine Lösung, da $a \cdot x = a \cdot (a^{-1} \cdot b) = (a \cdot a^{-1}) \cdot b = e \cdot b = b$. Sei x' eine weitere Lösung, dann wäre $a \cdot x = b = a \cdot x'$. Wegen der Kürzungsregel folgt $x = x'$

□

[18]

Definition. Untergruppe
Gegeben sei eine Gruppe $(G, \cdot, e)$. Eine Teilmenge $H \subseteq G$ wird Untergruppe von G genannt, wenn die Verknüpfung $\cdot$ die Menge $H \times H$ auf H abbildet und $(H, \cdot, e)$ ebenfalls eine Gruppe bildet.

[18]

Definition. Die Menge $\mathbb{Z}_n$ der Restklassen modulo n ist $\{0, 1, \ldots, n-1\}$, wobei $a+b = c$ genau dann, wenn c der Rest von $a + b$ bei Division durch n ist, und $ab = c$ genau dann, wenn c der Rest von $a + b$ bei Division durch n ist.

Definition. Größter gemeinsamer Teiler
Seien $m, n \in \mathbf{N}^*$ beliebig, wobei $\mathbf{N}^*$ die natürlichen Zahlen ohne 0 sind. Dann ist der größte gemeinsame Teiler wie folgt definiert.

$$\mathrm{ggT}(m, n) := \max\{t \in \mathbf{N}^* | \exists r, s \in \mathbf{N}^* : (m = rt \wedge n = st)\}$$

[15]

Definition. Sei $a \in \mathbf{Z}$ ein Teiler von $b \in \mathbf{Z}$, so schreibt man dafür $a|b$.

Definition. Teilerfremd
Seien $m, n \in \mathbf{n}^*$. m,n werden teilerfremd genannt wenn gilt $\mathrm{ggT}(m, n) = 1$

Definition. $\mathbb{Z}_n^* := \{k \in \{1, 2, \ldots, n-1\} : \mathrm{ggT}(k, n) = 1\}$. Man nennt $\mathbb{Z}_n^*$ die Einheitengruppe.

Eine Einheitengruppe beinhaltet die invertierbaren Elemente. Falls p eine Primzahl ist, dann ist $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$

Für die Multiplikation in Restklassen werden zwei Elemente der Restklasse miteinander multipliziert und anschließend mod n gerechnet.
Um die Multiplikation optisch darzustellen, kann eine Cayley-Tafel verwendet werden. Im Folgenden wird die Verknüpfungstafel für $(\mathbb{Z}_5 \backslash \{0\}, \cdot)$ gezeigt.

$\cdot$	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Handelt es sich bei der Restklasse um eine prime Einheitengruppe, so ist $\mathbb{Z}_p^*$ eine abelsche Gruppe. Es gilt, dass alle $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$ Einheitengruppen sind. [20]

Definition. Wenn sich alle Elemente einer Gruppe als Potenz eines einzigen Elements schreiben lassen, so nennt man diese Gruppe zyklisch.

[18]

Definition. Als Ring wird eine Menge A mit zwei Verknüpfungen $(+, \cdot)$ genannt die folgende Eigenschaften besitzt:

(i). A ist eine abelsche Gruppe bezüglich der Addition, mit neutralem Element 0 und additiven Inversen $-a$ für $a \in A$

(ii). Die Multiplikation ist assoziativ. Es gilt also $\forall a, b, c, \in A : a \cdot (b \cdot c) = (a \cdot b) \cdot c$

(iii). Die Distributivgesetze gelten. $\forall a, b, c \in A : (a + b) \cdot c = a \cdot c + b \cdot c$ und $c \cdot (a\ + b) = c \cdot a + c \cdot b$

(iv). Es existiert ein Einselement $1 = 1_A$ in A mit $1 \cdot a = a \cdot 1 = a\ \forall a \in A$

Gilt darüber hinaus $a \cdot b = b \cdot a\ \forall a, b \in A$ so ist der Ring kommutativ.

Definition. Sei $(A, +, \cdot)$ ein kommutativer Ring. Gibt es zu jedem $a \in A \setminus \{0\}$ ein $a^{-1} \in A$ mit $a^{-1}a = 1$, so nennt man A einen Körper.

Definition. Gegeben sei $n \in \mathbf{N}^*$. Als eulersche ϕ-Funktion bezeichnet man

$$\phi(n) := |\{m \in \mathbf{N}^* | ((0 < m < n) \wedge (\mathrm{ggT}(m, n) = 1))\}|$$

Das bedeutet, dass die eulersche ϕ-Funktion angibt, wie viele positive ganzzahlige Elemente exisitieren, die zu der gegebenen Zahl n relativ prim sind, also keinen gemeinsamen Teiler außer der Zahl 1 besitzen.

Definition. Eine Zahl $p \in \mathbf{N}^*$ mit ≥ 2 wird genau dann Primzahl genannt, wenn $\phi(p) = p - 1$ gilt.

Satz 2.1.6. *Eulersche ϕ-Funktion für Primzahlprodukte*
Seien $p, q \in \mathbf{N}^*$ mit $p \neq q$ zwei beliebige Primzahlen, dann gilt

$$\phi(pq) = (p-1)(q-1).$$

Beweis. Seien p und q zwei Primzahlen. Das Produkt pq ist nicht teilerfremd zu den Zahlen $q, 2q, 3q, ..., (p-1)q$ und $0, p, 2p, 3p, \ldots, (q-1)p$. Das heißt, sie besitzen einen größten gemeinsamen Teiler, der nicht 1 ist. Da p und q Primzahlen sind, gilt für alle anderen Zahlen x mit $0 \leq x \leq pq$, dass sie teilerfremd zu pq sind. Daher folgt

$$\phi(pq) = pq - q - (p-1) = (p-1)(q-1)$$

□

Beispiel. Sei $p = 2$ und $q = 5$. Daher ist $n = 2 \cdot 5 = 10$. Es gilt $\phi(10) = 4$, da 10 genau zu 4 Zahlen teilerfremd ist und zwar zu den Zahlen 1,3,7 und 9. Durch den Satz Eulersche ϕ-Funktion für Primzahlprodukte gilt $\phi(10) = 1 \cdot 4 = 4$

[15]

Definition. a kongruent b modulo c
Seien $a, b, c \in \mathbf{Z}$. a wird kongruent zu b modulo c genannt wenn gilt:

$$a \equiv b \mod c :\Leftrightarrow \exists d \in \mathbf{Z} : a = b + dc$$

Es gilt folgende Notationskonvention

$$a := b \mod c := min\{r \in \mathbf{N} | r \equiv b \mod c\}$$

Das bedeutet, durch die Benutzung des Gleichheitszeichens wird das b auf der rechten Seite stets durch die eindeutig bestimmte kleinste natürliche Zahl a ersetzt, die kongruent zu b modulo c ist.

Satz 2.1.7. *Gegeben seien $a_1, a_2 \in \mathbf{Z}$. Für a_2 gilt $a_2 \geq 1$ und für a_1 gilt $a_1 \neq 0$. Durch wiederholtes Dividieren mit Rest werden die Elemente $a_3, a_4, ... \in \mathbf{Z}$ gebildet.*

$$a_i = q_i a_{i+1} + a_{i+2} \text{ für } q_i \in \mathbf{Z} \text{ und } 0 \leq a_{i+2} < a_{i+1}.$$

Dadurch exisitiert ein $n \in \mathbf{N}$ mit $a_n \neq 0$ sowie $a_{n+1} = 0$. Dann gilt $a_n = \mathrm{ggT}(a_1, a_2)$.

Beweis. Es gilt $a_1 > a_3 > ... \geq 0$. n wird so gewählt, dass ein $a_n \neq 0$ und $a_{n+1} = 0$ existiert. Dadurch können die Gleichungen des euklidischen Algorithmus gebildet werden.

$$\begin{array}{llll}
a_1 & = q_1 a_2 + a_3, & q_1 \in \mathbf{Z}, & 0 < a_3 < a_2 \\
a_2 & = q_2 a_3 + a_4, & q_2 \in \mathbf{Z}, & 0 < a_4 < a_3 \\
. & . & . & . \\
. & . & . & . \\
. & . & . & . \\
a_{n-2} & = q_{n-2} a_{n-1} + a_n, & q_{n-2} \in \mathbf{Z}, & 0 < a_n < a_{n-1} \\
a_{n-1} & = q_{n-1} a_n + a_{n+1}, & q_{n-1} \in \mathbf{Z}, & 0 = a_{n+1}
\end{array}$$

Wird die Gleichungsfolge nun in verkehrter Reihenfolge durchlaufen so erhält man

$$a_n|a_{n-1} \rightarrow a_n|a_{n-2} \rightarrow ... \rightarrow a_n|a_2, a_n|a_1.$$

Daher ist a_n sowohl Teiler von a_1 als auch a_2. Nun muss noch überpruft werden, dass a_n der größte gemeinsame Teiler von a_1 und a_2 ist. Dazu sei t ein beliebiger gemeinsamer Teiler von a_1 und a_2. Wird die Gleichungskette von oben nach unten durchlaufen, so erhält man

$$t|a_1, a_2 \rightarrow t|a_3 \rightarrow ... \rightarrow t|a_{n-1} \rightarrow t|a_n.$$

Daher ist t Teiler von a_n und daher ist $a_n = \text{ggT}(a_1, a_2)$ □

[8]

Satz 2.1.8. *erweiterter euklidischer Algorithmus*

Seien $a_1, a_2 \in \mathbf{Z}$ und $a_n = \text{ggT}(a_1, a_2)$. Dann gibt es zwei Zahlen $x, y \in \mathbf{Z}$ sodass

$$a_n = xa_1 + ya_2.$$

Beweis. Für den erweiterten euklidischen Algorithmus werden dieselben Gleichungen wie bei dem euklidischen Algorithmus benötigt. Nun sollen $x_k, y_k \in \mathbf{Z}$ so konstruiert werden, dass

$$a_k = x_k a_1 + y_k a_2 \text{ für } k = 0, 1, 2, ..., n$$

Nun wählt man (x_0, y_0) :=$(1, 0)$ und $(x_1, y_1) := (0, 1)$.

Wegen der Gleichung $a_{k+1} = a_{k-1} - q_k a_k$ und der Rekursionsformel erhält man folgende Gleichung

$$(x_{k+1}, y_{k+1}) := (x_{k-1}, y_{k-1}) \text{ - } q_k(x_k, y_k).$$

Da $a_n = \text{ggT}(a_1, a_2)$ erfüllt $(x, y) := (x_n, y_n)$ die Gleichung

$$a_n = xa_1 + ya_2$$

□

[8]

Beispiel. Im Folgendem wird der euklidische Algorithmus anhand eines Beispiels demonstriert. Dabei ist $a_1 = 99$ und $a_2 = 78$.

$$\begin{aligned} 99 &= 1 \cdot 78 + 21 \\ 78 &= 3 \cdot 21 + 15 \\ 21 &= 1 \cdot 15 + 6 \\ 15 &= 2 \cdot 6 + 3 \\ 6 &= 2 \cdot 3 + 0 \end{aligned}$$

Durch Zurückrechnen ergibt sich

$$\begin{aligned} 3 &= 15 - 2 \cdot 6 \\ &= 15 - 2 \cdot (21 - 1 \cdot 15) \\ &= 3 \cdot 15 - 2 \cdot 21 \\ &= 3 \cdot (78 - 3 \cdot 21) - 2 \cdot 21 \\ &= 3 \cdot 78 - 11 \cdot 21 \\ &= 3 \cdot 78 - 11 \cdot (99 - 1 \cdot 78) \\ &= 14 \cdot 78 - 11 \cdot 99 \end{aligned}$$

Somit ergibt sich für $x = 14$ und für $y = 11$

Satz 2.1.9. *Seien* $a, b, n \in \mathbf{N}^*$. *Falls* $\mathrm{ggT}(n, a) = 1$ *und* $n|ab$ *gelten, dann gilt* $n|b$.

Beweis. Es existieren $r, s \in \mathbf{Z}^*$ mit $rn + sa = 1$. Daher gilt $b = rnb + sab$. Da n jeden Summanden der rechten Seite teilt, folgt $n|b$ □

Satz 2.1.10. *Chinesischer Restsatz*

Gegeben seien paarweise teilerfremde natürliche Zahlen $m_1, m_2, ..., m_n \in \mathbf{N}$. *Weiters seien* $a_1, a_2, ..., a_n \in \mathbf{N}$ *beliebig. Dann exisitiert genau eine Zahl* $x \in \mathbf{N}$ *für die gilt* $0 \leq x \leq \prod_{i=1}^{n} m_i$ *für die folgende n simultanen Kongruenzen gelten*

$$x \equiv a_i \mod m_i,\ 1 \leq i \leq n.$$

Beweis. Sei $m := \prod_{i=1}^{n} m_i$ und $M_i := \frac{m}{m_i}$ für $1 \leq i \leq n$. Dadurch gilt $\mathrm{ggT}(m_i, M_i)) = 1$ für $1 \leq i \leq n$. Mithilfe des euklidischen Algorithmus werden natürliche Zahlen y_i bestimmt, sodass $y_i M_i \equiv 1 \mod m_i$ gilt für $1 \leq i \leq n$. Daraus folgt, dass $a_i y_i M_i \equiv a_i \mod m_i$ für $1 \leq i \leq n$. Wegen m_i ist ein Teiler von M_j für $i \neq j$, gilt dass $a_j y_j M_j \equiv 0 \mod m_i$ für alle $i, j \in \{1, 2, ..., n\}$ mit $i \neq j$. Weiters wird x wie folgt definiert

$$x := \sum_{i=1}^{n} a_i y_i M_i \mod m$$

Daraus folgt direkt $x \equiv a_i \mod m_i$ für $1 \leq i \leq n$. Dadurch wurde die Existenz einer Lösung gezeigt. Diese Lösung ist eindeutig. Wäre sie nicht eindeutig dann gäbe es zwei Lösungen x und x' mit $0 \leq x, x' < m$. Für diese Lösungen würde folgende simultane Kongruenz gelten

$$x \equiv a_i \equiv x' \mod m_i \text{ für } 1 \leq i \leq n.$$

Aus m_i sind paarweise teilfremd folgt, dass $x \equiv x' \mod m$ und wegen $0 \leq x, x' < m$ folgt $x = x'$ □

[15]

Satz 2.1.11. *Satz von Fermat und Euler*
Seien $m, n \in \mathbf{N}^*$ *beliebige teilerfremde Zahlen, es gilt also* $\mathrm{ggT}(m, n) = 1$. *Dann gilt weiters*

$$m^{\phi(n)} \equiv 1 \mod n.$$

Beweis. Seien $x_1, \ldots, x_{\phi(n)}$ zu n teilerfremde Zahlen von 1 bis $n-1$. Dann sind $mx_1 \mod n$, $mx_2 \mod n, \ldots, mx_{\phi(n)} \mod n$ dieselben Zahlen nur anders gereiht, da $mx_i \mod n$ ebenfalls teilerfremd zu n und liegen ebenfalls zwischen 1 und n - 1. Weiters folgt durch die Anwendung der Kürzungsregel $mx_i \equiv mx_j \mod n$ ist dasselbe wie $x_i \equiv x_j \mod n$. Daraus wiederum folgt $x_i = x_j$ da $1 \leq x_i, x_j < n$.
Also ist $x_1 \cdot \cdots \cdot x_{\phi(n)} \equiv mx_1 \cdot \ldots \cdot mx_{\phi(n)} = m^{\phi(n)} \cdot x_1 \cdot \ldots \cdot x_{\phi(n)} \mod n$. Da $\mathrm{ggT}(x_i, n)=1$ für alle i und auch $\mathrm{ggT}(x_i \cdot \ldots \cdot x_{\phi(n)}, n) = 1$ kann die Kürzungsregel angewendet werden und es folgt $1 \equiv m^{\phi(n)} \mod n$ □

Beispiel. Für $m := 10$ und $n := 3$ gilt $\mathrm{ggT}(10, 3) = 1$, also ist die Voraussetzung des Satzes von Fermat und Euler erfüllt. Wegen $\phi(3) = 2$ gilt somit

$$10^2 = 100 = 33 \cdot 3 + 1 \equiv 1 \mod 3$$

[14]

Satz 2.1.12. *Folgerung aus dem Satz von Fermat und Euler*
Seien $p, q \in \mathbf{N}^*$ mit $p \neq q$ zwei beliebige Primzahlen und $m, r \in \mathbf{N}^*$. Dann gilt

$$m^{r(p-1)(q-1)+1} \equiv m \mod pq.$$

Beweis. Zu zeigen ist, dass

$$m^{r(p-1)(q-1)+1} \equiv m \mod p \text{ und } m^{r(p-1)(q-1)+1} \equiv m \mod q$$

gilt. Wenn die Beziehung gilt, dann würde daraus folgen, dass p und q die Zahl $m^{r(p-1)(q-1)+1} - m$ teilt. Da sowohl p als auch q Primzahlen sind, teilt auch ihr Produkt die Zahl $m^{r(p-1)(q-1)+1} - m$. Daraus folgt dann die Behauptung. Aufgrund der Symmetrie genügt es die Beziehung

$$m^{r(p-1)(q-1)+1} \equiv m \mod p$$

zu zeigen. Der Satz von Fermat und Euler liefert wegen $\phi(p) = p - 1$ und falls $\mathrm{ggT}(m, p)$ $m^{r(p-1)(q-1)+1} = (m^{p-1})^{r(q-1)} \cdot m \equiv 1^{r(q-1)} \cdot m = m \mod p$.
Falls $\mathrm{ggT}(m, p) \neq 1$, gilt, dass m durch p teilbar sein muss, da $p \in \mathbf{P}$. Daraus folgt, dass p jede Potenz von m teilt und damit auch $m^{r(p-1)(q-1)+1} - m$. Daraus folgt wiederum

$$m^{r(p-1)(q-1)+1} \equiv m \mod p.$$

Was zu zeigen war.

□

[15]

Beispiel. Sei $p := 2, q := 3, m := 5$ und $r := 4$. Dann gilt

$$5^{4 \cdot 1 \cdot 2 + 1} = 5^9 = 25 \cdot 25 \cdot 25 \cdot 25 \cdot 5 \equiv 1 \cdot 1 \cdot 1 \cdot 1 \cdot 5 = 5 \mod 6$$

Definition. Sei g ein Element aus $\mathbb{Z}_p^*$. Lassen sich alle anderen Elemente aus $\mathbb{Z}_p^*$ als Potenz von g schreiben, so wird g als Primitivwurzel bezeichnet.

Anders ausgedrückt, eine Primitivwurzel ist eine Basis, die alle möglichen Reste $\mod p$ erzeugen kann, wenn sie potenziert wird.
[5]

Lemma 2.1.13. *Sei $n \in \mathbb{N}^+$, dann gilt $\sum_{d|n} \phi(d) = n$.*

Das bedeutet, dass die Summe der Eulerschen Φ-Funktion für alle Teiler von n mit der Anzahl der Teiler von n übereinstimmt.

Beweis. Gegeben seien mehrere Mengen die definiert sind durch
$M_d = \{x \in \{1, 2, \ldots, n\} \operatorname{ggT}(x, n) = d\}$. Die Mengen M_d sind paarweise disjunkt, da die größten gemeinsamen Teiler verschiedener Zahlen disjunkt sind.

Aus $M_d \neq \emptyset$ folgt $d|n$ und $\bigcup_{d|n} M_d = \{1, 2, \ldots, n\}$.
Da es $\phi(\frac{n}{d})$ ganze Zahlen gibt, die kleiner oder gleich $\frac{n}{d}$ sind und teilerfremd zu $\frac{n}{d}$ und somit auch zu n, ist Anzahl der Elemente der Menge M_d ist $\phi(\frac{n}{d})$.

Daher ist $\sum_{d|n} \phi(\frac{n}{d})$ die Anzahl der Elemente in M_d. Da die Abbildung $d \mapsto \frac{n}{d}$ eine bijektive Abbildung auf der Menge der Teiler von n ist, gilt $n = \sum_{d|n} \phi(d)$ □

[17]

Definition. Ist eine Gruppe endlich, so wird die Anzahl der Elemente der Gruppe Ordnung genannt und mit ord abgekürzt.

Definition. Für die Ordnung von k modulo p ist muss gelten

$$\operatorname{ord}_p(k) := min\{j \in \mathbb{N} : k^j = 1 \mod p\}$$

Es ist $\operatorname{ord}_p(k) = u$ genau dann, wenn die Ordnung der von k erzeugten Untergruppe von $\mathbb{Z}_p$, also von $\{k^j \mod p : j \in \mathbf{N}\}$, gleich u ist.

Ist die Ordnung von k gleich $p - 1$, dann ist $k \mod p$ eine Primitivwurzel. [17]

Satz 2.1.14. *Sei K ein Körper und f ein Polynom mit Grad n für welches gilt $f \in K[x]$ und $f \neq 0$, dann besitzt f höchstens n Nullstellen.*

Beweis. Mithilfe vollständiger Induktion soll der Satz bewiesen werden. Für $n = 0$ gilt die Aussage, da $f \in K$ und $f \neq 0$. Sei $n > 0$. Besitzt f keine Nullstelle so ist die Aussage wahr. Besitzt f jedoch eine Nullstelle a, so gilt, dass $f = (x - a)q$ wobei der Grad von $q = n - 1$. Aufgrund der Induktionsvorraussetzung besitzt q höchstens $n - 1$ Nullstellen. Da K keine Nullteiler enthält hat f daher höchstens n Nullstellen. □

[8]

Satz 2.1.15. *Sei p eine Primzahl, $d \in \mathbb{N}$ mit $d|n$ und h ein Polynom für das gilt:*

$$h : \mathbb{Z}_p \to \mathbb{Z}_p$$
$$x \mapsto h(x) := x^d - 1.$$

Dann besitzt h genau d Nullstellen in $\mathbb{Z}_p$.

Beweis. Wähle $k \in \mathbb{N}$ so, dass $dk = p - 1$. Weiters wird $f(x)$ definiert als

$$f(x) := \sum_{l=0}^{k-1} (x^d)^l$$

$f(x)$ ist also die Summe der k Potenzen von x^d. $g(x) := h(x)f(x) = x^{p-1} - 1$. Offensichtlich hat f den Grad $kd - d$ und kann daher höchstens $kd - d = p - 1 - d$ Nullstellen haben. Nach dem Satz von Fermat und Euler gilt $g(x) = 0$ für alle $x \in \mathbb{Z}_p^*$, also g hat $p - 1$ Nullstellen. Da f höchstens $p - 1 - d$ Nullstellen hat und in einem Körper ein Produkt nur dann 0 sein kann, wenn mindestens einer der Faktoren 0 ist, muss h mindestens d Nullstellen haben. Andererseits kann h als Polynom vom Grad d höchstens d Nullstellen haben, und somit muss h genau d Nullstellen haben.

□

[17]

Lemma 2.1.16. *Gegeben seien drei natürliche Zahlen, also $a, b, n \in \mathbf{N}$ mit Eigenschaft* $\mathrm{ggT}(\mathrm{ord}_n(a), \mathrm{ord}_n(b)) = 1$. *Es gilt*

$$\mathrm{ord}_n(ab) = \mathrm{ord}_n(a)\,\mathrm{ord}_n(b)$$

Beweis. Für jedes $k \in \mathbf{N}$ gilt $(ab)^k = a^k b^k$. Insbesondere gilt für $k = \mathrm{ord}_n(a)\,\mathrm{ord}_n(b)$, dass $(ab)^k = a^k b^k = 1$, und daher muss $\mathrm{ord}_n(ab) \leq \mathrm{ord}_n(a)\,\mathrm{ord}_n(b)$ gelten.
Wenn für $(ab)^k = 1$, dann ist $a^k = b^{-k}$ und daher gilt sowohl $\mathrm{ord}_n(a^k) | \,\mathrm{ord}_n(a)$ als auch $\mathrm{ord}_n(a^k) = \mathrm{ord}_n(b^{-k}) | \,\mathrm{ord}_n(b)$.
Da $\mathrm{ggT}(\mathrm{ord}_n(a), \mathrm{ord}_n(b)) = 1$ muss $\mathrm{ord}_n(a^k) = 1$ gelten. Daraus folgt $a^k = 1$, woraus wiederum folgt, dass $\mathrm{ord}_n(a)|k$. Es gilt ebenfalls $b^k = 1$, und daher $\mathrm{ord}_n(b)|k$.
Weil $\mathrm{ggT}(\mathrm{ord}_n(a), \mathrm{ord}_n(a)) = 1$ gilt $\mathrm{ord}_n(a)\,\mathrm{ord}_n(b)|k$. Deshalb ist $\mathrm{ord}_n(ab) = \mathrm{ord}_n(a)\,\mathrm{ord}_n(b)$.

□

Satz 2.1.17. *Für jedes $p \in \mathbf{P} > 2$ existiert mindestens eine Primitivwurzel $k \mod p$. Außerdem gilt* $\mathrm{ord}_p(k) = \phi(p) = p - 1$.

Beweis. Gegeben sei $p \in \mathbf{P}$. q wird so gewählt, dass es ein Primfaktor von $p - 1$. Weiters wird $k \in \mathbf{N}$ so gewählt, dass $q^k | p - 1$. Das Polynom $g(x) := x^{q^k} - 1$ besitzt genau q^k mod p Nullstellen in der Menge $\{1, ..., p - 1\}$.

Sei a_q eine dieser Nullstellen, dann gilt

$$\begin{aligned} a_q^{q^k} - 1 \mod p &= 0 \\ a_q^{q^k} \mod p &= 1 \end{aligned}$$

Daher folgt, dass $\text{ord}(a_q)|q^k$.

Wäre $\text{ord}(a_q) < q^k$, so müsste a_q eine Nullstelle von $x^{q^{k-1}} - 1$ sein. Es gilt, dass genau q^{k-1} solcher Nullstellen existieren. Daher kann es maximal q^{k-1} von solchen Nullstellen a_q geben, die zusätzlich $\text{ord}_p(a_q) = q^j$ mit $j < k$ erfüllen. Daher gibt es $q^k - q^{k-1}$ Zahlen $a_q \in \{1, ..., p-1\}$, sodass gilt

$$\text{ord}_p(a_q)|q^k \text{ und } \text{ord}_p(a_q) \nmid q^j \quad \forall j < k.$$

Da q eine Primzahl ist, folgt $q^k = \text{ord}(a_q)$.
$P(p-1)$ ist die Menge der Primteiler von $p-1$ und $k_q := \max\{k \in \mathbb{N} : q^k | p-1\}$. Daher ist

$$p - 1 = \prod_{q \in P(p-1)} q^{k_q}$$

Für $q \in P(p-1)$ wird a_q so gewählt, dass $\text{ord}(a_q) = q^{k_q}$. Setze $\alpha := \prod_{q \in P(p-1)} a_q$.
Für q_1 und q_2 g $\in \mathbb{P}(p-1)$ mit $q_1 \neq q_2$ gilt, dass

$$\text{ggT}(\text{ord}(a_{q_1}), \text{ord}(a_{q_2})) = \text{ggT}(q_1^{a_{q_1}}, q_2^{a_{q_2}}) = 1$$

daraus folgt

$$\text{ord}(a_{q_1} a_{q_2}) = \text{ord}(a_{q_1}) \, \text{ord}(a_{q_2}) = q_1^{k_{q_1}} q_2^{k_{q_2}}$$

daher gilt

$$\begin{aligned} \text{ord}(\alpha) = \text{ord}_p \left(\prod_{q \in P(p-1)} a_q \right) &= \prod_{q \in \mathbb{P}(p-1)} \text{ord}(a_q) \\ &= \prod_{q \in \mathbb{P}(p-1)} q^k \\ &= p - 1 \\ &= \phi(p) \end{aligned}$$

□

Somit wurde gezeigt, dass für jede ungerade Primzahl eine Primitivwurzel existiert. [17]

Satz 2.1.18. *Eine prime Restklassengruppe* $\mod n$ *ist genau dann zyklisch wenn* $n \in \{2, 4, p^j, 2p^j\}$. *Wobei* p *eine Primzahl* > 2 *ist und* j *eine natürliche Zahl.*

Beweis. Für den Beweis ist es wichtig, dass eine prime Restklassengruppe dann zyklisch ist, wenn sie eine Primitivwurzel besitzt. Daher werden nun im folgenden alle Fälle einzeln untersucht.

- Fall $n = 2$
  Bei $n = 2$ gibt es die Restklassen $\{0, 1\}$. Da $1^1 \equiv 1 \mod 2$ und $1^2 \equiv 1 \mod 2$, ist 1 eine Primitivwurzel $\mod 2$. Somit ist die Restklassengruppe $\mathbf{Z}_2^*$ zyklisch.

- Fall $n = 4$
  Um zu zeigen, dass $\mathbf{Z}_4^*$ ebenfalls zyklisch ist, wird dieselbe Vorgehensweise wie bei $n = 2$ angewandt. Die Restklassen sind $\{0, 1, 2, 3\}$. Die Potenzen von 3 $\mod 4$ sind $3^1 \mod 4 \equiv 3$, $3^2 \mod 4 \equiv 1$, $3^3 \mod 4 \equiv 3$. Daher ist 3 eine Primitivwurzel $\mod 4$ und somit ist $\mathbf{Z}_4^*$ zyklisch.

- Fall p^j für $p > 2$
  Sei p nun eine ungerade Primzahl und j eine natürliche Zahl. Dann existiert eine Primitivwurzel $g \mod p$. g kann so gewählt werden, dass $g^{p-1} \not\equiv 1 \mod p^2$. Wäre nämlich dies der Fall, dann würde $g' = g + p$ gewählt werden und es wäre

  $$(g')^{p-1} = (g+p)^{p-1} = g^{p-1} + (p-1)g^{p-2}p + \sum_{i=2}^{p-1} \begin{pmatrix} p-1 \\ i \end{pmatrix} g^{p-1-i}p^i$$

  Zu beachten ist, dass jeder Summand der Summe durch p^2 teilbar ist. Daraus folgt also

  $$(g')^{p-1} \equiv g^{p-1} + (p-1)g^{p-2}p \equiv 1 + p^2g^{p-2} - g^{p-2}p \equiv 1 - g^{p-2}p \mod p^2$$

  Da p nicht Teiler von g^{p-2} ist, wurde gezeigt, dass $(g')^{p-1} \not\equiv 1 \mod p^2$ gilt.
  Im nächsten Schritt wird mithilfe vollständiger Induktion gezeigt, dass für $j > 1$ gilt

  $$g^{p^{j-1}(p-1)} \not\equiv 1 \mod p^j$$

  Für den Induktionsanfang wird $j = 2$ gewählt, und g wird so gewählt, dass

  $$g^{p-1} \not\equiv 1 \mod p^2$$

  Da diese Aussage nun für eine natürliche Zahl j gilt, soll gezeigt werden, dass sie auch für $j + 1$ gilt. Mithilfe des Satzes von Fermat und Euler gilt

  $$g^{p^{j-2}(p-1)} = g^{\phi(p^{j-1})} \equiv 1 \mod p^{j-1}$$

  Daraus kann geschlossen werden, dass eine Zahl $z \in \mathbf{Z}$ existiert, sodass

$$g^{p^{j-2}(p-1)} = 1 + zp^{j-1}$$

gilt. Wegen der Induktionsannahme ist p kein Teiler von z. Wenn z ein Vielfaches von p wäre, würde

$$g^{p^{j-2}(p-1)} \equiv 1 \mod p^j$$

gelten, was jedoch ein Widerspruch wäre. Wird die Gleichung $g^{p^{j-2}(p-1)} = 1 + zp^{j-1}$ nun mit p potenziert, so ergibt sich

$$g^{p^{j-1}(p-1)} \equiv 1 + pzp^{j-1} = 1 + zp^j \mod p^{j+1}$$

Da z kein Vielfaches von p ist gilt $g^{p^{j-1}(p-1)} \not\equiv 1 \mod p^{j+1}$. Somit ist die Induktion vollständig durchgeführt.
Nun soll gezeigt werden, dass g eine Primitivwurzel von p^j mit $j > 1$ ist. m wird gesetzt als $m = \text{ord}(g)$. m ist also die Ordnung von $g \mod p^j$. m muss ebenfalls Teiler von $\phi(p^j) = p^{j-1}(p-1)$ sein. Jedoch ist $g^m \equiv 1 \mod p^j$, daher muss auch $g^m \equiv 1 \mod p$ gelten. Dies bedeutet, dass $p-1$ ein Teiler von m sein muss. Es existiert also sowohl $z_1 \in \mathbb{Z}$ mit $z_1 m = p^{j-1}(p-1)$, als auch $z_2 \in \mathbb{Z}$ mit $z_2(p-1) = m$. Daher ist $z_1 z_2 (p-1) = p^{j-1}(p-1)$. Dadurch ergibt sich, dass z_1 und z_2 Teiler von p^{j-1} sind und somit Potenzen von p sind. m hat daher die Form $m = p^s(p-1)$ wobei s eine natürliche Zahl ist, für die gilt $0 \leq s \leq j-1$. Würde für s gelten $s < j-1$, dann wäre $g^{p^s(p-1)} \equiv 1 \mod p^j$ was jedoch ein Widerspruch ist, was durch vollständige Induktion gezeigt wurde. Daher ist $s = j-1$ und somit ist $m = p^{j-1}(p-1) = \phi(p^j)$. Daher ist g eine Primitivwurzel $\mod p^j$ und alle Restklassen $\mod p^j$ zyklisch.

- Fall $n = 2p^j$
  p^j und 2 sind teilerfremd. Daher gilt nach dem chinesischen Restsatz

  $$(\mathbb{Z}/2p^j\mathbb{Z}) \simeq (\mathbb{Z}/2\mathbb{Z}) \times (\mathbb{Z}/p^j\mathbb{Z}).$$

  Und somit auch

  $$(\mathbb{Z}/2p^j\mathbb{Z})^* \simeq (\mathbb{Z}/2\mathbb{Z})^* \times (\mathbb{Z}/p^j\mathbb{Z})^*.$$

  Wobei $(\mathbb{Z}/2\mathbb{Z})^* = \{1\}$ ist. Weil $(\mathbb{Z}/p^j\mathbb{Z})^*$ zyklisch ist, ist daher auch $(\mathbb{Z}/2p^j\mathbb{Z})^*$ zyklisch und es existiert eine Primitivwurzel $\mod 2p^j$.

□

[13]

Definition. Sei (G, ·) eine endliche zyklische Gruppe und g Primitivwurzel von G und $y \in G$. So wird die kleinste natürliche Zahl x mit $y = g^x \mod p$ diskreter Logarithmus von y zur Basis g genannt.

Definition nach [7]

Nach aktuellem Stand gibt es derzeit keinen Algorithmus der effizient den diskreten Logarithmus berechnen kann.
Im Zahlenraum der reellen Zahlen kann der Logarithmus leicht berechnet werden, da es sich dabei um die Umkehrung des Potenzierens handelt. Dabei wird folgende Gleichung gelöst:

$$a^x = b \iff x = log_a(b) \text{ mit } x \in \mathbb{R} \text{ und } a \in \mathbb{R}^+ \backslash \{\ 1\ \}$$

Die Berechnung des Logarithmus bei diskreten Exponentialfunktionen ist komplexer, da sich sowohl der Definitionsbereich, als auch der Wertebereich verändert. Definiert ist die diskrete Exponentialfunktion wie folgt:

Definition. Gegeben sei ein $p \in \mathbf{P}$, weiters gibt es ein $a \in \mathbf{N}$, sodass gilt $a < p$. x ist eine weitere natürliche Zahl. Die diskrete Exponentialfunktion ist wie folgt:

$$f^* : \mathbb{Z} \to \mathbb{Z}_p^* \text{ mit } f^*(x) = a^x \mod p$$

Da die diskrete Exponentialfunktion weder stetig noch monoton ist, entstehen bei der Berechnung des diskreten Logarithmus „unvorhersehbare“ Sprünge. Durch diese Sprünge ist die Berechnung des diskreten Logarithmus schwierig.
In folgender Grafik sind die Funktionswerte für die Primitivwurzel $g = 2$ und Primzahl $p = 13$ dargestellt.

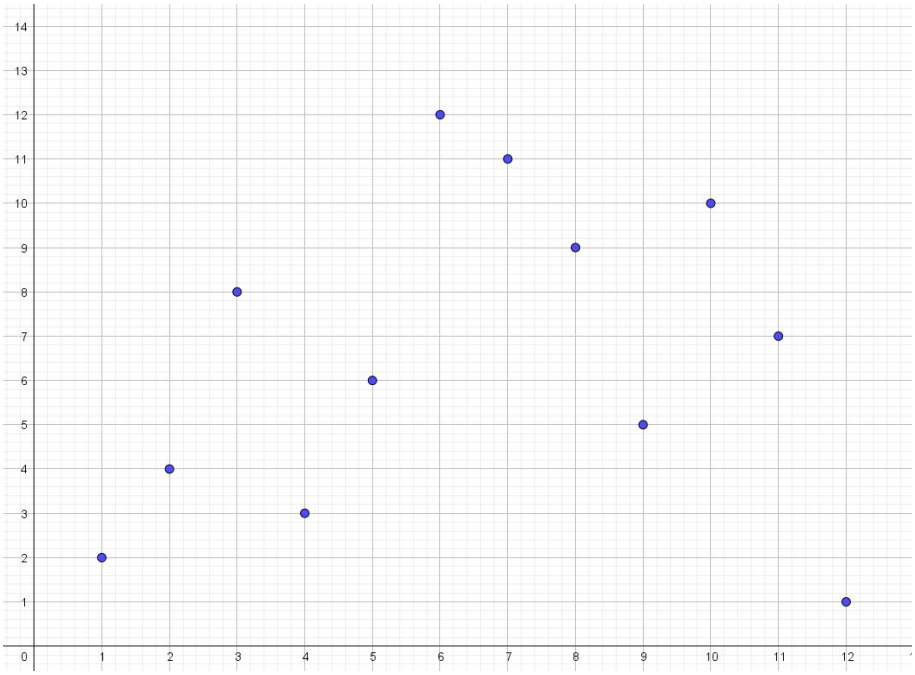


Abbildung 2.1: diskreter Logarithmus für g = 2 und p = 13

Um zu veranschaulichen, wie „willkürlich“ die Sprünge der Funktionswerte des diskreten Logarithmus sind, wurde eine zweite Grafik erstellt die die Werte der Exponentialfunktion mit $g = 5$ und Primzahl $p = 13$ abgebildet.

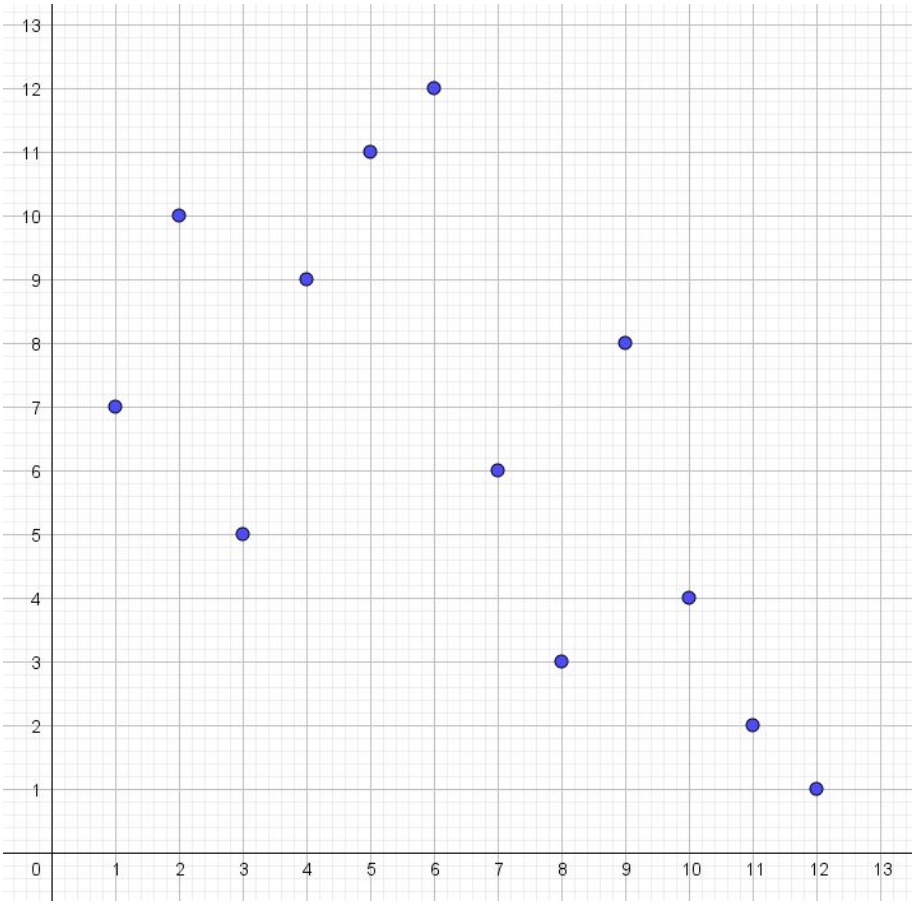


Abbildung 2.2: diskreter Logarithmus für g = 5 und p = 13

Wenn die Basis des diskreten Logarithmus keine Primitivwurzel der Restklassengruppe zur Primzahl p ist, wird die Berechnung des diskreten Logarithmus komplexer. Genau genommen ist, die Exponentialfunktion dann eingeschränkt auf $\mathbb{Z}_p^*$, nicht mehr injektiv und somit nicht mehr umkehrbar. Daher wird die Funktion zur Einwegfunktion.
[5]

Beispiel. In diesem Beispiel wurde als Primzahl $p = 13$ gewählt und als Primitivwurzel $g = 2$. Anhand der unten stehenden Tabelle kann der diskrete Logarithmus für jede Zahl $y \equiv g^a \mod p$ abgelesen werden, also $y \in \{1, 2, 3, 4, 5, 6\}$ zur Basis 2.
In der folgenden Tabelle befindet sich der diskrete Logarithmus für jede Zahl $y \equiv g^a \mod p$ also $y \in \{1, 2, 3, 4, 5, 6\}$ zur Basis 2.

$2^1 = 2 \mod 13 = 2$
$2^2 = 4 \mod 13 = 4$
$2^3 = 8 \mod 13 = 8$
$2^4 = 16 \mod 13 = 3$
$2^5 = 32 \mod 13 = 6$
$2^6 = 64 \mod 13 = 12$
$2^7 = 128 \mod 13 = 11$
$2^8 = 254 \mod 13 = 9$
$2^9 = 512 \mod 13 = 5$
$2^{10} = 1024 \mod 13 = 10$
$2^{11} = 2048 \mod 13 = 7$
$2^{12} = 4096 \mod 13 = 1$

y	2	4	8	3	6	12	11	9	5	10	7	1
$log_{13}(y)$	1	2	3	4	5	6	7	8	9	10	11	0

[8]

Definition. Unter einer **Permutation** einer Menge M versteht man eine spezielle Anordnung von M. Die Menge aller Permutationen ist S_M.

Beispiel. Die Menge M besitzt die Elemente {a,b,c}. Dann hat M sechs Permutationen: (a,b,c); (a,c,b); (b,a,c); (b,c,a); (c,a,b); (c,b,a).

Definition. Eine **Transposition** ist eine Permutation aus S_M, wobei zwei bestimmte aber verschiedene Zahlen $i, k \in \{1, 2, ..., m\}$ vertauscht werden, während alle anderen Zahlen ihren Platz behalten.

Definition. Sei A ein Alphabet, m $\in \mathbb{N}$ und $w = w_1, w_2, \dots w_m \in A^m$, wobei $w_i \in A$ für $i \in \mathbb{N}$ und $1 \leq i \leq m$ ist.
Weiter sei $P = \begin{pmatrix} 1 & ... & m \\ p_1 & ... & p_m \end{pmatrix}$ eine Permutation.
Somit ist $f_p : A^m \longrightarrow A^m$ mit $f_p(w_1...w_m) := w_{p_1} \dots w_{p_m}$ ein Transpositionsschlüssel.
Sei $u = u_1...u_k \in A^{mk}$ mit $u_i \in A^m$ für $i \in \mathbb{N}$ und $i \in \{1, \dots, k\}$ dann definiert man $f_p(u_1...u_k) = f_p(u_1)...f_p(u_k)$.

[7]

Beispiel. Gegeben sei der Klartext "BEISPIEL "sowie die Permutation $P = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$.
Die Transposition dieser Nachricht ist wie folgt:
f_p(BEISPIEL) = f_p(BEIS) f_p(PIEL) = IBSE EPLI

2.2 Zahlensysteme

Das am weitesten verbreitete Zahlensystem ist das Dezimalsystem. In der Datenverarbeitung hat jedoch das Binärsystem eine wichtige Rolle. Es gibt aber auch weitere Zahlensysteme wie zum Beispiel das Hexadezimalsystem. In diesem Kapitel soll der Aufbau dieser Zahlensysteme erklärt werden.

2.2.1 Dezimalsystem

Das Dezimalsystem beruht auf der Basis 10. Das bedeutet, dass es 10 verschiedene Ziffern gibt $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$. Bei Dezimalzahlen kann zwischen ganzen Dezimalzahlen (also $\mathbb{Z}$) und echten Dezimalzahlen unterschieden werden. Zahlen können auf verschiedene Varianten angeschrieben werden. [9]

Ganze Zahlen

Als Stellenschreibweise wird die übliche Schreibweise wie zum Beispiel 73201 verstanden. Die Zahl steht jedoch für $73201 = 7 \cdot 10000 + 3 \cdot 1000 + 2 \cdot 100 + 0 \cdot 10 + 1 \cdot 1$
Der Wert der einzelnen Ziffer hängt daher von der Stelle ab, an der sie steht. Der Wert der Ziffer nimmt von rechts nach links immer um dem Faktor $B = 10$ zu. Daher wäre die korrekte Potenzschreibweise folgende:
$73201 = 7 \cdot 10^4 + 3 \cdot 10^3 + 2 \cdot 10^2 + 0 \cdot 10^1 + 1 \cdot 10^0$
Die Potenz gibt daher die Stelle der Ziffer in der Zahl an. Allgemein ausgedrückt wird die Stellenwertschreibweise wie folgt formuliert.

Definition. Sei z eine ganze Dezimalzahl mit $n+1$ Stellen $a_0, ..., a_n \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
Dann lautet die Stellenschreibweise für $z = a_n a_{n-1} ... a_1 a_0$

Definition. Die Potenzschreibweise für ganze Dezimalzahlen lautet
$z = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + ... + a_1 \cdot 10^1 + a_0 \cdot 10^0$
Für $a_0, ..., a_n \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Die Potenzschreibweise kann auch mithilfe des Summensymbols als
$z = \sum_{i=0}^{n} a_i \cdot 10^i$ dargestellt werden.

Echte Dezimalzahlen

Auch Zahlen aus **Q** können als Stellenwert und als Potenzschreibweise angeschrieben werden. Bei der Potenzschreibweise haben Ziffern, welche rechts vom Komma stehen, negative Hochzahlen.
Beispiel. z = 0,75 $z = 7 \cdot 10^{-1} + 5 \cdot 10^{-2}$

Allgemeine Dezimalzahlen

Ganz allgemein kann jede Dezimalzahl als Summe angegeben werden. Es gilt
$z = \sum_{i=-\infty}^{n} a_i \cdot 10^i$
Wobei $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

[9]

2.2.2 Polyadische Zahlensysteme

Neben dem weitverbreiteten Dezimalsystem, existieren viele weitere Zahlensysteme. Polyadische Zahlensysteme besitzen eine andere Basis als 10. Beispiele dafür sind das Dual-System mit Basis 2, das Quinär-System mit Basis 5, das Oktal-System mit Basis 8 oder auch das Hexadezimal-System mit Basis 16. Alle polyadischen Zahlensysteme können, wie das Dezimalsystem, in Potenzschreibweise geschrieben werden. Der einzige Unterschied ist die Basis, die je nach System unterschiedlich ist. So wird das Oktaklsystem allgemein als

$$z = \sum_{i=-\infty}^{n} a_i \cdot 8^i$$

angegeben. Das Hexadezimalsystem besitzt die Basis 16. Das bedeutet, dass das verwendete Alphabet aus 16 verschiedenen Ziffern besteht. Da es aus dem üblichen Dezimalalphabet nur 10 verschiedene Ziffern gibt, wurden für das Hexadezimalsystem weitere 6 Ziffern hinzugefügt. Diese wären $0', 1', 2', 3', 4', 5'$,. Oft werden diese 6 weiteren Ziffern auch als Buchstaben bezeichnet und die Buchstaben A, B, C, D, E, F für das Hexadezimalsystem verwendet. Dieses System wird in der Datenverarbeitung eingesetzt, da bei 4 Bit genau $4^2 = 16$ Zustände angegeben werden können und dies genau einer Stelle im Hexadezimalsystem entpricht. Beim Hexadezimalsystem würde die Potenzschreibweise wie folgt lauten

$$z = \sum_{i=-\infty}^{n} a_i \cdot 16^i$$

Beispiel. $z = 641 = 6 \cdot 16^2 + 4 \cdot 16^1 + 1 \cdot 16^0$

Ganz allgemein können polyadische Zahlensystem folgendermaßen angeschrieben werden:

$$z = \sum_{i=-\infty}^{n} a_i \cdot B^i$$

Wobei B die Basis des Zahlensystems angibt und für a_i gilt $0 \leq a_i < B$. [9] [2]

2.2.3 Dual-System

Das Dual-System gehört zu den polyadischen Zahlensystemen besitzt aber einige Eigenheiten, welche es zu einem Spezialfall machen. Dieses Zahlensystem besitzt die Basis 2, das bedeutet, dass es lediglich 2 Symbole in seinem Alphabet hat. Dabei werden die Ziffern 0 und 1 genutzt. Dual-Zahlen können ebenfalls in Stellenschreibweise oder in Potenzschreibweise geschrieben werden. Die Stellenwertschreibweise einer Dual-Zahl ist

$$z = a_n a_{n-1} ... a_1 a_0, a_{-1} a_{-2} ... a_{m+1} a_m$$

Die Potenzschreibweise hingegen ist

$$z = \sum_{i=-\infty}^{n} a_i \cdot 2^i$$

Ein Vergleich von Zahlen aus dem Dual-System zu den Zahlen aus dem Dezimalsystem können der folgenden Tabelle entnommen werden.

Dual	Dezimal
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15
10000	16
10001	17
...	...

Dabei ist offensichtlich, dass die Stellenzahl bei dem Dual-System schneller zunimmt als im Dezimalsystem.

2.2.4 Binär-System

Das Binär-System und das Dual-System haben sehr viele Gemeinsamkeiten und werden oft als ein- und dasselbe Zahlensystem angesehen, jedoch gibt es Unterschiede in den beiden Systemen. Das Dual-System gehört, wie bereits erwähnt, zu den polyadischen Zahlensystemen und besitzt die Basis 2. Daher werden alle weiteren Zahlen aus den Potenzen von 2 aufgebaut. Streng genommen bedeutet jedoch Binär-System, dass es sich um ein beliebiges Zahlensystem mit zwei Zuständen handelt. Somit handelt es sich um eine Überkategorie des Dual-Systems. Der Begriff Binärsystem ist dennoch weiter verbreitet als der Begriff Dual-System. [9] [2]

2.2.5 Umrechnung Zahlensysteme

Umrechnung ins Dezimalsystem
Um polyadische Zahlensysteme, welche nicht die Basis 10 besitzen, in das Dezimalsystem umzuwandeln, wird die Stellenwertschreibweise benötigt. Denn verschiedene Zahlen in verschiedenen Zahlensystemen haben verschiedene Werte. Das Prinzip der Umrechnung ist bei jedem System das gleiche. Die Zahl wird als Stellenwertzahl angeschrieben und addiert.

Beispiel. Die Zahl 3751 mit Basis 8 soll in das Dezimalsystem umgerechnet werden.

$$3751_8 = 3 \cdot 8^3 + 7 \cdot 8^2 + 5 \cdot 8^1 + 1 \cdot 8^0$$
$$3751_8 = 2025_{10}$$

Beispiel. Im Dual-System werden hingegegen 2er Potenzen verwendet.

$$11001_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$
$$11001_2 = 25_{10}$$

Beispiel. Da es beim Hexadezimalsystem mehr Ziffern gibt, als im Dezimalsystem, müssen bei der Umrechnung zuerst alle Ziffern mit entsprechenden Ziffern des Dezimalsystem ausgetauscht werden.

Hex:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Dez:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

$$C3E_{16} = 12 \cdot 16^2 + 3 \cdot 16^1 + 14 \cdot 16^0$$
$$C3E_{16} = 3134_{10}$$

Umrechnung von Dezimalsystem
Um Zahlen vom Dezimalsystem in andere Zahlensysteme umzurechnen wird die Divisionsmethode verwendet. Dabei wird das Prinzip der fortgesetzten Division verwendet. Dabei wird die Zahl aus Basis 10 durch die Basis des Zielzahlensystems dividiert. Die jeweiligen Reste ergeben dann die umgewandelte Zahl. Dabei hat jedoch der letzte Rest den höchsten Stellenwert. [9]

Beispiel. Die Zahl 169_{10} soll in das Dual-System umgerechnet werden.

169 : 2	=	84	Rest 1
84 : 2	=	42	Rest 0
42 : 2	=	21	Rest 0
21 : 2	=	10	Rest 1
10 : 2	=	5	Rest 0
5 : 2	=	2	Rest 1
2 : 2	=	1	Rest 0
1 : 2	=	0	Rest 1

Somit gilt $169_{10} = 10101001_2$

Diese Art der Umrechnung kann für alle Umrechnungen vom Dezimalsystem in ein polyadisches Zahlensystem verwendet werden.

Beispiel. Die Zahl 169 soll diesmal in das Hexadezimalsystem, also in eine Zahl zur Basis 16, umgerechnet werden.

$$\begin{array}{lcll} 169 : 16 & = & 10 & \text{Rest } 9 \\ 10 : 16 & = & 0 & \text{Rest } 10 \end{array}$$

Da 10 im Hexadezimalensystem nicht einen anderen Wert hat, muss für den Rest 10, der Buchstabe A gewählt werden. Somit ergibt sich $169_{10} = A9_{16}$

[3]

2.2.6 Rechenoperationen im Binärsystem

Addition und Subtraktion sind im Binärsystem relativ simpel und vergleichbar mit dem Dezimalsystem. Multiplikationen und Divisionen sind dahingegen komplexer. Bei der Addition werden zwei Binärzahlen durch stellenweises Addieren mit Übertrag addiert. Dabei wird folgende Additionstabelle verwendet.

				Übertrag	$\sum$
0	+	0	=	0	0
0	+	1	=	0	1
1	+	0	=	0	1
1	+	1	=	1	0

Beispiel. Die Zahlen 25 und 7 sollen addiert werden. Zunächst werden die Zahlen ins Binärsystem umgewandelt und auf 8 Bits ergänzt. 25 = 00011001 7 = 00000111

$$\begin{array}{cccccccccc} & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ + & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ & & & & 1 & 1 & 1 & 1 & 1 & \\ \hline & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Die Subtraktion von zwei Binärzahlen kann auf die Addition zurück geführt werden indem das Zweierkomplement des Subtrahenden gebildet und danach addiert wird.

Definition. Unter Zweierkomplement wird das vollständige Komplement einer Binärzahl verstanden. Es entsteht durch das Invertieren jeder einzelnen Komponente der Zahl und dem anschließeneden Addieren von 1.

Entsteht bei der Addition ein Übertrag, so ist das Ergebnis positiv. Wert des Ergebnisses bilden die 8 Bits ohne Übertrag. Existiert kein Übertrag, so ist das Ergebnis negativ.

Beispiel. Die Zahl 26 soll von der Zahl 55 subtrahiert werden. Dazu werden die Binärentwicklungen benötigt. $A = 55_{10} = 00110111_2$
$B = 26_{10} = 00011010_2$ Als erstes wird nun B invertiert.
$B' = 11100101$ Das Zweierkomplement entsteht durch das Addieren von 1. $\bar{B} = 11100110$ Nun wird A mit dem Zweierkomplement addiert.

$$\begin{array}{rcccccccc} & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ + & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ & & & 0 & & 1 & 1 & & \\ \hline \mathbf{1} & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}$$

Da das Ergebnis einen Übertrag hat, ist es positiv.

[3]
Die Multiplikation von Bitblöcken mit beliebiger Länge n kann wie folgt konzipiert werden. Die Positionen der Bits werden von 0 bis $n-1$ nummeriert. t wird auf $t = 0...10$ gesetzt. und für i mit $0 \leq i \leq n-1$ wird t^i definiert als $t^i = t \odot t \odot ...i... \odot t$. Dies ist der Bitstring, welcher das Bit 1 genau an der Stelle i hat. Die restlichen Bits sind 0. Dabei ist $t^0 = 0...01$ Das Einselement und $t^i \odot t^j = (t \odot t...i... \odot t) \odot (t \odot t...j... \odot t) = t^{i+j}$. Für Bytes kann mithilfe der Rekursion die Formel $t^8 = t^4 \oplus t^3 \oplus t^1 \oplus t^0$ verwendet werden. [7]

2.2.7 Anwendung der Zahlensysteme in der Kryptografie

In der Kryptografie wird hauptsächlich das Binär- oder Hexadezimalsystem verwendet. Die Schlüssel bei verschiedenen Verschlüsselungsmethoden werden in Bytes gespeichert die wiederum das Binärsystem benötigen. Aber auch die zu verschlüsselnden Nachrichten werden in das Binär- oder Hexadezimalsystem umgewandelt, bevor sie verschlüsselt werden. So kann jede Art von Information in Binärcode gespeichert werden. Texte werden im ASCII-Code umgewandelt. Auch das JPEG-Format für Bilder speichert alle Information in Binär ab. Töne können als MP3-Datei gespeichert werden, welche wieder das Binär-System verwendet. [7]

ASCII-Code
Für Austausch und Speicherung von Zeichen und Zeichenketten ist der ASCII-Code der Standard. ASCII steht für American Standard Code for Information Interchange. Er war früher ein 7 bit langer Binärcode, welcher verschiedene Zeichen darstellte. Durch die Länge von 7 bit konnten 128 verschiedene Zeichen dargestellt werden. Diese Zeichen reichten aber durch die technische Entwicklung nicht mehr aus und der ASCII-Code wurde auf 8 bit (256 Zeichen) erweitert. Auch diese Variante wurde durch den Unicode abgelöst, jedoch wurde der ASCII-Code in den Unicode integriert. Der Unicode ist ein 16-bit Code, welches 65.536 Zeichen entspricht, es wurden jedoch 17 verschiedene Ebenen dieses Codes geschaffen, die jeweils 65.536 Zeichen beinhalten und somit ist nun Platz für 1.141.112 Zeichen. Diese sind jedoch noch nicht vollständig belegt. [7]

Zeichen	DEZ	HEX	OKT	BIN	Zeichen	DEZ	HEX	OKT	BIN
NUL	0	0	0	0	SPACE	32	20	40	100000
SOH	1	1	1	1	!	33	21	41	100001
STX	2	2	2	10	„	34	22	42	100010
ETX	3	3	3	11	#	35	23	43	100011
EOT	4	4	4	100	$	36	24	44	100100
ENQ	5	5	5	101	%	37	25	45	100101
ACK	6	6	6	110	&	38	26	46	100110
BEL	7	7	7	111	,	39	27	47	100111
BS	8	8	10	1000	(	40	28	50	101000
TAB	9	9	11	1001	)	41	29	51	101001
LF	10	A	12	1010	*	42	2A	52	101010
VT	11	B	13	1011	+	43	2B	53	101011
FF	12	C	14	1100	,	44	2C	54	101100
CR	13	D	15	1101	–	45	2D	55	101101
SO	14	E	16	1110	.	46	2E	56	101110
SI	15	F	17	1111	/	47	2F	57	101111
DLE	16	10	20	10000	0	48	30	60	110000
DC1	17	11	21	10001	1	49	31	61	110001
DC2	18	12	22	10010	2	50	32	62	110010
DC3	19	13	23	10011	3	51	33	63	110011
DC4	20	14	24	10100	4	52	34	64	110100
NAK	21	15	25	10101	5	53	35	65	110101
SYN	22	16	26	10110	6	54	36	66	110110
ETB	23	17	27	10111	7	55	37	67	110111
CAN	24	18	30	11000	8	56	38	70	111000
EM	25	19	31	11001	9	57	39	71	111001
SUB	26	1A	32	11010	:	58	3A	72	111010
ESC	27	1B	33	11011	;	59	3B	73	111011
FS	28	1C	34	11100	<	60	3C	74	111100
GS	29	1D	35	11101	=	61	3D	75	111101
RS	30	1E	36	11110	>	62	3E	76	111110
US	31	1F	37	11111	?	63	3F	77	111111

Abbildung 2.3: ASCII-Tabelle 7-bit

Zeichen	DEZ	HEX	OKT	BIN	Zeichen	DEZ	HEX	OKT	BIN
@	64	40	100	1000000	`	96	60	140	1100000
A	65	41	101	1000001	a	97	61	141	1100001
B	66	42	102	1000010	b	98	62	142	1100010
C	67	43	103	1000011	c	99	63	143	1100011
D	68	44	104	1000100	d	100	64	144	1100100
E	69	45	105	1000101	e	101	65	145	1100101
F	70	46	106	1000110	f	102	66	146	1100110
G	71	47	107	1000111	g	103	67	147	1100111
H	72	48	110	1001000	h	104	68	150	1101000
I	73	49	111	1001001	i	105	69	151	1101001
J	74	4A	112	1001010	j	106	6A	152	1101010
K	75	4B	113	1001011	k	107	6B	153	1101011
L	76	4C	114	1001100	l	108	6C	154	1101100
M	77	4D	115	1001101	m	109	6D	155	1101101
N	78	4E	116	1001110	n	110	6E	156	1101110
O	79	4F	117	1001111	o	111	6F	157	1101111
P	80	50	120	1010000	p	112	70	160	1110000
Q	81	51	121	1010001	q	113	71	161	1110001
R	82	52	122	1010010	r	114	72	162	1110010
S	83	53	123	1010011	s	115	73	163	1110011
T	84	54	124	1010100	t	116	74	164	1110100
U	85	55	125	1010101	u	117	75	165	1110101
V	86	56	126	1010110	v	118	76	166	1110110
W	87	57	127	1010111	w	119	77	167	1110111
X	88	58	130	1011000	x	120	78	170	1111000
Y	89	59	131	1011001	y	121	79	171	1111001
Z	90	5A	132	1011010	z	122	7A	172	1111010
[	91	5B	133	1011011	{	123	7B	173	1111011
\	92	5C	134	1011100	|	124	7C	174	1111100
]	93	5D	135	1011101	}	125	7D	175	1111101
^	94	5E	136	1011110	~	126	7E	176	1111110
_	95	5F	137	1011111	DEL	127	7F	177	1111111

Abbildung 2.4: ASCII-Tabelle 7-bit

3 Symmetrische Verschlüsselung

Bei der symmetrischen Verschlüsselung wird zur Ver- und Entschlüsselung derselbe Schlüssel verwendet. Da dieser Schlüssel geheim gehalten werden muss, damit keine außenstehenden Personen die zu verschlüsselnde Nachricht entschlüsseln können, trägt dieses Verfahren auch den Namen „private Key Verfahren“. Die ersten symmetrischen Verschlüsselungsverfahren sind schon von den Spartanern und Römern entwickelt worden, aber auch heutzutage gibt es moderne symmetrische Verfahren. Grundsätzlich kann ein symmetrisches Verschlüsselungsverfahren wie folgt definiert werden.

Definition. Als symmetrisches Verschlüsselungsverfahren, oder auch symmetrisches Kryptosystem, wird ein Tupel $(K, P, C, KeyGen, Enc, Dec)$ genannt, welches folgende Eigenschaften besitzt.

- K ist eine Menge, welche Schlüsselraum genannt wird. Alle Elemente von K werden als Schlüssel bezeichnet.
- P ist ebenfalls eine Menge. P wird Klartextraum genannt und alle Elemente von P werden Klartexte genannt.
- C ist eine Menge, die Chiffretextraum genannt wird und alle Elemente dieser Menge werden Chiffretexte beziehungsweise Schlüsseltexte genannt.
- KeyGen ist ein Algorithmus, der auf Zufällen basiert. Dieser Algorithmus wird zur Schlüsselerzeugung verwendet und liefert den Schlüssel K $\in K$.
- Enc ist der Verschlüsselungsalgorithmus der mit Eingabe des Schlüssels und Klartextes einen Chiffretext erzeugt.
- Dec ist ein deterministischer Algorithmus. Er liefert einen Klartext bei Eingabe des Schlüssels und des Chiffretextes. Er wird auch Entschlüsselungsalgorithmus genannt.
- Bei Mitgabe des Schlüssels K und eines Klartextes P erzeugt der Verschlüsselungsalgorithmus einen Chiffretext C.
- Bei Mitgabe des Schlüssels K und eines Chiffretextes C liefert der Entschlüsselungsalgorithmus den Klartext P.

[8] Klartexte und Chiffretexte bestehen aus Zeichen welche einem Alphabet entsprechen.

Definition. Als Alphabet wird eine endliche, nicht leere Menge Σ verstanden. Die Anzahl der Elemente von Σ wird als Länge bezeichnet und die Elemente von Σ als Zeichen, Buchstaben oder Symbole von Σ.

Ein sehr bekanntes Alphabet, welches in dieser Arbeit hauptsächlich verwendet wird, ist das lateinische Alphabet.

Beispiel. $\Sigma = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z\}$
Die Länge beträgt 26

In der Datenverarbeitung wird häufig das Alphabet $\mathbf{Z}_2$ verwendet, welches mit verschiedenen Zeichen dargestellt werden kann.

Beispiel. $\Sigma = \{1, 0\}$
oder
$\Sigma = \{t, f\}$

Auch der ASCII-Zeichensatz ist ein häufig genutztes Alphabet.

Definition. Sei Σ ein Alphabet, dann ist ein Wort über Σ eine endliche Zeichenfolge aus Σ. Die leere Folge ϵ ist ebenfalls ein Wort über Σ und wird das leere Wort genannt. Ein Wort kann als $\vec{w}$ dargestellt werden.

Definition. Die Länge eines Wortes $\vec{w}$ über Σ ist die Anzahl der enthaltenen Zeichen und wird mit $|\vec{w}|$ bezeichnet. Das leere Wort hat die Länge 0.

Definition. Σ^* bezeichnet die Menge aller möglichen Wörter über Σ.

Definition. Sei $n \in \mathbf{N}$, dann ist Σ^n die Menge aller Wörter mit Länge n.

[8]
Die symmetrische Verschlüsselung unterteilt sich wiederum in verschiedene Verfahren.

- **Transpositionsalgorithmus**
  Bei diesem Verfahren findet mit den Zeichen des Textes eine Permutation statt. Das bedeutet, die einzelnen Buchstaben werden vertauscht, jedoch nicht verändert. Es entsteht daher ein Anagramm.

- **Substitutionsalgorithmus**
  Bei diesem Verfahren wird jeweils ein Zeichen mit einem bestimmten anderen Zeichen ersetzt. Bei einfachen Substitutionsalgorithmen kann mithilfe der Verteilungsstatistik Annahmen darüber gemacht werden, welches Wort verschlüsselt wurde.

[4]
August Kerckhoff formulierte 1883 das Prinzip der Sicherheit bei Verschlüsselungen. „Die Sicherheit eines Kryptosystems darf nicht von der Geheimhaltung des Algorithmus abhängen. Die Sicherheit gründet sich ausschließlich auf die Geheimhaltung des Schlüssels“. [6] Dabei gab er sechs Anforderungen an, die eine Verschlüsselungsmethode benötigt.

(i). „Das System muss im Wesentlichen mathematisch unentschlüsselbar sein.

(ii). Es darf keine Geheimhaltung erfordern und es kann bequem in die Hände des Feindes fallen.

(iii). Der Schlüssel wird zur Verfügung gestellt und ist ohne die Hilfe von Notizen zu behalten und wird entsprechend gewechselt oder verändert.

(iv). Es muss für telegrafische Korrespondenz kompatibel sein.

(v). Es muss portabel sein und seine Handhabung oder Bedienung darf nicht die Unterstützung von mehreren Personen erfordern.

(vi). Schließlich ist im Falle der kontrollierten Anwendung notwendig, dass das System einfach zu bedienen ist und es weder geistige Anstrengung noch die Kenntnis einer langen Reihe von zu beachtenden Regeln erfordert . “

[6] Wobei Anforderung 2 und 3 zum heutigen Kerckhoffschem Prinzip führen.

3.1 Transpositionsverfahren

Bei Transpositionen wird die Position der zu verschlüsselnden Nachricht verändert, sodass der ursprüngliche Text nicht mehr zu erkennen ist. Falls bei der Transposition ein neuer, sinnvoller Ausdruck entsteht, wird dieser Anagramm genannt. Ein wesentliches Merkmal dieses Verfahrens ist, dass alle Zeichen enthalten bleiben.
Mathematisch greift dieses Verfahren auf die Permutation zurück, wobei eine Permutation aus mehreren Transpositionen besteht.
Alle Transpositionsverfahren können mithilfe einer Verschlüsselungsfunktion definiert werden. Die Permutation gibt dabei den Schlüsssel an.
Bekannte Vertreter dieses Verfahrens ist die Skytale von Sparta und die Verschlüsselungsschablonen nach Fleißner.

3.1.1 Skytale

Die Skytale wurde von den Spartanern im 7.Jahrhundert v. Chr. entwickelt. Das Prinzip dieser Verschlüsselung ist ziemlich elementar. Bei der Skytale handelt es sich um einen (Holz-)Stab der mit einem Band umwickelt ist. Auf dem Band befinden sich Symbole, welche die Nachricht ergeben. Das Kernelement für die Ver- und Entschlüsselung ist der Durchmesser des Stabes. [7]

Definition. Sei k ein Kreis. Eine Sehne ist ein Segment $\bar{AB}$ mit $A, B \in k$. Der **Durchmesser** ist eine Sehne, die den Mittelpunkt des Kreises k enthält.

[21]

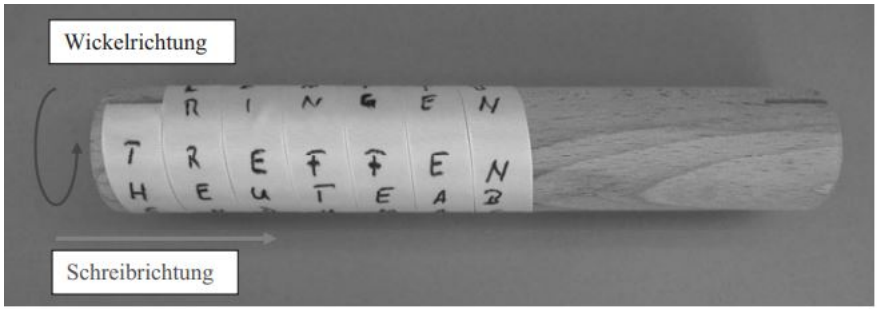


Abbildung 3.1: Abbildung einer Skytale [7]

Je nach Größe des Durchmessers ergeben sich unterschiedliche Transpositionen der Zeichen des Textes. Das Verschlüsselungsverfahren der Skytale kann auch anhand von Permutationstabellen dargestellt werden. Um diese Tabelle darzustellen, wird zunächst eine Tabelle erstellt mit m Zeilen und n Spalten. Wobei m die Anzahl der Buchstaben ist, die pro Umdrehung der Skytale auf einer Zeile steht und n sei die Anzahl der Wickelungen des Bandes um den (Holz-)Stab. Falls die letzte Wickelung nicht vollständig mit Text oder Symbolen ausgefüllt ist, wird die letzte Spalte der Tabelle ebenfalls nicht vollständig ausgefüllt. Soll der Text „Verschlüsselung gewährleistet Sicherheit“ mithilfe einer Skytale

verschlüsselt werden und wird der Durchmesser des Stabes so gewählt, dass acht Zeilen mit jeweils bis zu fünf Zeichen entstehen. So kann der Text nun in eine Tabelle mit fünf Spalten und acht Zeilen eingetragen werden. Diese Tabelle sieht wie folgt aus.

V	E	R	S	C
H	L	Ü	S	S
E	L	U	N	G
G	E	W	Ä	H
R	L	E	I	S
T	E	T	S	I
C	H	E	R	
H	E	I	T	

Um den Klartext zu verschlüsseln, wird der Text zeilenweise von links nach rechts eingetragen. Der verschlüsselte Text entsteht, wenn die Tabelle spaltenweise ausgelesen wird. Somit wäre in diesem Fall der verschlüsselte Text "VHEGRTCHELLELEHERÜUWETEISSNÄISRTCHSGHSI ". Die allgemeine Permutationstabelle für die obige Verschlüsselung ist die folgende Tabelle.

1	9	17	25	33
2	10	18	26	34
3	11	19	27	35
4	12	20	28	36
5	13	21	29	37
6	14	22	30	38
7	15	23	31	
8	16	24	32	

Dabei wurde anstatt der Buchstaben die jeweilige Buchstabennummer in die Tabelle eingetragen. Die resultierende Permutation ergibt sich durch zeilenweises Auslesen der Tabelle. In diesem Fall wäre es „1 9 17 25 33 2 10 18 26 34 3 11 19 27 35 4 12 20 28 36 5 13 21 29 37 6 14 22 30 38 7 15 23 31 8 16 24 32“

Typisch für die Syktale ist die entstehende Struktur der Permutation. Buchstaben die in der selben Zeile stehen, werden jeweils um n Stellen verschoben. Nach m Buchstaben startet die Permutation mit der Nummer der aktuellen Zeile erneut.

Beispiel. Ein weiteres Beispiel für die Skytale kann wie folgt aussehen. Der Text „Dieser Text soll verschluesselt werden“ wird verschlüsselt. Der Durchmesser wird so gewählt, dass fünf Zeilen mit jeweils bis zu sechs Zeichen entstehen. Der zu verschlüsselnde Text wird also in eine Tabelle mit fünf Zeilen und 6 Spalten eingetragen.

D	I	E	S	E	R
T	E	X	T	S	O
L	L	V	E	R	S
C	H	L	U	E	S
S	E	L	T	W	
E	R	D	E	N	

Durch spaltenweises Auslesen des Textes ergibt sich der verschlüsselte Text „DTLCSEIELHEREXVLLDSTETUTEESREWNROSS“

1	7	13	19	25	31
2	8	14	20	26	32
3	9	15	21	27	33
4	10	16	22	28	34
5	11	17	23	29	
6	12	18	24	30	

Die Permutation die sich aus dieser Tabelle ergibt ist „1 7 13 19 25 31 2 8 14 20 26 32 3 9 15 21 27 33 4 10 16 22 28 34 5 11 17 23 29 6 12 18 24 30“

[7]

3.1.2 Fleissner Schablone

Die Fleissner Schablone wurde von Eduard Fleissner 1881 erstmalig erklärt. Der ursprüngliche Name dieses Verfahrens war „Neue Patronenschrift“. Fleissner beschreibt dieses Verfahren wie folgt.

„Das Prinzip dieser Geheimschrift beruht auf durchlöcherten Patronen, durch welche man einfach Buchstaben oder Ziffern schreibt. Da die Löcher in der Patrone in verschiedenen Entfernungen angebracht sind, so werden die Worte oder Zahlen der Klarschrift aus ihrem Zusammenhang gebracht und nur wieder mit der Hilfe der Patrone lesbar. Die Patrone wird während des Schreibens der geheimen Depesche mehrmals auf demselben Raume gedreht, wobei die Löcher derselben dem Schreiber stets die Stellen anweisen, wohin die Buchstaben oder Ziffern der Worte oder Zahlen zu setzen sind. Die Patronen sind so construiert, dass bei diesen Drehungen ein Loch nie auf eine bereits beschriebene Stelle zu stehen kommt. Schliesslich erscheint die Schrift in regelmässiger Figur, aber unlesbar und nur zu entziffern vom Besitzer der gleichen Patrone.“ [19]

Dabei ist mit „Patrone“ ein Raster in quadratischer Form gemeint, bei dem bestimmte Zellen ausgestanzt sind. Die Löcher müssen so gewählt werden, dass bei jeder Drehung neue Felder sichtbar werden und kein Feld zweimal auftaucht. Da in der heutigen Zeit dieser Raster als Schablone bezeichnet wird, erhält dieses Verfahren seinen Namen. [7]

Um die Sicherheit und Komplexität der Schablone zu erhöhen, wird die quadratische Schablone in zusätzliche Quadrate oder Dreiecke unterteilt. Durch die Unterteilung entstehen mehrere kleinere Quadrate oder Bereiche, in denen verschiedene Transformationen oder Operationen angewendet werden können. Jeder Teilbereich kann unterschiedliche Schreib- und Lesemuster besitzen wie der Klartext geschrieben beziehungsweise die verschlüsselte Nachricht gelesen werden soll. Um die Komplexität dieser Verschlüsselung zu steigern, kann eine bestimmte Reihenfolge vorgegeben werden, wie die Unterteilungen der Schablone auf den Text anzuwenden sind.

Um einen Klartext mithilfe der Fleissnerschablone zu chiffrieren, wird die Schablone auf ein leeres Feld gelegt. In die Löcher der Schablone wird die Nachricht geschrieben. Sobald alle Felder voll sind, wird das Raster um 90° im mathematisch positiven Sinn gedreht. Durch die Drehung entstehen neue Felder, in die der Klartext weiter hineingeschrieben werden kann. Dieser Vorgang muss zwei weitere Male wiederholt werden. Im Anschluss wird die Schablone entfernt und es entsteht ein Quadrat mit Buchstaben.

Um die Nachricht zu dechiffieren benötigen Sender und Empfänger dasselbe Raster. Weiters muss zunächst vereinbart werden, wie die Schablone als erstes aufgelegt werden muss. Verfügt der Empfänger über diese Informationen, kann er durch einfaches Ablesen, die verschlüsselte Nachricht entschlüsseln.

Beim Verfahren der Fleissner-Schablone handelt es sich um einen algorithmischen Vorgang, der sich das Prinzip der Permutation zu nutzen macht.
Laut Fleissner bietet die Verschlüsselung mithilfe der Fleissner-Schablone einige Vorteile:

(i). „Sie eignet sich für jede Sprache und jedes Schriftzeichen.

(ii). Sie kann optisch und auch telegrafisch befördert werden.

(iii). Sie ist eine Buchstabenchiffre, die sich vorzüglich für untergeordnete militärische, staatspolizeiliche und kommerzielle Zwecke, sowie für den privaten Gebrauch eignet.

(iv). Sie bietet keine Schwierigkeit und jeder, der des Lesens und Schreibens mächtig ist, kann sie in kürzester Zeit erlernen.

(v). Sie ist außerordentlich sicher, wie es nur die besten Chiffrenmethoden sind.

(vi). Billionen von Korrespondenten können jeder mit einem anderen Schlüssel beteiligt werden.

(vii). Mit jedem der zugehörigen Schlüssel kann die Geheimschrift ungemein leicht, beinahe fließend gelesen werden.

(viii). Das System dieser Geheimschrift erlaubt unzählige Abänderungen und kann auf die mannigfaltigste Art verkompliziert werden.“

[7]

Diese Vorteile sind heutzutage schon etwas veraltet. Aus heutiger Sicht, trifft der erste Vorteil, dass die Schablone in jeder Sprache verwendet werden kann, nach wie vor zu. Der zweite Punkt kann durch die elektronische Weitergabe ergänzt werden, wenn die Buchstaben in Binärcodierung angegeben sind. Die Vorteile 4 und 7 ist heutzutage nur mehr gültig um Lernenden eine Einführung in die Kryptografie zu geben. Beispielsweise können SchülerInnen diese Geheimschrift sehr rasch erlernen und erproben. Der Vorteil 3, welcher sich auf die Sicherheit bezieht, ist schon seit etwa 1916 nicht mehr gültig. Texte, welche mit der Fleissnerschablone verschlüsselt wurden, sind seit dem 20. Jahrhundert leicht zu entschlüsseln. [7]

3.2 Substitutionsverfahren

Bei Substitutionsverfahren werden Schriftzeichen der zur verschlüsselnden Nachricht durch andere Schriftzeichen ersetzt. Das Ersetzten der Zeichen erfolgt dabei immer nach einem gewissen Schema.
Die Substitutionsverfahren werden in drei unterschiedliche Bereiche eingeteilt.

- **Monoalphabetische Verschlüsselung**
  Bei Nachrichten, die mit einer Monoalphabetischen Verschlüsselung in Geheimtexte umgewandelt werden, wird jeder Buchstabe aus dem Alphabet A jeweils mit genau einem anderen Buchstaben ersetzt. Daraus folgt, dass es jeweils zu jedem Klartext immer genau einen bestimmten Geheimtext gibt.

- **Homophone Verschlüsselung**
  Bei homophonen Verfahren werden häufige Zeichen oder Zeichenfolgen einer Sprache mit verschiedenen Zeichen bzw. Zeichenfolgen ersetzt. Daher entstehen verschiedene Möglichkeiten der Verschlüsselung.

- **Polyalphabetische Verschlüsselung**
  Bei Polyaphabetischen Verfahren gibt es verschiedene Geheimtextalphabete und einem Buchstaben des Klartextes wird ein Zeichen aus einem der Geheimtextalphabete zugeordnet.

- **Monographische, bigrafische und polygrafische Verschlüsselungen**
  Bei einer Mongraphischen Verschlüsselung wird ein Einzelzeichen des Klartextes ersetzt. Bei einer bigraphischen Verschlüsselung werden zwei Zeichen, ein Bigramm, des Klartextes ersetzt und bei einer polygrafischen Verschlüsselung werden mehrere Zeichen, ein Polygramm, ersetzt.

Um die Sicherheit von Substitutionsverfahren zu erhöhen, können verschiedene Methoden gemischt verwendet werden. Auch kann ein Transpositions- mit einem Substitutionsverfahren kombiniert werden. Verfahren aus der heutigen Zeit, welche solch eine Kombination verwenden, sind zum Beispiel der DES oder AES.

Monoalphabetische und monografische Substitution

Die monoalphabetische bzw. monografische Substitution beziehen sich auf dieselbe Art der Substitution. Der Begriff monografische Substitution wird als allgemeinere Bezeichnung verwendet, um jede Art der Buchstabenersetzung im Text zu beschreiben. Wobei hingegen der Begriff monoalphabetische Substitution verwendet wird, um zu betonen, dass jeder Buchstabe des Klartextes durch genau einen anderen Buchstaben ersetzt wird und die Zuordnung gleich bleibt.

Definition. Für die monografische Substitution werden zwei Alphabete A, B benötigt. A^* bezeichnet die Menge der Wörter über A und B^* bezeichnet die Menge der Wörter über B. Sei $w \in A^*$ mit $w = a_1 \dots a_n$, wobei $a_1, \dots, a_n \in A$. Weiters sei $w' \in B^*$. Für

die Verschlüsselung wird eine Funktion $g : A \longrightarrow B$ benötigt. Mithilfe dieser Funktion wird nun die Substitution durchgeführt sodass $w' = g(a_1) \dots g(a_n)$.

Zu beachten ist, dass für das Nullwort $\varnothing$ $g(\varnothing) = \varnothing$ gilt.

Der bekannteste Vertreter dieser Verschlüsselung ist die Cäsar-Verschlüsselung. [7]

Homophone Verschlüsselungen

Homophone Verschlüsselungen bilden einen Sonderfall der Monographischen bzw. monoalphabetischen Verschlüsselung. Sie lassen sich ebenfalls mit einer Verschlüsselungsfunktion darstellen.

Definition. Seien A, B Alphabete, wobei $A = a_1, ..., a_n$. Die Anzahl der Elemente aus B muss mindestens der Anzahl der Elemente aus A entsprechen.
Die homophone Substitution ist eine Abbildung $f : A \to B^*$. Sei $w = a_1 \dots a_n \in A$ und $w' \in B$. Dann ist $w' = f(a_1) \dots f(a_n)$. Für das Nullwort $\varnothing$ gilt $f(\varnothing) = \varnothing$.

Polyalphabetische Substitution

Die Polyalphabetische Substitution wird wie folgt definiert:

Definition. Seien $A, B_1, \dots, B_r$ Alphabete mit $r \in \mathbb{N}$ und $r > 1$
$g_j : A \to B_j$ ist eine Funktion, die jeden Buchstaben oder jedes Zeichen $a \in A$ auf ein entsprechenden Buchstaben oder ein Zeichen $b \in B_j$ abbildet. Dabei ist j die Indexnummer des spezifischen Alphabets B_j.

Die Funktion $h : \mathbf{N}^* \to \{1, 2, \dots, r\}$ ordnet jedem Buchstaben im Klartext den Index des spezifischen Alphabets, mit dem dieser Buchstabe verschlüsselt werden soll, zu. Da jede Funktion g_j injektiv ist, gibt es keine zwei verschiedenen i_1 und i_2 für die $h(i_1) = h(i_2)$ gilt.
Durch die Funktion h wird sichergestellt, dass Buchstaben im Klartext nicht vorhersehbar einem bestimmten spezifischen Alphabet zugeordnet werden können.

Sei nun $w = a_1 \dots a_n \in A$ und $w' \in (B_1, \dots B_r)*$, dann ist $w' = g_{h_{(1)}}(a_1) \dots g_{h_{(m)}}(a_m)$

Ein bekannter Vertreter dieser Substitution ist die Vigenère-Verschlüsselung.

3.2.1 Cäsar-Verschlüsselung

Die Cäsar-Verschlüsselung, oder auch Cäsar-Chiffre, wurde von Julius Cäsar bereits in der Antike entwickelt. Sie zählt zu den einfachsten und ältesten Verschlüsselungstechniken der Welt. Diese Verschlüsselung wurde von Julius Cäsar verwendet, um geheime militärische Botschaften zu verschlüsseln und so die Kommunikation mit seinen Feldherren zu sichern. Obwohl dieses Verfahren einfach ist, erwies es sich für die meisten seiner Zeitgenossen

als ausreichend sicher. Die Verschlüsselungsmethode gewährte jedoch nur eine geringe Sicherheit, da die Verschiebungszahl leicht erraten werden konnte.

Abbildung 3.2: Chiffrierscheibe für das Cäsar-Verfahren [7]

Die Cäsar-Verschlüsselung ist anfällig für Brute-Force-Angriffe, da es nur 25 mögliche Verschiebungswerte gibt. Unter einem Brute-Force-Angriff versteht man das Durchprobieren aller Möglichkeiten. Es besteht auch die Möglichkeit, die Häufigkeitsanalyse anzuwenden, um den Schlüsselwert durch die Analyse von Buchstabenhäufigkeiten in der verschlüsselten Nachricht zu erraten. [7]
Die Häufigkeitsverteilung der Buchstaben der deutschen Sprache sieht wie folgt aus.

Buchstabe	a	b	c	d	e	f	g	h	i
Häufigkeit in %	6,51	1,89	3,06	5,08	17,4	1,66	3,01	4,76	7,55
Buchstabe	j	k	l	m	n	o	p	q	r
Häufigkeit in %	0,27	1,21	3,44	2,53	9,78	2,51	0,79	0,02	7
Buchstabe	s	t	u	v	w	x	y	z	
Häufigkeit in %	7,27	6,15	4,35	0,67	1,89	0,03	0,04	1,13	

[12]
Die Cäsar-Verschlüsselung basiert auf einem einfachen Verschiebeverfahren, bei dem jeder Buchstabe im Klartext durch einen bestimmten Buchstaben ersetzt wird. Dabei gibt der Verschiebungsschlüssel an, um wie viele Buchstaben der Klartext verschoben wird. Die Verschiebung kann sowohl nach rechts, als auch nach links erfolgen.
Verschiebechiffren wie die Cäsar-Verschlüsselung werden anhand des folgenden Beispiels erklärt.

Definition. Sei $A = \{a_1, \ldots, a_n\}$ ein Alphabet und $s \in \mathbb{N}$ mit $1 \leq s \leq n$. Sei $f : A \longrightarrow A$. Die Funktion ersetzt a_i durch a_j wobei $j = i + s \mod n$.

[7]

Die Cäsar-Verschlüsselung kann mit der fundamentalen Idee der Zahl verbunden werden. Dazu wird jeder Buchstabe des Alphabets durchnummeriert. Aus a wird 1, aus b wird 2, etc. Für den Geheimtext wird nun jede Zahl mit dem Verschiebungsschlüssel addiert. Anschließend muss das Ergebnis noch modulo 26 gerechnet werden, damit sich der Geheimtext im selben Zahlenbereich befindet wie der Klartext. Nun kann jede Zahl wieder in Buchstaben umgewandelt werden.
[7]

Beispiel. Im folgenden Beispiel wird das Wort „Verschlüsselung“ verschlüsselt. Als Schlüssel wird 4 gewählt.

Alphabet	a	b	c	d	e	f	g	h	i	j	k	l	m
Nummerierung	1	2	3	4	5	6	7	8	9	10	11	12	13
Alphabet	n	o	p	q	r	s	t	u	v	w	x	y	z
Nummerierung	14	15	16	17	18	19	20	21	22	23	24	25	26

Klartext	v	e	r	s	c	h	l	u	e	s	s	e	l	u	n	g
Nummerierung	22	5	18	19	3	8	12	21	5	19	19	5	12	21	14	7
Verschiebung	26	9	22	23	7	12	16	25	9	23	23	9	16	25	18	11
Geheimtext	z	i	v	w	g	l	p	y	i	w	w	i	p	y	r	k

Der Geheimtext ist daher zivwglpyiwwipyrk.

Beispiel. Ein weiteres Beispiel für die Cäsar-Verschlüsselung. Das Wort „Substitution“ soll verschlüsselt werden. Als Verschiebungsschlüssel wird 10 gewählt.

Alphabet	a	b	c	d	e	f	g	h	i	j	k	l	m
Nummerierung	1	2	3	4	5	6	7	8	9	10	11	12	13
Alphabet	n	o	p	q	r	s	t	u	v	w	x	y	z
Nummerierung	14	15	16	17	18	19	20	21	22	23	24	25	26

Klartext	s	u	b	s	t	i	t	u	t	i	o	n
Nummerierung	19	21	2	19	20	9	20	21	20	9	15	14
Verschiebung	3	5	12	3	4	19	4	5	4	19	25	24
Geheimtext	c	e	l	c	d	s	d	e	d	s	y	x

Somit ist der verschlüsselte Text „celcdsdedsyx“

3.2.2 Vigenère-Verschlüsselung

Die Vigenère-Verschlüsselung zählt zu den polyalphabetischen Verschlüsselungen und ist einer der bekanntesten Vertreter dieser Art. Sie wurde im 16. Jahrhundert von Blaise de Vigenère entwickelt, wodurch sie auch ihren Namen erlangte. Da sie sich auf das

Prinzip der polyalphabetischen Substitution bezieht, werden mehrere alphabetische Verschiebungsoperationen gleichzeitig angewendet. Dabei wird über die gesamte Länge des Klartextes ein Schlüsselwort, bzw. ein Schlüssel, wiederholt geschrieben. Als Nächstes wird jeder Buchstabe des Klartextes durch einen Buchstaben des Schlüsselworts verschoben. Die Verschiebung beruht auf der Cäsar-Verschlüsselung, wobei aber jeder Buchstabe eine unterschiedliche Verschiebung erhält.
Die Vigenère-Verschlüsselung gilt heutzutage als unsicher, ist jedoch ein wichtiges historisches Beispiel für polyalphabtische Verschlüsselungen.

Die Vigenère-Verschlüsselung kann wie folgt definiert werden.

Beispiel. Erneut soll das Wort „Verschlüsselung“ verschlüsselt werden. Als Schlüssel wird das Wort „Vigenere“ gewählt. Anhand des Vigenère-Tableus kann nun das Wort „Verschlüsselung“ Buchstabe für Buchstabe verschlüsselt werden.
Der Vorgang kann ebenfalls tabellarisch dargestellt werden.

Klartext	V	E	R	S	C	H	L	U	E	S	S	E	L	U	N	G
Schlüsselwort	V	I	G	E	N	E	R	E	V	I	G	E	N	E	R	E
Klartext in Ziffern	22	5	18	19	3	8	12	21	5	19	19	5	12	21	14	7
Verschiebungszahl	21	8	6	4	13	4	17	4	21	8	6	4	13	4	17	4
addiert	43	13	24	23	16	12	29	25	26	27	25	9	25	25	31	11
mod 26	17	13	24	23	16	12	3	25	0	1	25	9	25	25	5	11
Geheimtext	R	N	Y	X	Q	M	D	Z	A	B	Z	J	Z	Z	F	L

Das Beispiel kann auch anhand der Definition wie folgt geschrieben werden.

Beispiel. A = A, ..., Z ist das Klartextalphabet. Mit dem Schlüsselwort (VIGENERE) ergibt sich S zu S = 17, 13, 24, 23, 16, 12, 3, 25, 0, 1, 25, 9, 25, 25, 5, 11 und f zu
$f(VERSCHLUESSELUNG) = g_{21}(V)g_8(E)g_6(R)g_4(s)g_{13}(C)g_4(H)g_{17}(L)g_4(U)g_{21}(E)g_8(S)g_6(S)$
$g_4(E)g_{13}(L)g_4(U)g_{17}(N)g_4(G) = RNYXQMDZABZJZZFL$

Beispiel. In diesem Beispiel wird ebenfalls das Wort „Substitution“ verschlüsselt. Als Schlüssel wird das Wort „Text“ gewählt. Der Vorgang kann ebenfalls tabellarisch dargestellt werden.

Klartext	S	U	B	S	T	I	T	U	T	I	O	N
Schlüsselwort	T	E	X	T	T	E	X	T	T	E	X	T
Klartext in Ziffern	19	21	2	19	20	9	20	21	20	9	15	14
Verschiebungszahl	19	4	24	19	19	4	24	19	19	4	24	19
addiert	38	25	26	38	39	13	44	40	39	13	39	33
mod 26	12	25	26	12	13	13	18	14	13	13	13	7
Geheimtext	L	Y	Z	L	M	M	R	N	M	M	M	G

Der verschlüsselte Text ist daher „LYZLMMRNMMMG“.

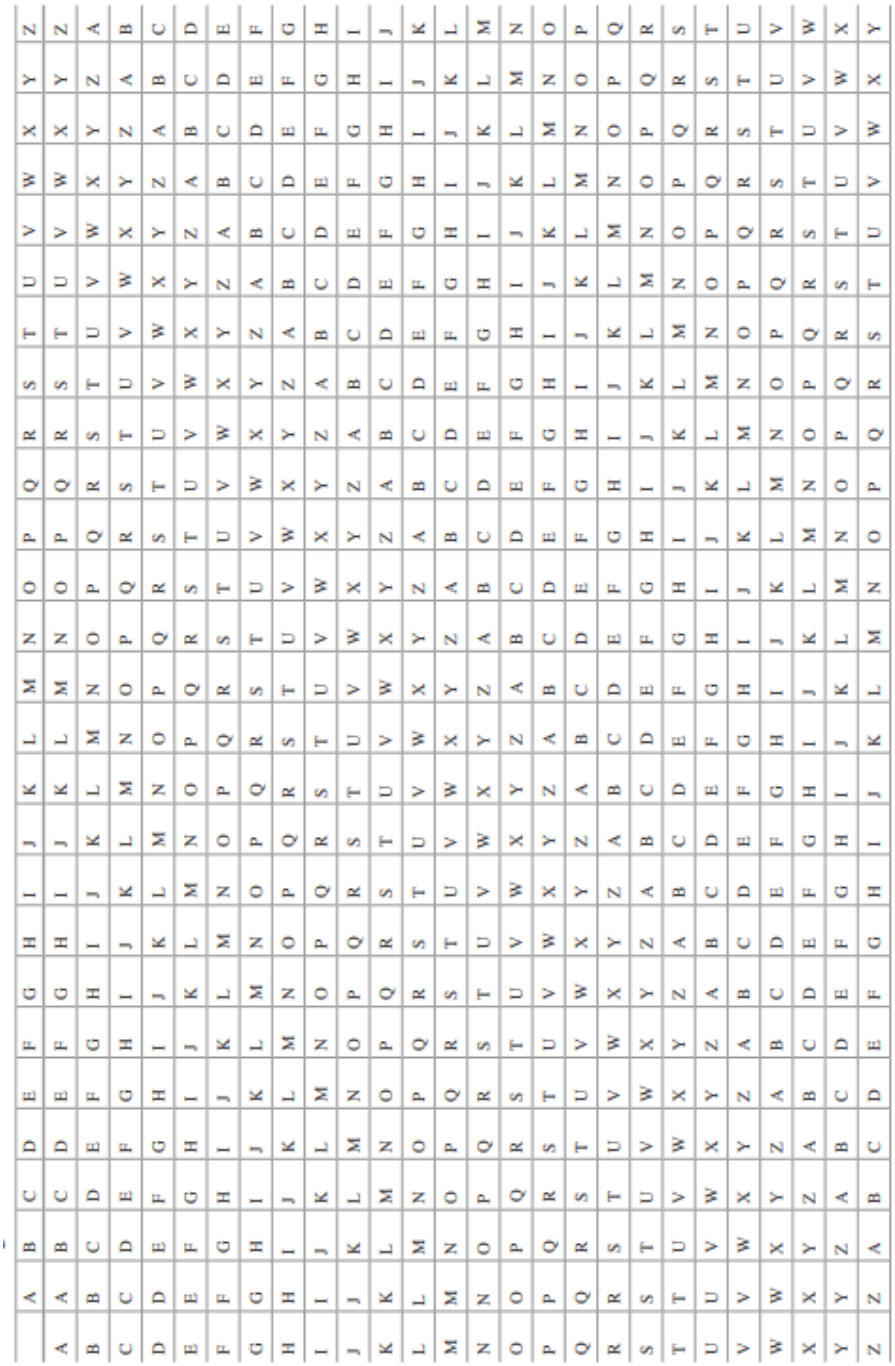


Abbildung 3.3: Vigenère-Tabelle

4 Moderne symmetrische Verschlüsselungen

Historische Verschlüsselungssysteme sind nach heutigen Standards unsicher und können daher schon seit vielen Jahren nicht mehr verwendet werden. Daher wurden mit der Zeit weitere symmetrische Verschlüsselungsverfahren entwickelt. Vorab wird noch der Begriff Blockchiffre erläutert, da moderne Verschlüsselungsverfahren auf Blockchiffren basieren. [16]

4.1 Blockchiffre

Definition. Sei Σ ein Alphabet und n eine natürliche Zahl, welche die Blocklänge angibt, dann ist ein Blockchiffre ein determinitstisches Verschlüsselungsverfahren, dessen Klartext- und Schlüsseltextraum die Menge Σ^n aller Wörter der Länge n über Σ sind. Blockchiffren der Länge 1 sind Substitutionschiffren.

[8]
Bei Blockchiffren wird also für jeden Klartext mit selbem Schlüssel jeweils derselbe Geheimtext erzeugt. Der Klartext- und Schlüsselraum wird in Blöcke der Länge n geteilt, welche dann separat verschlüsselt werden.
Claude Shannon formulierte 1949 zwei Gütekriterien für Blockchiffre.

- Konfusion
  Zwischen dem Klartext und dem Geheimtext soll keine Beziehung erkennbar sein, da diese für Angriffe ausgenutzt werden könnte. Besonders betroffen davon sind statistische Verteilungen der Zeichen im Klar- und Geheimtext.

- Diffusion
  Der Chiffretext soll durch alle Zeichen des Klartextes und Schlüssels beeinflusst werden.

[16]

4.1.1 Feistel-Chiffre

Die Feistel-Chiffre wurde 1973 von Horst Feistel entwickelt und bietet eine Grundlage für moderne Blockchiffren. Für die Feistel-Chiffre wird ein binäres Alphabet $0, 1$ benötigt. Der Klartext m wird in Blöcke $m_1, ..., m_n$ unterteilt. Die Blocklänge beträgt $2t$ wobei $t \in \mathbb{N}$. Ein Schlüsselraum K wird definiert. Dieser enthält $k_1, ..., k_n$ Schlüssel, welche in verschiedenen Runden verwendet werden. Jeder Block wird nun in eine linke Hälfte L_1 und eine rechte Hälfte R_1 aufgeteilt. Diese Hälften sind t lang. Anschließend werden

$r \geq 1$ Runden durchgeführt, dabei kann r frei gewählt werden. In jeder Runde wird die linke Hälfte und die rechte Hälfte des Blockes neu berechnet. Das Schema zum Berechnen der Hälften wird im Folgenden erklärt. Dazu wird noch die Funktion F benötigt. Sie ist die Rundenfunktion, die mit dem rundenspezifischen Schlüssel k_i angewendet wird. Meist ist F eine bijektive Funktion, dies ist aber nicht unbedingt notwendig.

$$R_{i+1} := L_i \oplus F_{k_i}(R_i)$$
$$L_{i+1} := R_i$$

Der Vorteil dieser Art der Verschlüsselung ist, dass F nicht invertierbar sein muss. Zur Entschlüsselung kann die selbe Funktion verwendet werden.

$$R_i = L_{i+1}$$
$$L_i := R_{i+1} \oplus F_{k_i}(R_i + 1)$$

[8]

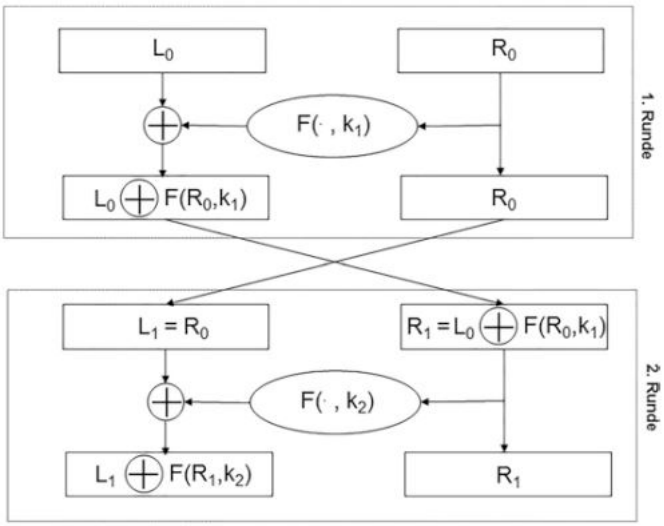


Abbildung 4.1: Vorgehensweise der Feistel-Chiffre

[16]

4.2 DES-Algorithmus

Der DES-Algorithmus galt, in der USA, lange als Standard und wurde auch weltweit eingesetzt. Doch auch der DES-Algorithmus gilt nach heutigem Standard nicht mehr als sicher.
Er war ein wichtiger Grundstein für heutige symmetrische Verschlüsselungsverfahren. Er beruht auf dem Feistel-Chiffre. Für den DES-Algorithmus wird ein binäres Alphabet

$\{0,1\}$ benötigt und die Blocklänge beträgt 64. Der Klartext- und Geheimtextraum ist je $\{0,1\}^{64}$. Die Schlüssel, die beim DES-Algorithmus zur Anwendung kommen, sind Bitstrings welche die Länge 64 besitzen. Ein Bitstring ist eine Abfolge von Bits. Jeder Bit repräsentiert eine 0 oder 1. Daher ist ein Bitstring mit Länge 64 eine Abfolge mit 64 Stellen von Nullen oder Einsen. Diese Bitabfolge muss beim DES-Algorithmus jedoch eine Eigenschaft besitzen. Wird der Schlüssel in 8 Bytes aufgeteilt, wobei jedes Byte 8 Bit beinhaltet, so muss bei jedem Byte das letzte Bit so gewählt werden, dass die Quersumme aller Bits im betreffenden Byte ungerade ist. Für den Schlüsselraum K gilt also folgende Definition:

Definition. $K = \{(b1, ..., b4) \in \{0,1\}^{64} : \sum_{i=1}^{8} b_{8k+i} \equiv 1 \bmod 2,\ 0 \leq k \leq 7\ \}$

Mit $b_1, ..., b_{64}$ werden die Bits des Schlüssels bezeichnet. Bei einem DES-Schlüssel können nur 56 Bits, immer die ersten 7 eines Bytes, frei gewählt werden. Das letzte Bit pro Byte ergibt sich durch die Eigenschaft, dass die Quersumme ungerade ist. Dies ist nötig für die Korrektur von Speicher- und Übertragungsfehlern. Durch diese Voraussetzungen ergeben sich 2^{56} verschiedene Möglichkeiten für einen DES-Schlüssel. Da es sich um eine symmetrische Verschlüsselung handelt, wird zur Ver- und Entschlüsselung derselbe Schlüssel verwendet. Dieser kann als Hexadezimalzahl oder auch als Binärzahl geschrieben werden.
Nachdem der Schlüssel festgelegt wurde, wird der Klartext p einer initalen Perumutation unterzogen. Diese Permutation findet nur vor der ersten Runde statt. Sie ist vom Verfahren festgelegt und vom Schlüssel unabhängig. Die initale Permutation ist in der DES-Spezifikation vorgegeben. Sie sieht wie folgt aus: In der ersten Zeile steht die umgekehrte zweite Spalte von p. In der zweiten Zeile steht die umgekehrte vierte Spalte von p usw. . Zusammengefasst bedeutet das, ist $p \in 0,1$ mit $p = p_1, p_2, p_3, ...p_{64}$ so ist $IP(p) = p_{58}, p_{50}, p_{42}, ..., p_7$.

Beispiel. Die Hexadezimalzahl $p = 012345678ABCDEF$ soll mithilfe des DES-Algorithmus verschlüsselt werden. Die Binärentwicklung des Textes kann der Grafik entnommen werden.

0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	1
0	1	1	0	0	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	1
1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1

Abbildung 4.2: Binärentwicklung von 0123456789ABCDEF

Die inital Permutation sieht wie folgt aus.

1	1	0	0	1	1	0	0
0	0	0	0	0	0	0	0
1	1	0	0	1	1	0	0
1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0
1	0	1	0	1	0	1	0
1	1	1	1	0	0	0	0
1	0	1	0	1	0	1	0

Abbildung 4.3: Initial Permutation von 0123456789ABCDEF

Wird die Initial Permutation nun in die linke Hälfte und in die rechte Hälfte geteilt, so entsteht:

$L_0 = 11001100000000000110011001111111$
$R_0 = 11110000101010101111000010101010.$

Als Nächstes wird 16 Mal eine Feistel-Chiffre angewandt. Dazu wird für jede Runde ein Rundenschlüssel erstellt. Dieser wird aus dem DES-Schlüssel abgeleitet. Für den Rundenschlüssel K_i mit $1 \leq i \leq 16$ muss zunächst v_i definiert werden.

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

Weiters werden zwei Funktionen PC1 und PC2 benötigt.

$$\text{PC1: } \{0,1\}^{64} \to \{0,1\}^{28} \times \{0,1\}^{28}$$
$$\text{PC2: } \{0,1\}^{28} \times \{0,1\}^{28} \to \{0,1\}^{48}$$

Durch die Funktion $PC1$ wird ein Bitstring k mit Länge 64 auf zwei weitere Bitstrings C und D mit jeweils Länge 28 abgebildet. Das bedeutet, der ursprüngliche Schlüssel wird um 8 Bits verkürzt. Dabei handelt es sich um die Bits, die angeben, dass das Byte ungerade ist. $PC1$ bewirkt nun, dass die Reihenfolge der Bits permutiert und in zwei Hälften geteilt wird. C und D werden jeweils um eine bestimmte Anzahl an Positionen nach links verschoben, welches v_i angibt. $PC2$ bildet einen Bitstring der Länge 56, welcher aus je einem Paar von C und D besteht, auf einen String der Länge 48 ab. Dabei ist $PC2$ erneut eine Permutationsoperation. Wie genau die Bitstrings aussehen, kann aus der folgenden Grafik entnommen werden.

PC1						
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

PC2					
14	17	11	24	1	5
3	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	42	50	36	29	32

Abbildung 4.4: Permutationsfunktionen des Schlüssels

Beispiel. Als Schlüssel wird die Hexzahl 133457799BBCDFF1 verwendet, welches der Binärentwicklung 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001 entspricht. Daraus berrechnet sich

$C_0 = 11110000110011001010101111$
$D_0 = 01010101011001100111100011111$
$C_1 = 11100001100110010101011111$
$D_1 = 10101010110011001111100011110$

Der Rundenschlüssel für die erste Runde ist daher
$K_1 = 0001\ 1011\ 0000\ 0010\ 1110\ 1111\ 1111\ 1100\ 0111\ 0000\ 0111\ 0010$.

In jeder der 16 Runden werden die zu verschlüsselnden Daten erweitert, substituiert und eine Permutation vorgenommen. Bei der Erweiterung, auch Expansion genannt, wird die 32-Bit-Datenhälfte auf 48 Bit erweitert, um eine größere Eingabe für die Substitutionsoperationen zu erhalten. Expansion wird durch die Expansionsfunktion $E : \{0,1\}^{32} \to \{0,1\}^{48}$ angegeben. Wie $E(R)$ aussieht, kann aus der Tabelle entnommen werden.

E					
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

P			
16	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

Abbildung 4.5: Funktion E und P

Im nächsten Schritt wird der String $E(R) \oplus K$ gebildet und in 8 Blöcke $B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8$ mit jeweils Länge 6 aufgeteilt. Es gilt daher $E(R) \oplus K = B_1B_2B_3B_4B_5B_6B_7B_8$.
Als Nächstes werden Funktionen $S_i : \{0,1\}^6 \to \{0,1\}^4$ mit $1 \leq i \leq 8$ gebildet. Diese Funktionen werden S-Boxen genannt. Jede S-Box kann durch eine Tabelle mit vier Zeilen und

16 Spalten beschrieben werden. $S_i(B)$ mit $B = b_1b_2b_3b_4b_5b_6$ berechnet den Funktionswert, indem die Binärentwicklung von b_1b_6 als Zeilenindex und die Binärentwicklung $b_2b_3b_4b_5$ als Spaltenindex interpretiert wird. Die Einträge der S-Boxen werden binär dargestellt und fehlende Stellen mit 0 aufgefüllt, sodass die Länge jeweils 4 beträgt. Nun werden alle Ergebnisse von $S_i(B_i)$ mit $1 \leq i \leq 8$ aneinandergefügt und einer letzten Permutation unterzogen. Das Ergebnis ist $f_K(R)$.

S1	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	0101	0011	1000
10	0100	0001	1110	1000	1101	0110	0010	1011	1111	1100	1001	0111	0011	1010	0101	0000
11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101

S2	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1111	0001	1000	1110	0110	1011	0011	0100	1001	0111	0010	1101	1100	0000	0101	1010
01	0011	1101	0100	0111	1111	0010	1000	1110	1100	0000	0001	1010	0110	1001	1011	0101
10	0000	1110	0111	1011	1010	0100	1101	0001	0101	1000	1100	0110	1001	0011	0010	1111
11	1101	1000	1010	0001	0011	1111	0100	0010	1011	0110	0111	1100	0000	0101	1110	1001

S3	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1010	0000	1001	1110	0110	0011	1111	0101	0001	1101	1100	0111	1011	0100	0010	1000
01	1101	0111	0000	1001	0011	0100	0110	1010	0010	1000	0101	1110	1100	1011	1111	0001
10	1101	0110	0100	1001	1000	1111	0011	0000	1011	0001	0010	1100	0101	1010	1110	0111
11	0001	1010	1101	0000	0110	1001	1000	0111	0100	1111	1110	0011	1011	0101	0010	1100

S4	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0111	1101	1110	0011	0000	0110	1001	1010	0001	0010	1000	0101	1011	1100	0100	1111
01	1101	1000	1011	0101	0110	1111	0000	0011	0100	0111	0010	1100	0001	1010	1110	1001
10	1010	0110	1001	0000	1100	1011	0111	1101	1111	0001	0011	1110	0101	0010	1000	0100
11	0011	1111	0000	0110	1010	0001	1101	1000	1001	0100	0101	1011	1100	0111	0010	1110

S5	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

S6	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1100	0001	1010	1111	1001	0010	0110	1000	0000	1101	0011	0100	1110	0111	0101	1011
01	1010	1111	0100	0010	0111	1100	1001	0101	0110	0001	1101	1110	0000	1011	0011	1000
10	1001	1110	1111	0101	0010	1000	1100	0011	0111	0000	0100	1010	0001	1101	1011	0110
11	0100	0011	0010	1100	1001	0101	1111	1010	1011	1110	0001	0111	0110	0000	1000	1101

S7	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0100	1011	0010	1110	1111	0000	1000	1101	0011	1100	1001	0111	0101	1010	0110	0001
01	1101	0000	1011	0111	0100	1001	0001	1010	1110	0011	0101	1100	0010	1111	1000	0110
10	0001	0100	1011	1101	1100	0011	0111	1110	1010	1111	0110	1000	0000	0101	1001	0010
11	0110	1011	1101	1000	0001	0100	1010	0111	1001	0101	0000	1111	1110	0010	0011	1100

S8	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1101	0010	1000	0100	0110	1111	1011	0001	1010	1001	0011	1110	0101	0000	1100	0111
01	0001	1111	1101	1000	1010	0011	0111	0100	1100	0101	0110	1011	0000	1110	1001	0010
10	0111	1011	0100	0001	1001	1100	1110	0010	0000	0110	1010	1101	1111	0011	0101	1000
11	0010	0001	1110	0111	0100	1010	1000	1101	1111	1100	1001	0000	0011	0101	0110	1011

Abbildung 4.6: S-Boxen des DES-Algorithmus

Beispiel. Durch die Schlüsseladdition wird der String $E(R) \oplus K = 0110\ 0001\ 0001\ 0111\ 1011\ 1010\ 1000\ 0110\ 0110\ 0101\ 0010\ 0111$ gebildet. Mithilfe der S-Boxen werden nun die verschlüsselten Bytes generiert.

$S_1(011000) = 0101$
$S_2(010001) = 1100$
$S_3(011110) = 1000$
$S_4(111010) = 0010$
$S_5(100001) = 1011$
$S_6(100110) = 0101$
$S_7(010100) = 1001$
$S_8(100111) = 0111$

Somit entsteht der Binärstring S = 0101 1100 1000 0010 1011 1111 1001 0111. Dieser wird nun erneut einer Permutation unterzogen, welche in der untenstehenden Tabelle angegeben ist. Durch die Permutationstabelle P ergibt sich
$f_{K1}(R_0) = 00100011010010101010100110111011$

P			
16	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

Abbildung 4.7: Permutationstabelle P

[8]

4.3 AES-Algorithmus

Da der DES-Algorithmus als unsicher galt, gab es 1997 eine Ausschreibung für eine Nachfolge des Algorithmus. Bei dieser Ausschreibung fiel die Wahl auf das Rijndael-Verfahren von Joan Daemen und Vincent Rijmen. Dieses Verfahren wurde dann als AES (Advanced Enryption Standard) offiziell zum Federal Information Processing Standard in den USA.
Als Grundlage für diesen Standard wird die Addition und Multiplikation von Bytes benötigt. Dafür werden zwei Eigenschaften vorausgesetzt.

(i). Es existiert ein additives Inverses, sodass gilt $B \oplus \overline{B} = 0...0$

(ii). Es existiert ein multiplikatives Inverses, sodass gilt $B \odot \overline{B} = 0...1$

Bei Bitblocks handelt es sich mathematisch gesehen um einen Körper.

Im Gegensatz zum DES-Algorithmus baut der AES auf Bytes und nicht auf Bits auf. Ein Klartextblock des AES hat 128 Bit, also 16 Bytes. Alle Bytes werden in die AES-State-Matrix eingetragen, sodass eine 4×4 Matrix entsteht.

SubBytes

Der eigentliche Substitutionsanteil des AES-Verfahrens ist der SubByte. Es ist eine nicht lineare Funktion und transformiert die einzelnen Bytes. Diese Transformation wird, wie beim DES-Verfahren, S-Box genannt. Aus jedem Byte a aus der State-Matrix entsteht mithilfe der S-Box ein neues Byte. Dafür wird für jedes Byte a_i das multiplikative Inverse berechnet, sodass $b_i = a_i^{-1}$ falls a_i ungleich 00000000 ist. Ist $a_i = 00000000$, dann wird für $b_i = 00000000$ gesetzt. Jedes Byte $b_i = \beta_7^{(i)}...\beta_0^{(i)}$ wird als Bitstring der Länge 8 geschrieben. Die Bits $\beta_7^{(i)}...\beta_0^{(i)}$ werden folgendermaßen transformiert.

$$\begin{array}{lll} \beta_0^{(i)} \rightarrow \beta_{0'}^{(i)} = \lambda_{00} \cdot \beta_0^{(i)} + \ldots + \lambda_{07} \cdot \beta_7^{(i)} + \delta_0 \\ \vdots \quad\quad \vdots \quad\quad\quad\quad\quad \vdots \\ \beta_7^{(i)} \rightarrow \beta_{7'}^{(i)} = \lambda_{70} \cdot \beta_0^{(i)} + \ldots + \lambda_{77} \cdot \beta_7^{(i)} + \delta_7 \end{array}$$

Abbildung 4.8: AES-Transformation

Dabei sind die Bits δ_{ji} und γ_j vom AES-Algorithmus vorgegeben.

$$(\lambda_{ji}) = \begin{pmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{pmatrix} \text{ und } (\delta_j) = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

Abbildung 4.9:

Durch SubByte wird dann jedes Byte a_i durch die transformierten Bytes in der State-Matrix ersetzt.

AES-ShiftRow

ShiftRow ist der zweite Baustein des AES-Algorithmus. Er verändert die Zeilen der State-Matrix. Die erste Zeile bleibt dabei jedoch unverändert. Die zweite Zeile wird eine

Stelle zyklisch nach links geschoben, die dritte Zeile zwei Stellen und die vierte drei Stellen. Dabei wird ein jedes Byte a_i' in ein Byte a_i'' gewandelt.

AES-MixColum

Der nächste Baustein sorgt dafür, dass die Spalten der State-Matrix verändert werden. $e = 00000001$ bezeichnet das Einselement und für s gilt $s = e \oplus t = 00000011$. Die Einträge $a_1''', a_2''', a_3''', a_4'''$ werden folgendermaßen berechnet.

$$
\begin{aligned}
a_1''' &= t \odot a_1'' \oplus s \odot a_2'' \oplus e \odot a_3'' \oplus e \odot a_4'' \\
a_2''' &= e \odot a_1'' \oplus t \odot a_2'' \oplus es \odot a_3'' \oplus e \odot a_4'' \\
a_3''' &= e \odot a_1'' \oplus e \odot a_2'' \oplus t \odot a_3'' \oplus s \odot a_4'' \\
a_4''' &= s \odot a_1'' \oplus e \odot a_2'' \oplus e \odot a_3'' \oplus t \odot a_4''
\end{aligned}
$$

Die Element $a_5''', ...a_16'''$ aus der State-Matrix werden auf die selbe Art gebildet.

AES-AddRoundKey
Der letzte Baustein für die Chiffrierung des AES ist der AddRoundKey. Pro Runde wird wie beim DES-Algorithmus ein Rundenschlüssel erstellt. Dieser wird auf Basis des 128-, 192- oder 256-stellligen AES Schlüssels konstruiert. Bei einem 128-stellligen Schlüssel wird dieser zunächst in vier Blöcke k_0, k_1, k_2, k_3 mit jeweils 32 Bit unterteilt. AddRoundKey addiert nun in der Vorrunde den eigentlichen AES-Schlüssel k als Rundenschlüssel bitweise auf den Klartextblock $a_1a_2...a_{16}$. Jede Runde erhält einen 32-Bit Block, welcher sich rekursiv ableitet.

$$
\begin{array}{lcll}
k_{4j} & = & k_{4j-4} & \oplus \quad T(k_{4j-1}) \\
k_{4j+1} & = & k_{4j-3} & \oplus \quad k_{4j} \\
k_{4j+2} & = & k_{4j-2} & \oplus \quad k_{4j+1} \\
k_{4j+3} & = & k_{4j-1} & \oplus \quad k_{4j+2}
\end{array}
$$

Dabei ist T die Transformation des 32-Bit-Blocks k_{4j-1}. Diese Transformation besteht aus 4 Byte.

Definition. Seien c_1^j bis c_4^j 4 Bytes und $k_{4j-1} = c_1^j c_2^j c_3^j c_4^j$ dann sieht die Transformation T wie folgt aus

$$
\begin{array}{c}
c_1^j \rightarrow S(c_2^j) + T^{j-1} \\
c_2^j \rightarrow S(c_3^j)
\end{array}
$$

$$c_3^j \rightarrow S(c_4^j)$$
$$c_4^j \rightarrow S(c_1^j)$$

Wobei S die Transformation von dem Baustein SubByte und $t = 00000010$ ist.

In jeder Runde wird nun der Rundenschlüssel auf die Verkettung $a_1'''||a_2'''||...||a_{15}'''||a_{16}'''$ der State-Matrix bitweise addiert mit der Verknüpfung $\oplus$.
[8]

5 Asymmetrische Verschlüsselung

Die asymmetrische Verschlüsselung wird auch Public-Key-Verschlüsselung genannt und wurde in den 1970er Jahren erfunden. Mittlerweile existieren verschiedene asymmetrische Verschlüsselungen. Public-Key-Verfahren haben den Vorteil, dass Sender und Empfänger nicht über denselben Schlüssel zum Verschlüsseln beziehungsweise zum Entschlüsseln benötigen. Daher muss der Schlüssel auch nicht gesendet werden und das Risiko, dass eine dritte Person an den Schlüssel gelangt, wird minimiert.

Bei jeder asymmetrische Verschlüsselung haben Sender und Empfänger jeweils zwei verschiedene Schlüssel. Beide Parteien besitzen einen privaten Schlüssel (Private Key) und einen öffentlichen Schlüssel (Public Key). Aufgrund des öffentlichen Schlüssels wird die asymmetrische Verschlüsselung auch Public-Key-Verfahren genannt.

Je nach Anwendungsfall werden die Schlüssel zur Ver- und Entschlüsselung der Nachricht gebraucht. In der Kryptografie wird der Sender einer Nachricht meist „Alice“ und der Empfänger „Bob“ genannt. Im Folgenden werden in dieser Arbeit nun auch die beiden Partein des Nachrichtenaustausches „Alice“ und „Bob“ genannt. Ein Dritter, der die Nachricht abhören möchte, wird „Eve“ genannt.

Bei der klassischen asymmetrischen Verschlüsselung hat Alice einen privaten Schlüssel und einen öffentlichen Schlüssel. Der private Schlüssel dient zum Entschlüsseln und allein Alice kennt diesen Schlüssel. Der öffentliche Schlüssel ist allen bekannt, auch Eve hat Zugriff auf diesen Schlüssel. Beide Schlüssel stehen in einem eindeutigen mathematischen Verhältnis. Verschiedene Verschlüsselungsmethoden verwenden verschiedene mathematische Problemstellungen, um den öffentlichen Schlüssel zu generieren. Daher sind der private Schlüssel und der öffentliche Schlüssel nicht ident, können aber auf den jeweils anderen Schlüssel rückgeführt werden.

Bob besitzt ebenfalls einen öffentlichen und einen privaten Schlüssel. Wenn Alice nun eine Nachricht verschlüsseln möchte, benützt sie Bobs öffentlichen Schlüssel, um die Information zu verschlüsseln. Anschließend sendet sie die verschlüsselte Nachricht an Bob. Daraufhin kann Bob die verschlüsselte Information mit seinem privaten Schlüssel entschlüsseln, da seine beiden Schlüssel in einer eindeutigen mathematischen Beziehung stehen. Die Verschlüsselung einer Nachricht ist somit immer möglich, solange Alice den öffentlichen Schlüssel von Bob kennt. Allein Bob kann die Nachricht entschlüsseln, da er allein Kenntnis über seinen privaten Schlüssel hat. Falls der private Schlüssel von Bob verloren geht, ist es unmöglich die verschlüsselte Nachricht zu entschlüsseln. Bei dieser Methode ist es wichtig, dass die privaten Schlüssel nur ihren „Besitzern“ zugänglich sind.

Dritte können die Nachrichten nicht entschlüsseln, solange sie nicht im Besitz der privaten Schlüssel sind.

Die einzige Schwachstelle der asymmetrischen Verschlüsselung ist, wenn jemand Drittes, Eve, sich zwischen die beiden Kommunikationspartner schaltet. Eve könnte dann selbst öffentliche Schlüssel generieren und sie an Bob und an Alice senden. Dadurch werden Alice und Bob in dem Glauben gelassen, sie kommunizieren miteinander, jedoch kann Eve alle Nachrichten abfangen und lesen. Diese Form des Angriffes wird „man-in-the-middle-Attacke“ genannt. [11]

Es existieren zwei Anforderungen, die eine asymmetrische Verschlüsselung erfüllen muss. Die erste Anforderung ist, dass das Kerckhoff-Prinzip erfüllt ist. Hierbei muss sichergestellt werden, dass die Sicherheit des Verschlüsselungssystems nicht darauf beruhen darf, dass die Ver- und Entschlüsselungsfunktion geheim gehalten wird. Das heißt, dass die Funktion öffentlich bekannt sein darf. Ist dies der Fall, so ergibt sich direkt die zweite Anforderung an asymmetrische Verschlüsselungen. Der Schlüssel darf nicht mit Kenntnis der Funktion zu berechnen sein. Daher ist die zweite Anforderung ein großer Schlüsselraum. Der aktuelle Stand besagt, dass die Schlüssellänge mindestens 128 Bit lang sein soll. Für einige Systeme werden 2048 oder 4096 Bit empfohlen. [10]

5.1 Diffie-Hellman-Verfahren

Das Diffie-Hellman-Verfahren wird in der Kryptografie verwendet um einen sicheren Schlüsselaustausch zu gewährleisten. Es ist daher kein Verschlüsselungsverfahren. Es ermöglicht zwei Parteien, meist werden sie Alice und Bob genannt, ohne vorherige Kommunikation die notwendigen Schlüssel sicher und zuverlässig auszutauschen. Das Verfahren wurde 1976 von Whitfield Diffie und Martin Hellman entwickelt. Es beruht auf dem diskreten-Logarithmus-Problem.

5.1.1 Diffie-Hellman-Schlüsselaustausch

Damit zwei Teilnehmer, Alice und Bob, Informationen austauschen können, steht ihnen zunächst eine unsichere und abhörbare Kommunikationsverbingung zur Verfügung. Über diese Verbindung einigen sie sich auf einen Schlüssel K. Um ihre Nachrichten verschlüsselen und dann auch wieder entschlüsseln zu können, wählen Alice und Bob gemeinsam eine Primzahl p. Weiters einigen sie sich auf eine Primitivwurzel g für die gilt $2 \leq g \leq p-2$. Da Alice und Bob noch keine sichere Verbindung aufgebaut haben, sind g und p öffentlich zugänglich, sie müssen also nicht geheim sein.

Im nächsten Schritt berechnet Alice k_A. Dazu wählt sie zufällig ein $a \in \{0, 1, ..., p-2\}$. Alice teil des öffentlichen Schlüssel berechnet sie wie folgt

$$k_A = g^a \mod p.$$

k_A wird nun öffentlich an Bob gesendet. Wichtig zu beachten ist, dass der Exponent a geheim gehalten wird.
Bob berechnet analog k_B. Dazu wählt er $b \in \{0, 1, ..., p-2\}$ zufällig aus und berechnet

$$k_B = g^b \mod p.$$

Das Ergebnis wird ebenfalls über die öffentliche Verbindung an Alice gesendet. Der Exponent b ist ebenfalls unbedingt geheim zu halten.
Weder hat Alice Informationen über den Exponenten b noch hat Bob Kenntnis über den Exponenten a.
Beide besitzen jedoch genug Informationen, um den geheimen und gemeinsamen Schlüssel K zu berechnen.

Dazu berechnet Alice

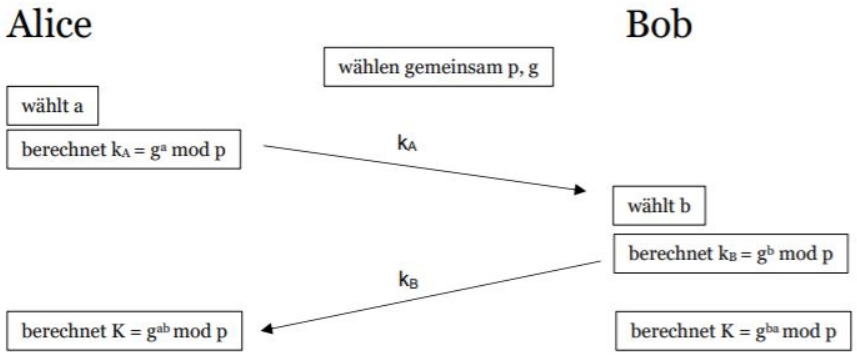
$$k_B^a \mod p = g^{ab} \mod p$$

Bob berechnet

$$k_A^b \mod p = g^{ab} \mod p.$$

Daraus ergibt sich der gemeinsame Schlüssel $K = g^{ab} \mod p$.
[8]

In nachstehender Grafik wird das Diffie-Hellman-Verfahren optisch zusammengefasst:



Beispiel. Im Folgenden wird das Diffie-Hellman-Verfahren exemplarisch berechnet. Um die Funktionsweise hervorzuheben wurden einfache Zahlen verwendet.

Für die öffentlich bekannten Zahlen p, g wählen Alice und Bob $p = 13$ und $g = 2$. Der geheime Exponent von Alice ist $a = 8$.

Nun kann Alice ihren Teil des öffentlichen Schlüssels berechnen.

$$\begin{aligned} k_A &= g^a \mod p \\ k_A &= 2^8 \mod 13 \\ k_A &= 9 \end{aligned}$$

Sie sendet nun $k_A = 9$ an ihren Kommunikationspartner Bob. Dieser wählt für seinen Exponenten $b = 3$ und berechnet ebenfalls seinen Teil des öffentlichen Schlüssels.

$$\begin{aligned} k_B &= g^b \mod p \\ k_B &= 2^3 \mod 13 \\ k_B &= 8 \end{aligned}$$

Nun sendet Bob an Alice $k_B = 8$.

Beide können nun den geheimen Schlüssel berechnen. Alice führt dazu folgende Schritte

aus

$$\begin{aligned} K &= k_B^a \mod p \\ K &= 8^8 \mod 13 \\ K &= 1 \end{aligned}$$

Bob berechnet dazu analog

$$\begin{aligned} K &= k_A^b \mod p \\ K &= 9^3 \mod 13 \\ K &= 1 \end{aligned}$$

Daraus ergibt sich der gemeinsame und geheime Schlüssel $K = 1$.

Beispiel. Ein weiteres Beispiel für das Diffie-Hellman-Verfahren mit größeren Zahlen folgt.

Für p und g einigen sich Alice und Bob auf $p = 97$ und $g = 23$. Alice wählt $a = 20$. Bob wählt $b = 31$.

Alice berechnet nun

$$\begin{aligned} k_A &= g^a \mod p \\ k_A &= 23^{20} \mod 97 \\ k_A &= 43. \end{aligned}$$

Bob berechnet

$$\begin{aligned} k_B &= g^b \mod p \\ k_B &= 23^{31} \mod 97 \\ k_B &= 87. \end{aligned}$$

Nun tauschen Alice und Bob ihre berechneten Werte. Dadurch können beide den gemeinsamen Schlüssel berechnen.

Alice berechnet

$$\begin{aligned} K &= k_B^a \mod p \\ K &= 87^{20} \mod 97 \\ K &= 73. \end{aligned}$$

Bob berechnet

$$\begin{aligned} K &= k_A^b \mod p \\ K &= 43^{31} \mod 97 \\ K &= 73. \end{aligned}$$

Beispiel. Ein weiteres Beispiel zum Diffie-Hellman-Schlüsselaustausch.
Alice und Bob einigen sich auf $p = 27803$ und $g = 5$. Alice wählt außerdem $a = 21131$ und berechnet

$$\begin{aligned} k_A &= g^a \mod p \\ k_A &= 5^{21131} \mod 27803 \\ k_A &= 21420. \end{aligned}$$

Bob hingegen wählt $b = 17555$ und berechnet

$$\begin{aligned} k_B &= g^b \mod p \\ k_B &= 5^{17555} \mod 27803 \\ k_B &= 17100 \end{aligned}$$

Alice sendet nun ihren öffentlichen Schlüssel k_A an Bob und Bob sendet seinen öffentlichen Schlüssel k_B an Alice. Nun können beide den gemeinsamen geheimen Schlüssel berechnen. Dazu rechnet Alice

$$\begin{aligned} K &= k_B^a \mod p \\ K &= 17100^2 1131 \mod 27803 \\ K &= 11134. \end{aligned}$$

Bob rechnet

$$\begin{aligned} K &= k_A^b \mod p \\ K &= 21420^1 7555 \mod 27803 \\ K &= 11134. \end{aligned}$$

[7]
[8]

5.1.2 Sicherheit

Die Werte p, g, k_A und k_B werden über einen öffentlichen Kanal versendet und sind somit für alle einsehbar. Jedoch reichen diese Informationen für einen Angreifer, Eve, nicht um den geheimen Schlüssel zu berechnen, da die diskreten Logarithmen a und b geheim sind. Diese sind essenziell, um den geheimen Schlüssel K zu berechnen. Eve könnte nun

versuchen a und b zu berechnen. Damit dies nicht (leicht) möglich ist muss die Primzahl p und die Primitivwurzel g so ausgewählt werden, dass der diskrete Logarithmus nicht effizient berechenbar ist. Dies ist der Fall wenn die Zahlen groß genug sind. p muss daher eine Primzahl sein die mindestens 2048 Bit groß ist. Dies entspricht einer Zahl mit 617 Ziffern. [1] Der Wert für die Primitivwurzel g soll so gewählt werden, dass die entstehende Untergruppe $\mathbb{Z}_p^*$ möglichst groß ist. Wäre dies nicht der Fall und g würde nur eine kleine Untergruppe erzeugen, so könnte Eve mithilfe einer Tabelle aus allen Potenzen von g die diskreten Logarithmen berechnen. Jedoch wurde bist heute nicht bewiesen, dass ein Angreifer allein mit dem diskreten Laogarithmus $g^{ab} \, mod \, p$ bestimmen kann. [5]

man-in-the-middle

Bei einem man-in-the-middle-Angriff, können die Kommunikationsteilnehmer Alice und Bob nicht sicher sein, dass sie miteinander kommunizieren. Ein Angreifer schaltet sich zwischen die zwei Partein und möchte die jeweiligen Nachrichten abfangen. Da die Primzahl p und die Primitivwurzel g öffentlich zugänglich sind, kann jede Person, auch ein Angreifer, darauf zugreifen. Möchte der Angreifer nun die Information der Nachrichten erhalten, muss er eine Zahl c aus der Menge $\{1, \ldots, p-1\}$ wählen.
Daraufhin berechnet er

$$k_C = g^c \mod p.$$

Nun sendet der Angreifer k_C sowohl an Alice als auch an Bob. Dadurch sind beide Kommunikationsteilnernehmer im Glauben, dass sie eine sichere Verbindung miteinander aufgebaut haben. Alice antwortet daher mit k_A und Bob antwortert mit k_B. Nun ist der Angreifer im Besitz aller Informationen um sich

$$k_{CA} = k_A^c \, mod \, p$$

sowie

$$k_{CB} = k_B^c \, mod \, p$$

zu berechnen. Solange Alice und Bob denken, dass sie miteinander kommunizieren, kann der Angreifer alle Nachrichten mitlesen. [5]

5.2 RSA-Verschlüsselung

Die RSA-Verschlüsselung ist die am weitesten verbreitete asymmetrische Verschlüsselung. Dieses Verfahren wurde von Ronals Rivest, Adi Shamir und Leonard Adleman entwickelt. Durch die Initialen der Entwickler bekam diese Verschlüsselungsmethode ihren Namen. Das RSA-Verfahren wurde 1978 entwickelt. Wie bereits im Kapitel asymmetrische Verschlüsselung erwähnt wurde, liegt jedem Public-Key-Verfahren ein mathematisches Konzept zu Grunde, mit dem die zu übermittelnde Nachricht verschlüsselt wird. Die RSA-Verschlüsselung beruht auf der Folgerung vom Satz von Fermat und Euler. Dafür müssen zunächst einige mathematische Begriffe definiert werden.

Um das Prinzip der RSA-Verschlüsselung zu erläutern, werden die zwei Teilnehmer der Verschlüsselung „Alice“ und „Bob“ genannt.
Bob sendet eine geheime Botschaft an Alice. Die geheime Nachricht wird mit $m \in \mathbf{N}^*$ bezeichnet. Alice möchte diese Nachricht empfangen.

Dazu wählt Alice zwei große Primzahlen $p, q \in \mathbf{N}^*$ mit $p \neq q$, die sie geheim hält. Weiters sei $n = pq$. Zusätzlich wählt Alice eine Zahl $e \in \mathbf{N}^*$ für die gilt

$$1 < e < \phi(n) = (p-1)(q-1) \text{ und } \mathrm{ggT}(e, (p-1)(q-1)) = 1 \ .$$

Alice berechnet die natürliche Zahl d für die gilt

$$1 < d < (p-1)(q-1) \text{ und } de \equiv 1 \bmod ((p-1)(q-1)).$$

Da $\mathrm{ggT}(e, (p-1)(q-1)) = 1$ existiert die Zahl d und sie kann mithilfe des erweiterten euklidischen Algorithmus berechnet werden. e muss eine ungerade Zahl sein. Das Tuppel (n, e) ist nun der öffentliche Schlüssel von Alice und d ist der private Schlüssel von Alice, welcher nicht veröffentlicht werden darf. Üblicherweise ist pq mindestens 1024 Bit lang und p und q jeweils mindestens 1024/2-Bit-Zufallsprimzahlen.
Will Alice nun eine Nachricht von Bob empfangen, so wählt Bob eine Nachricht $m \in \mathbf{Z_m}$. Nun benötigt Bob den öffentlichen Schlüssel (e, n) von Alice. Bob verschlüsselt die Nachricht m folgendermaßen:

$$c = m^e \pmod{n}.$$

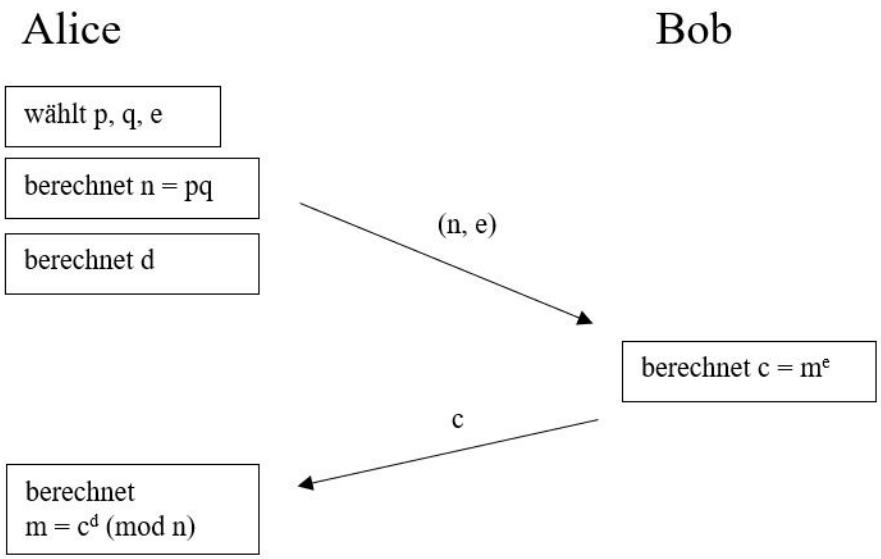
c ist nun die verschlüsselte Nachricht, die Alice erhält. Zur Entschlüsselung berechnet Alice

$$c^d = (m^e)^d = m^{ed} \equiv m^{ed(mod\phi(n))} = m^1 = m \pmod{n}.$$

Wenn m teilerfremd zu n ist, dann kann der Satz von Euler und Fermat angewendet werden. Sonst ist $m = 0$ oder $m = kp$ (mit $k \in \{1, 2, \ldots, q-1\}$) oder $m = kq$ (mit $k \in \{1, 2, \ldots, p-1\}$). Falls $m = kp$ mit $k \in \{1, 2, \ldots, q-1\}$, dann ist $(kp)^{(q-1)} \equiv 1 \mod q$

und daher $(kp)^{1+(p-1)(q-1)} \equiv (kp)\left((kp)^{q-1}\right)^{p-1} \equiv kp \mod q$, und $(kp)^{1+(p-1)(q-1)} \equiv 0 \equiv kp \mod p$. Nach dem Chinesischen Restsatz ist dann $m^{ed} \equiv m \mod n$, da $n = pq$.

Die folgende Grafik zeigt eine optische Zusammenfassung der RSA-Verschlüsselung.



Das RSA-Verfahren kann nur auf kurze Klartexte angewendet werden, da das Verschlüsseln von langen Klartexten sehr langsam ist. In der Praxis werden lange Klartexte daher symmetrisch verschlüsselt und der Schlüssel der symmetrischen Verschlüsselung asymmetrisch übertragen. Diese Art wird Hybridverschlüsselung genannt.

Beispiel. Alice wählt zwei Primzahlen $p = 7$ und $q = 13$. Daraus berechnet sie

$$\begin{aligned} n &= pq \\ n &= 7 \cdot 13 = 91. \\ \phi(91) &= (7-1)(13-1) \\ &= 72. \end{aligned}$$

Für e wählt Alice 11 und somit ist das Tupel $(n, e) = (91, 11)$ Alices öffentlicher Schlüssel. Bob möchte Alice nun die Nachricht $m = 10$ senden, dazu berechnet er

$$\begin{aligned} c &= m^e \mod n \\ c &= 10^{11} \mod 91 \\ c &= 82. \end{aligned}$$

Bob sendet nun $c = 82$ an Alice. Damit Alice die Nachricht entschlüsseln kann, muss sie das modulare Inverse von e, also d berechnen. Mithilfe des erweiterten euklidischen Algorithmus.

$$\begin{array}{lcllcl} 72 & = & 6 \cdot 11 + 6 & \text{also} \quad 6 & = & 72 \text{ - } 6 \cdot 11 \\ 11 & = & 1 \cdot 6 + 5 & \text{also} \quad 5 & = & 11 \text{ - } 1 \cdot 6 \\ 6 & = & 1 \cdot 5 + 1 & \text{also} \quad 1 & = & 6 \text{ - } 1 \;\; 5 \end{array}$$

Durch Zurückrechnen ergibt sich $d = -13 \equiv 59 \pmod{72}$.

$$\begin{array}{rcl} 1 & = & 6 \text{ - } 1 \cdot 5 \\ & = & 6 \text{ - } 1 \cdot (11 \text{ - } 1 \cdot 6) \\ & = & 2 \cdot 6 \text{ - } 1 \cdot 11 \\ & = & 2 \cdot (72 \text{ - } 6 \cdot 11) \text{ - } 1 \cdot 11 \\ & = & 2 \cdot 72 \text{ - } (12 + 1) \cdot 11 \\ & = & 2 \cdot 72 \text{ - } 13 \cdot 11 \end{array}$$

Alice möchte nun die Nachricht von Bob entschlüsseln. Dazu berechnet sie

$$\begin{aligned} c^d &= 82^{59} \mod 91 \\ c^d &\equiv 10 = m. \end{aligned}$$

Beispiel. Ein weiteres Beispiel für die RSA-Verschlüsselung sieht wie folgt aus.

Alice wählt $p = 11$ und $g = 23$. Daraus berechnet sie sie $n = 253$. $\phi(253) = (11 - 1)(23 - 1) = 220$. Für e wählt Alice $e = 3$. Alice öffentlicher Schlüssel ist somit $(220, 3)$. Bob möchte nun die Nachricht $m = 12$ verschlüsseln.

$$\begin{aligned} c &= m^e \mod n \\ c &= 12^3 \mod 253 \\ c &= 210 \end{aligned}$$

Das modulare Inverse kann mithilfe des euklidischen Algorithmus berechnet werden und ist $d = 147$. Somit kann Alice die Nachricht von Bob entschlüsseln.

$$\begin{aligned} c^d &= 210^{147} \mod 253 \\ c &\equiv 12 = m \end{aligned}$$

Beispiel. Ein weiteres Beispiel zur RSA-Verschlüsselung.

Alice wählt zwei Primzahlen $p = 101$ und $q = 113$. Daraus berechnet sie

$$\begin{aligned} n &= pq \\ n &= 101 \cdot 113 \\ n &= 11413. \\ \phi(11413) &= (101-1)(113-1) \\ &= 11200 \end{aligned}$$

Da die Primfaktorenzerlegung von 11200 $2^6 5^2 7$ lautet wählt Alice für den Parameter $e = 3533$. Alice öffentlicher Schlüssel lautet daher $(11413, 3533)$. Bob möchte die Nachricht $m = 9726$ an Alice senden. Dazu berechnet er

$$\begin{aligned} c &= m^e \mod n \\ c &= 9726^{3533} \mod 11413 \\ c &= 5761. \end{aligned}$$

Bob sendet nun Alice die verschlüsselte Nachricht 5761. Damit Alice die Nachricht entschlüsseln kann, berechnet sie das modulare Inverse d von e. Mithilfe des euklidischen Algorithmus erhält Alice $d = 6597$. Nun kann Alice die ursprüngliche Nachricht rekonstruieren.

$$\begin{aligned} c^d &= 5761^{6597} \mod 11413 \\ c^d &\equiv 9726 = m \end{aligned}$$

[8] [14]

5.2.1 Sicherheit des RSA-Algorithmus

Um die Sicherheit des RSA-Alogirthmus zu garantieren, muss es so gut wie unmöglich sein anhand des öffentlichen Schlüssels auf den privaten Schlüssel zu schließen. Weiters ist es wichtig, dass die Zerlegung des RSA-Modul n in seine Primfaktoren schwierig ist.

Kennt ein Angreifer die Primfaktoren p und q kann dieser daraus mithilfe des Verschlüsselungsexponenten e den privaten Schlüssel d berechnen. Dazu muss folgende Kongruenz gelöst werden.

$$de \equiv 1 \mod ((p-1)(q-1))$$

Ebenfalls können mit Kenntnis von n, e, d die Faktoren p und q und somit n berechnet werden.
Um das zu zeigen, werden zunächst s und k definiert.

$$\begin{aligned} s &:= \max\{t \in \mathbf{N} : 2^t | (ed-1)\} \\ k &:= \tfrac{ed-1}{2^s} \end{aligned}$$

Damit n faktorisiert werden kann, wird zufällig und gleichverteilt eine Zahl a aus der Menge $\{1, \ldots, n-1\}$ gewählt. Im nächsten Schritt wird $g = \text{ggT}(a, n)$ berechnet. Falls $g > 1$ ist, ist g ein echter Teiler von n, welcher gesucht wurde. Wenn nicht, also $g = 1$, wird g wie folgt berechnet

$$g = \text{ggT}(a^{2^t k} - 1, n)$$
$$t = s-1, s-2, \ldots, 0$$

Wird dabei ein Teiler von n gefunden, so ist der Algorithmus fertig. Falls kein Teiler gefunden wird, muss ein neues a gewählt werden und die Berechnung neu durchgeführt werden.

Damit die Faktorisierung von n schwierig ist, müssen p und q groß genug gewählt werden. Die folgende Tabelle zeigt eine empfohlene Mindestgröße für p und q.

Schutz bis	Mindestgröße
2015	1248
2020	1776
2030	2432
2040	3248
für absehbare Zukunft	15.424

Abbildung 5.1: Mindestgröße von p und q

Die Primfaktoren p und q werden als zufällige $\frac{k}{2}$-Bit-Primzahlen so gewählt, dass $n = pq$ eine k-Bit-Primzahl ist. Dabei ist wichtig, dass p und q zufällig gewählt werden damit keine Faktorisierungsalgorithmen verwendet werden können, die die spezielle Struktur der Faktoren ausnutzen.
Damit die Sicherheit des RSA-Algorithmus gesteigert wird, ist die richtige Wahl von e wichtig. e soll so gewählt werden, dass die Verschlüsselung effizient möglich ist, jedoch darf die Sicherheit nicht gefährdet werden. e kann nicht 2 sein, da $\phi(n) = (p-1)(q-1)$ gerade ist, jedoch Voraussetzung ist, dass $\text{ggT}(e, (p-1)(q-1)) = 1$. Daher ist der kleinste Verschlüsselungsexponent 3, falls $\text{ggT}(3, (p-1)(q-1)) = 1$.

Beispiel. Gegeben sei $n = 253$, $e = 3$. Die zu verschlüsselnde Nachricht ist $m = 165$. Nun muss $m^e \mod n$ berechnet werden. Dazu kann zunächst $m^2 \mod n = 154$ berechnet werden. Im nächsten Schritt wird $m^3 \mod n = ((m^2 \mod n)m) \mod n = 154 \cdot 165 \mod 253 = 110$ berechnet. Die verschlüsselte Nachricht ist also 110.

Die Verwendung von $e = 3$ ist jedoch gefährlich, falls der Angreifer einen Low-Exponent-Angriff durchführt. Dabei kann ein Angreifer durch Ausprobieren aller möglichen kleinen Werte für m die ursprüngliche Nachricht berechnen, da die Kubikwurzel von c einfach berechnet werden kann, falls $e = 3$. Um dies zu verhindern, muss e groß genug gewählt

werden.

Beim RSA-Algorithmus kann entweder zuerst der öffentliche Schlüssel oder zuerst der private Schlüssel gewählt werden. Im vorherigen Abschnitt wird zunächst der öffentliche Schlüssel gewählt. Daraus ergibt sich der private Schlüssel d, der in derselben Größenordnung wie n liegt.

Es kann aber von Vorteil sein, den privaten Schlüssel zuerst zu wählen und darauf zu achten, dass dieser möglichst klein bleibt. Dies kann beispielsweise der Fall sein, wenn der private Schlüssel auf einer Chipkarte gespeichert wird. Dann ist es aus Sicherheitsgründen gut, wenn die Entschlüsselung ebenfalls auf der Chipkarte stattfindet und diese nicht verlässt. Umso kleiner der private Schlüssel d ist, umso schneller ist die Entschlüsselung. Ist der private Schlüssel jedoch klein, so kann das Verfahren einfach gebrochen werden. [8] [6] **Sichere Verwendung**

Trotz gut gewählter Parameter bietet die RSA-Verschlüsselung Platz für Angriffe. Daher wird in der Praxis der RSA-OAEP verwendet. Dabei steht OAEP für Optimal Asymmetric Encryption. Die Funktionsweise des OAEP lautet wie folgt:
Sei $t \in \mathbf{N}$, für welches gilt, dass die maximale Laufzeit eines Angreiferalgorithmus, deutlicher kleiner als 2^t ist. Weiters sei k die binäre Länge des RSA-Moduls n. Aufgrund von aktuellen Sicherheitsmaßnahmen ist $k > 1024$. Für l gilt $l = k - t - 1$. Es existiert eine Expansionsfunktion

$$G : \{0,1\}^t \longrightarrow \{0,1\}^l$$

und eine Kompressionsfunktion

$$H : \{0,1\}^l \longrightarrow \{0,1\}^t.$$

Beide Funktionen sind öffentlich und für jeden einsehbar. Der Klartextraum ist $\{0,1\}$. Um eine Nachricht $m \in \{0,1\}^l$ verschlüsselt werden, so wird zunächst eine Zufallszahl $r \in \{0,1\}^l$ gewählt. Die verschlüsselte Nachricht c berechnet sich wie folgt

$$c = ((m \oplus G(r)) \circ (r \oplus G(r)))^e \mod n.$$

Um wieder den Klartext zu erhalten wird zunächst

$$(m \oplus G(r)) \circ (r \oplus H(m \oplus G(r))) = c^d$$

berechnet. Im nächsten Schritt wird

$$r = (r \oplus H(m \oplus G(r))) \oplus H(m \oplus G(r))$$

berechnet, damit zum Schluss der Klartext ergibt.

$$m = (m \oplus G(r)) \oplus G(r)$$

Es gilt also, dass der Klartext zu $m \oplus G(r)$ randomisiert wird und der Zufallswert r zu $(r \oplus H(m \oplus G(r)))$ maskiert.
[8]

5.2.2 Anwendung in der Schule

Die mathematischen Voraussetzungen der RSA-Verschlüsselung erlernen Schüler:innen bereits in der Unterstufe. Jedoch werden die Potenzen sehr schnell sehr groß, selbst wenn man als Parameter niedrige Zahlen wählt. Daher ist die RSA-Verschlüsselung nicht geeignet um sie mit Schüler:innen der Unterstufe zu erarbeiten. Jedoch geht die Berechnung der Ver- und Entschlüsselung mit geeigneten technischen Hilfsmittel, einfacher. Daher kann die RSA-Verschlüsselung durchaus mit älteren Schüler:innen thematisiert werden.

5.3 El-Gamal-Verschlüsselung

Bei der El-Gamal-Verschlüsselung handelt es sich um eine spezielle Anwendung des Diffie-Hellman-Schlüsselaustauschs. Beide beruhen auf dem Problem des diskreten Logarithmus. Mit der El-Gamal-Verschlüsselung ist es jedoch nicht nur möglich einen Schlüssel sicher auszutauschen, mit dem Verfahren können auch Informationen ver- und entschlüsselt werden. Um das El-Gamal-Verfahren anwenden zu können muss die Nachricht in Form einer Zahl vorliegen.

Die Vorgehensweise ist ähnlich wie beim Diffie-Hellman-Verfahren. Zwei Teilnehmer, Alice und Bob, möchten zunächst eine sichere Verbindung aufbauen und dann Informationen austauschen.
Dazu wählt Alice, analog zum Diffie-Hellman-Verfahren, eine Primzahl p und eine Primitivwurzel g. p und g müssen sorgsam gewählt werden, denn für die beiden Zahlen muss $2 \leq g \leq p-2$ gelten. Weiters wählt Alice ebenfalls eine Zahl $a \in \{1, ..., p-2\}$. Daraus berechnet Alice

$$k_A = g^a \mod p.$$

Beim El-Gamal-Verfahren wird nun nicht wie beim Diffie-Hellman-Verfahren k_A veröffentlicht, sondern Alice sendet das Tripel (p, g, k_A), welches aus der Primzahl, der Primitivwurzel und dem berechneten Wert k_A besteht. Dieses Tripel ist nun der öffentliche Schlüssel von Alice, auf den Bob und jeder Außenstehende zugreifen kann. Wollen Alice und Bob öfter miteinander kommunizieren, so bleibt dieser Schlüssel erhalten. Es muss daher nur einmal ein Schlüssel konfiguriert werden. [7]

Da das Ziel der El-Gamal-Verschlüsselung nicht nur der sichere Austausch der Schlüssel ist, sondern auch das Senden und Empfangen verschlüsselter Nachrichten, möchte Bob nun einen Klartext m verschlüsseln. Wie bereits beschrieben, muss diese Nachricht in Form einer Zahl vorliegen. Dies kann zum Beispiel eine Binärzahl sein. Um nun die Nachricht m zu verschlüsseln, benötigt Bob das Tripel (p, g, k_A), welches Alice veröffentlicht hat. Die nächsten Schritte verlaufen analog zum Diffie-Hellman-Verfahren. Bob wählt ebenfalls ein $b \in \{1, \ldots, p-2\}$ und berechnet.

$$k_B = g^b \mod p$$

Beim Diffie-Hellman-Verfahren wäre nun $k_A^b \, mod \, p = g^{ab} \mod p$ der öffentliche Schlüssel. Beim El-Gamal-Verfahren hingegegen multipliziert Bob den Klartext m mit diesem Ausdruck und erhält

$$c = k_A^b m \mod p.$$

Die verschlüsselte Nachricht besteht daher aus dem Tupel (k_B, c). [8]
Damit Alice die Informationen aus der verschlüsselten Nachricht erhält, muss sie die Nachricht erst entschlüsseln. Dazu benötigt sie zwei Schritte.

Im ersten Schritt berechnet Alice den gemeinsamen Schlüssel K. Die Vorgehensweise verläuft ebenfalls analog zum Diffie-Hellman-Verfahren.

$$K = k_B^a \mod p$$

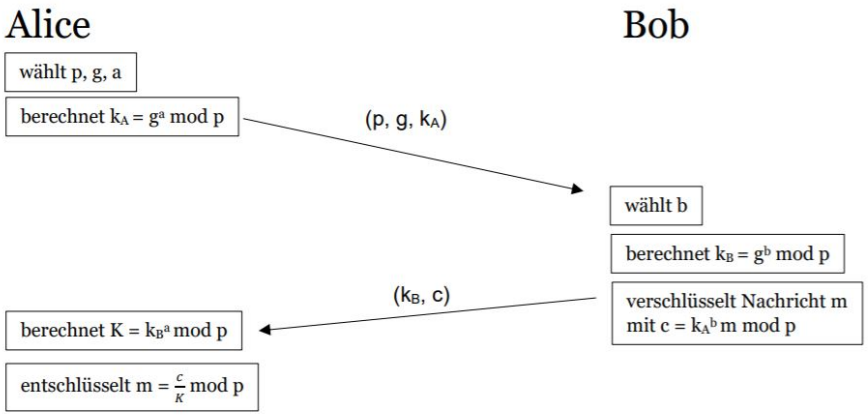
Mithilfe des gemeinsamen Schlüssels kann Alice nun den Klartext m wiederherstellen. Dazu berechnet sie

$$m' = \frac{c}{K} \mod p.$$

Für m' gilt $m' = m$ wegen

$$m = \frac{K \cdot m}{K} \mod p = \frac{k_A \cdot m}{k_B} \mod p = \frac{g^{ab} \cdot m}{g^{ba}} \mod p = m'.$$

Im folgenden ist eine optische Zusammenfassung der El-Gamal-Verschlüsselung:



Anhand mehrerer Beispiele mit einfachen Zahlen soll nun die Vorgehensweise der El-Gamal-Verschlüsselung aufgezeigt werden.

Beispiel. Alice muss die drei Werte p, g und a wählen. Als Primzahl wählt Alice $p = 97$ die Primitivwurzel ist $g = 37$ und für a wählt sie $a = 20$.
Für ihren öffentlichen Schlüssel muss sie noch den Wert k_A berechnen.

$$\begin{aligned} k_A &= g^a \mod p \\ k_A &= 37^{20} \mod 97 \\ k_A &= 88 \end{aligned}$$

Danach kann Alice das Tripel (p, g, k_A) veröffentlichen, welches in diesem Fall $(97, 37, 88)$ ist.

Im nächsten Schritt muss Bob für b einen Wert wählen. Er entscheidet sich für $b = 31$. Weiters möchte Bob die Nachricht $m = 17$ verschlüsseln. Dazu stellt er folgende Rechnung auf:

$$\begin{aligned} k_B &= g^b \mod p \\ k_B &= 37^{31} \mod 97 \\ k_B &= 56 \end{aligned}$$

Zusätzlich berechnet er:

$$\begin{aligned} c &= k_A^b \cdot m \mod p \\ c &= 88^{31} \cdot 17 \mod 97 \\ c &= 68 \end{aligned}$$

Nun kann Bob die verschlüsselte Nachricht an Alice senden. Er veröffentlicht daher das Tupel $(k_B, c) = (56, 68)$. Damit Alice die Informationen aus der verschlüsselten Nachricht erhält, muss sie den gemeinsamen Schlüssel K berechnen.

$$\begin{aligned} K &= k_B^a \mod p \\ K &= 56^{20} \mod 97 \\ K &= 4 \end{aligned}$$

Nun kann Alice den Klartext m rekonstruieren.

$$\begin{aligned} m &= \frac{c}{K} \mod p \\ m &= \frac{68}{4} \mod p \end{aligned}$$

Zu beachten ist, dass es sich hierbei nicht um die Division im Zahlenraum der rationalen Zahlen handelt, sondern um die Division im Restklassenkörper $\mathbb{Z}_{97}$. Um die Division durchzuführen, muss Alice das multiplikative Inverse von 4, also $4^{-1} \in \mathbb{Z}_{97}$ finden. Alice kann mithilfe des erweiterten euklidischen Algorithmus und der Vielfachsummendarstellung das multiplikative Inverse ausfindig machen. Nach Anwendung ergibt sich für das multiplikative Inverse $4^{-1} = 73$. Mithilfe dieser Information kann Alice nun die Nachricht weiter entschlüsseln.

$$\begin{aligned} m &= 68 \cdot 73 \mod 97 \\ m &= 17 \end{aligned}$$

Alice enthält daher den Klartext $m = 17$.

Beispiel. Es folgt ein weiteres Beispiel um die Vorgehensweise zu verdeutlichen. Alice wählt $p = 23,\ g = 7,\ a = 6$. Danach berechnet sie

$$\begin{aligned} k_A &= g^a \mod p \\ k_A &= 7^6 \mod 23 \\ k_A &= 4. \end{aligned}$$

Bob wählt $b = 3$ und kann sich daher seinen privaten Schlüssel k_B berechnen.

$$\begin{aligned} k_B &= g^b \mod p \\ k_B &= 7^3 \mod 23 \\ k_B &= 21 \end{aligned}$$

Bob möchte nun die Nachricht $m = 7$ berechnen.

$$\begin{aligned} c &= k_A^b \cdot m \mod p \\ c &= 4^3 \cdot 7 \mod 23 \\ c &= 11 \end{aligned}$$

Nun kann Bob das Tupel (21,11) an Alice senden. Um die Nachricht zu entschlüsseln, berechnet Alice wiederum

$$\begin{aligned} K &= k_B^a \mod p \\ K &= 21^6 \mod 23 \\ K &= 18. \end{aligned}$$

Dadurch kann sie die Nachricht von Bob entschlüsseln.

$$\begin{aligned} m &= \frac{c}{K} \mod p \\ m &= \frac{11}{18} \mod 23 \\ m &= 7 \end{aligned}$$

Beispiel. Ein weiteres Beispiel bei dem ebenfalls $p = 23$. Jedoch wählt Alice für $g = 2$ und $a = 5$.
Sie berechnet ihren öffentlichen Schlüssel.

$$\begin{aligned} k_A &= g^a \mod p \\ k_A &= 2^5 \mod 23 \\ k_A &= 9. \end{aligned}$$

Bob wählt $b = 7$ und kann sich daher seinen privaten Schlüssel k_B berechnen.

$$\begin{aligned} k_B &= g^b \mod p \\ k_B &= 2^7 \mod 23 \\ k_B &= 13 \end{aligned}$$

Bob möchte nun die Nachricht $m = 8$ verschlüsseln.

$$\begin{aligned} c &= k_A^b \cdot m \mod p \\ c &= 9^7 \cdot 8 \mod 23 \\ c &= 9 \end{aligned}$$

Bob sendet nun das Tupel (13,9) an Alice. Um die Nachricht zu entschlüsseln, berechnet Alice wiederum

$$\begin{aligned} K &= k_B^a \mod p \\ K &= 13^5 \mod 23 \\ K &= 4. \end{aligned}$$

Dadurch kann sie die Nachricht von Bob entschlüsseln.

$$\begin{aligned} m &= \frac{c}{K} \mod p \\ m &= \frac{9}{4} \mod 23 \\ m &= 8 \end{aligned}$$

Die Parameter der letzten beiden Beispiele wurde sehr ähnlich und p sogar gleich gewählt. Jedoch ist dies anhand des Schlüssels und der Verschlüsselung nicht ersichtlich.

5.3.1 Sicherheit

Ist bei der El-Gamal-Verschlüsselung der Schlüssel nicht bekannt, so wird der diskrete Logarithmus benötigt, um eine Nachricht zu entschlüsseln. Wenn dieser jedoch nicht effizient berechnet werden kann, sind die verschlüsselten Nachrichten sicher. Die Wahl von p und g ist ähnlich wie beim Diffie-Hellman-Verfahren. Die Werte werden so gewählt, dass

sie 2048 Bit groß sind. p soll eine weitere Bedingung erfüllen, der diskrete Logarithmus soll schwierig mit den bekannten Verfahren zu berechnen sein. Jedoch sind bis heute nicht alle Verfahren zur Berechnung des diskreten Logarithmus bekannt, daher gilt für p, dass es zufällig und gleichverteilt sein soll.

Die beiden Exponenten a und b sind zufällig gewählt und gleichverteilt in $\{0, \ldots, p-1\}$. Auch die verschlüsselte Nachricht ist zufällig gewählt und gleichverteilt in $\{1, \ldots, p-1\}^2$. Daraus ergibt sich, dass die El-Gamal-Verschlüsselung randomisiert ist. Die Randomisierung verursacht, dass die Verschlüsselung semantisch sicherer ist. Jedoch ist es einem Angreifer möglich, die Nachricht zu verfälschen bzw. zu verändern ohne, dass er die Nachricht entschlüsselt. Ist die Verschlüsselung von m das Tupel (k_B, c), dann ist die Verschlüsselung von $mx \mod p$ das Tupel $(k_B, cx \mod p)$ für alle $x \in \{0, \ldots, p-1\}$. [8]

5.3.2 Vorteil

Die El-Gamal-Verschlüsselung kann nicht nur in primen Restklassengruppen angewandt werden, sondern auch in jeder zyklischen Gruppe. Dies ist eines der größten Vorteile dieses Verfahrens. Jedoch ist nicht jede zyklische Gruppe gleich gut geeignet. Damit die Gruppe gut geeignet ist, muss darauf geachtet werden, dass das Diffie-Hellman-Problem nicht bzw. schwierig lösbar ist. Jedoch muss dabei die Schlüsselerzeugung, Verschlüsselung und Entschlüsselung effizient berechenbar bleiben.

Da weiterhin an der Lösung des diskreten Logarithmus Problems gearbeitet wird, kann jederzeit ein Algorithmus entdeckt werden, der die Berechnung des diskreten Logarithmus in einer bestimmten Gruppe effizient möglich macht. Daher muss das El-Gamal-Verfahren allgemein bleiben. Dies bedeutet, dass es in verschiedenen Gruppen anwendbar ist.

Die folgenden Gruppen sind Beispiele für Gruppen, die sich besonders gut für das El-Gamal-Verfahren eignen.

- Die Punktgruppe einer endlichen Kurve über einem endlichen Körper
- Die Jakobische Varietät hyperelliptischer Kurven über endlichen Körpern
- Die Klassengruppe imaginär-quadratischer Ordnungen

Bei jeder erneuten Versendung einer Nachricht, wird diese erneut verschlüsselt. Auch dies ist ein Vorteil der El-Gamal-Verschlüsselung.

5.3.3 Anwendung in der Schule

Durch die Problemstellung des diskreten Logarithmus und das weitere Verfahren, ist das El-Gamal-Verfahren sehr komplex und sollte daher im regulären Mathematikunterricht nicht thematisiert werden. Das Diffie-Hellman-Verfahren ist für Schüler:innen einfacher

zu verstehen und kann daher mit den Schüler:innen erarbeitet werden. Vorausgesetzte Kenntnisse für die Berechnung des Diffi-Hellman-Verfahren sind

- Potenzieren
- Berechnung von Modulo

Jedoch muss die Parameterwahl gut durchdacht sein, da auch bei kleinen Zahlen die Potenzen sehr groß werden können. Werden die Parameter jedoch zu passend gewählt, verliert das Diffie-Hellman-Verfahren an Komplexität. Um dies zu umgehen, sollte dieses Verfahren mit einem geeigneten CAS-Programm berechnet werden.
[7]

5.4 Rabin-Verschlüsselung

Dieses Verfahren ist verwandt mit der RSA-Verschlüsselung und beruht auf dem Faktorisierungsproblem.

Möchte Bob eine Nachricht m verschlüsseln, benötigt er den öffentlichen Schlüssel n von Alice. Dieser wird wie folgt berechnet.

Alice wählt zwei zufällige Zahlen $p, q \in \mathbf{P} : p \equiv q \equiv 3 \bmod 4$ und berechnet nun ihren öffentlichen Schlüssel $n = pq$. Dabei soll n eine k-Bit-Zahl sein. k ist der Sicherheitsparameter. Alice privater Schlüssel ist das Tupel (p, q).

Nun kann Bob die Nachricht m verschlüsseln, indem er den Schlüsseltext c (die verschlüsselte Nachricht) berechnet. Der Klartextraum ist die Menge $\{0, \ldots, n-1\}$.

$$c = m^2 \bmod n$$

Damit Alice die Nachricht entschlüsseln und somit lesen kann, berechnet sie Folgendes:

$$m_p = c^{(p+1)/4} \bmod p$$
$$m_q = c^{(q+1)/4} \bmod q$$

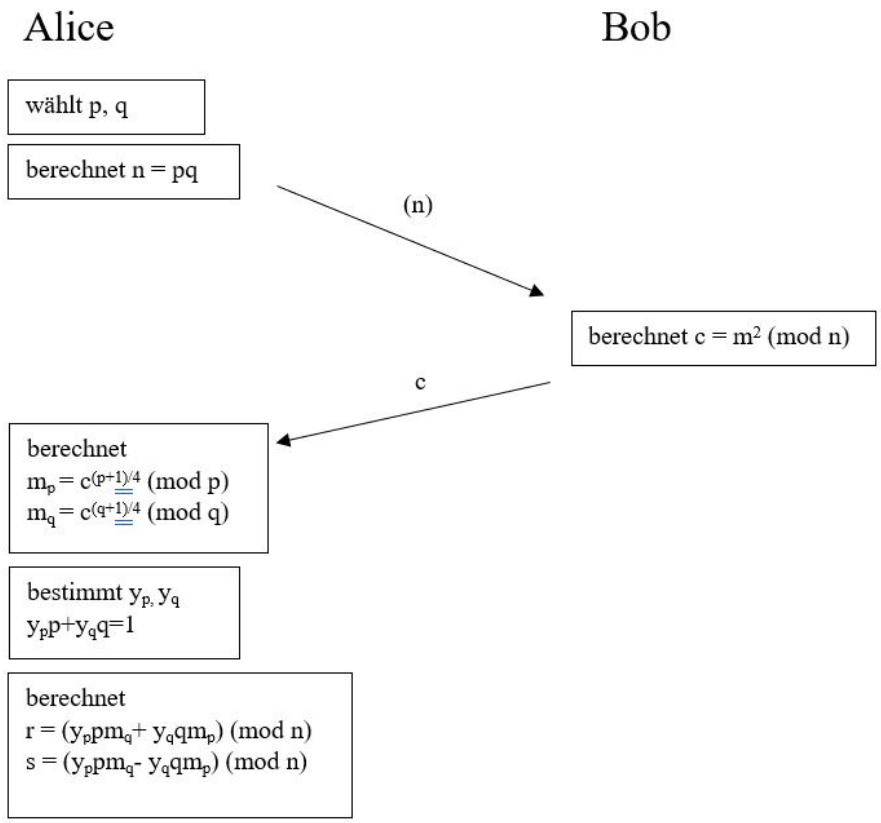
Somit sind $\pm m_p + p\mathbb{Z}$ die beiden Quadratwurzeln von $c + p\mathbb{Z}$ in $\mathbb{Z}/p\mathbb{Z}$ und $\pm m_q + q\mathbb{Z}$ die beiden Quadratwurzeln von $c + q\mathbf{Z}$ in $\mathbf{Z}/q\mathbb{Z}$. Die vier Quadratwurzeln von $c + n\mathbb{Z}$ in $\mathbb{Z}/n\mathbb{Z}$ kann Alice mit Hilfe des chinesischen Restsatzes berechnen. Alice bestimmt mit dem erweiterten euklidischen Algorithmus die Koeffizienten $y_p, y_q \in \mathbb{Z}$ mit

$$y_p p + y_q q = 1.$$

Danach berechnet sie:

$$r = (y_p p m_q + y_q q m_p) \bmod n$$
$$s = (y_p p m_q - y_q q m_p) \bmod n$$

Daraus folgt, dass $\pm r$ und $\pm s$ die Quadratwurzeln von c sind. Eine dieser Quadratwurzeln muss die Nachricht m sein.
Die folgende Grafik gibt einen bildlichen Überblick über den Verlauf der Rabin-Verschlüsselung. [8]



Um das Rabin-Verfahren etwas anschaulicher zu erklären, wird nun ein Beispiel mit kleinen Zahlen vorgerechnet

Beispiel. Alice wählt die Primzahlen $p = 11$ und $q = 23$. Sie berechnet nun:

$$\begin{aligned} n &= pq \\ n &= 11 \cdot 23 \\ n &= 253. \end{aligned}$$

Bob möchte den Klartext $m = 158$ an Alice senden. Zur Verschlüsselung berechnet er:

$$\begin{aligned} c &= m^2 \mod n \\ c &= 158^2 \mod 253 \\ c &= 170. \end{aligned}$$

Mit dem erweiterten euklidischen Algorithmus berechnet Alice die Koeffizienten $y_p = -2$

und $y_q = 1$. Im nächsten Schritt ermittelt Alice die Quadratwurzeln.

$$\begin{aligned} m_p &= c^{\frac{p+1}{4}} \mod p \\ &= 170^{\frac{11+1}{4}} \mod 11 \\ &= 170^3 \mod 11 \\ &= 4 \\ m_q &= c^{\frac{q+1}{4}} \mod q \\ &= 170^{\frac{23+1}{4}} \mod 23 \\ &= 170^6 \mod 23 \\ &= 3 \end{aligned}$$

Danach bestimmt Alice r und s.

$$\begin{aligned} r &= (y_p p m_q + y_q q m_p) \mod n \\ &= (-2) \cdot 11 \cdot 3 + 23 \cdot 4 \mod 253 \\ &= 26 \end{aligned}$$

$$\begin{aligned} s &= (y_p p m_q - y_q q m_p) \mod n \\ &= (-2) \cdot 11 \cdot 3 - 23 \cdot 4 \mod 253 \\ &= 95 \end{aligned}$$

Somit sind die Quadratwurzeln $26, 95, 158, 227$. Daraus kann Alice den Klartext 158 auswählen.
[8]
Beispiel. Ein weiteres Beispiel für die Rabin-Verschlüsselung lautet wie folgt:
Alice wählt als Primzahlen $p = 131$ und $q = 139$. Daraus berechnet sie

$$\begin{aligned} n &= pq \\ n &= 131 \cdot 139 \\ n &= 18209. \end{aligned}$$

Will Bob die Nachricht $m = 4273$ an Alice senden, muss er zur Verschlüsselung Folgendes berechnen:

$$\begin{aligned} c &= m^2 \mod n \\ c &= 4273^2 \mod 18209 \\ c &= 13111. \end{aligned}$$

Nun kann Bob die verschlüsselte Nachricht $c = 13111$ an Alice senden. Um Bobs Nachricht zu entschlüsseln, berechnet Alice mit dem erweiterten euklidischen Algorithmus die Koeffizienten $y_p = 52$ und $y_q = -49$. Nun muss Alice die Quadratwurzeln berechnen.

$$\begin{aligned} m_p &= c^{\frac{p+1}{4}} \mod p \\ &= 13111^{\frac{131+1}{4}} \mod 131 \\ &= 13111^{33} \mod 131 \\ &= 81 \\ m_q &= c^{\frac{q+1}{4}} \mod q \\ &= 13111^{\frac{139+1}{4}} \mod 139 \\ &= 170^{35} \mod 139 \\ &= 36 \end{aligned}$$

Danach bestimmt Alice r und s.

$$\begin{aligned} r &= (y_p p m_q + y_q q m_p) \mod n \\ &= 52 \cdot 131 \cdot 36 + (-49) \cdot 139 \cdot 81 \mod 18209 \\ &= 3094 \end{aligned}$$

$$\begin{aligned} s &= (y_p p m_q - y_q q m_p) \mod n \\ &= 52 \cdot 131 \cdot 36 - (-49) \cdot 139 \cdot 81 \mod 18209 \\ &= 13936 \end{aligned}$$

Damit ergeben sich für die Quadratwurzeln 3094, 13936, 4273 und 15115. Alice kann nun den richtigen Klartext auswählen.

Beispiel. Ein weiteres einfaches Beispiel für die Rabin-Verschlüsselung. Alice wählt als Primzahlen $p = 7$ und $q = 11$. Daraus berechnet sie nun den öffentlichen Schlüssel n.

$$\begin{aligned} n &= pq \\ n &= 11 \cdot 7 \\ n &= 77. \end{aligned}$$

Die Zahlen 7 und 11 können gewählt werden, da sie die Kongruenzbedingung

$$p \equiv q \equiv 3 \mod 4$$

erfüllen, da

$$7 \equiv 11 \equiv 3 \mod 4.$$

Bob möchte nun den Klartext $m = 20$ an Alice senden. Dazu berechnet er den Geheimtext

$$\begin{aligned} c &= m^2 \mod n \\ c &= 20^2 \mod 77 \\ c &\equiv 15 \end{aligned}$$

Es existieren genau vier verschiedene m für die denselben Geheimtext $c = 15$ haben.
Bob sendet die verschlüsselte Nachricht $c = 15$ an Alice senden. Um Bobs Nachricht zu entschlüsseln, berechnet Alice mit dem erweiterten euklidischen Algorithmus die Koeffizienten y_p und y_q.

$$\begin{aligned} 7 &= 0 \cdot 11 + 7 \\ 11 &= 1 \cdot 7 + 4 \\ 7 &= 1 \cdot 4 + 3 \\ 4 &= 1 \cdot 3 + 1 \\ 3 &= 3 \cdot 1 + 0 \end{aligned}$$

Durch Zurückrechnen ergibt sich

$$\begin{aligned} 1 &= 4 - 1 \cdot 3 \\ &= 4 - 1 \cdot (7 - 1 \cdot 4) \\ &= -1 \cdot 7 - 2 \cdot 4 \\ &= -1 \cdot 7 + 2 \cdot (11 - 1 \cdot 7) \\ &= 2 \cdot 11 - 3 \cdot 7 \\ &= 2 \cdot 11 - 3 \cdot (7 - 0 \cdot 11) \\ &= -3 \cdot 7 + 2 \cdot 11 \end{aligned}$$

Daher ist $y_p = -3$ und $y_q = 2$
Nun muss Alice die Quadratwurzeln berechnen.

$$\begin{aligned} m_p &= c^{\frac{p+1}{4}} \mod p \\ &= 15^{\frac{7+1}{4}} \mod 7 \\ &= 15^2 \mod 7 \\ &= 1 \\ m_q &= c^{\frac{q+1}{4}} \mod q \\ &= 15^{\frac{11+1}{4}} \mod 11 \\ &= 15^3 \mod 11 \\ &= 9 \end{aligned}$$

Danach bestimmt Alice r und s.

$$\begin{aligned} r &= (y_p p m_q + y_q q m_p) \mod n \\ &= -3 \cdot 7 \cdot 9 + 2 \cdot 11 \cdot 1 \mod 77 \\ &= 64 \end{aligned}$$

$$\begin{aligned} s &= (y_p p m_q - y_q q m_p) \mod n \\ &= -3 \cdot 7 \cdot 9 - 2 \cdot 11 \cdot 1 \mod 77 \\ &= 20 \end{aligned}$$

Die anderen Quadratwurzeln ergeben sich aus

$$\begin{aligned} -r &= n - r \\ 13 &= 77 - 64 \\ -s &= n - s \\ 57 &= 77 - 57. \end{aligned}$$

Damit ergeben sich für die Quadratwurzeln 64, 20, 13 und 57. Alice kann nun den richtigen Klartext auswählen.

Um die richtige Quadratwurzel auszuwählen und somit den Klartext zu bestimmen, gibt es verschiedene Möglichkeiten. Eine Methode wäre, dass Alice die Quadratwurzel auswählt, von der sie glaubt, dass sie die wahrscheinlichste Lösung ist. Dies ist aber nicht die sicherste Variante, da mehrere Lösungen passen könnten.
Daher gibt es auch die Möglichkeit, dem Klartext eine gewisse Struktur zu geben. Wenn dies der Fall ist, kann Alice die Nachricht auswählen, die der speziellen Struktur entspricht. Ein Beispiel dieser Struktur wäre, wenn die letzten 64 Bit des Klartextes den vorletzten 64 Bit entsprechen.

Die Rabin-Verschlüsselung ist effizienter als die RSA-Verschlüsselung, selbst mit Exponenten 3, da das Rabin-Verfahren nur eine Quadrierung modulo n benötigt. Die Entschlüsselung ist gleich effizient wie bei der RSA-Verschlüsselung.

In der Praxis werden Primzahlen gewählt die mindestens eine Größe von 10^{200} haben. Da dies die Berechnung um einiges komplexer macht wurden für die Beispiele deutlich kleinere Zahlen verwendet.

6 Sicherheit

Nachrichten und Daten werden verschlüsselt, damit sie nicht in Hände Dritter gelangen. Jedoch gibt es Angreifer, die diese Informationen erhalten wollen. In der Kryptografie werden die Angreifer oft mit „Eve“ bezeichnet. Es gibt verschiedene Arten, wie Angreifer an Daten kommen können. Dabei spielt es eine Rolle, was das Ziel und was die Mittel der Angreifer sind. Das Angriffsziel kann entweder der Schlüssel oder die Geheimnachricht sein. Mit dem Wissen des Schlüssels kann der Angreifer alle verschlüsselte Nachrichten entschlüsseln.

6.1 Angriffstypen

6.1.1 Ciphertext-Only-Angriff

Wie der Name bereits vermuten lässt, hat der Angreifer bei einem Ciphertext-Only-Angriff nur Kenntnis über den Chiffretext. Mithilfe dieser Information versucht der Angreifer den Klartext zu entziffern oder den verwendeten Schlüssel herauszufinden.
Dabei werden alle möglichen Schlüssel des Schlüsselraumes ausprobiert und der Chiffretext mit jedem dieser Schlüssel entschlüsselt. Dadurch entstehen viele Klartexte, jedoch sind die wenigsten sinnvoll. Mithilfe des Kontextes kann der Angreifer so die richtige Nachricht herausfinden. Diese Methode funktioniert jedoch nur, wenn der Schlüsselraum klein genug ist. Eine Verschlüsselung bei der ein Ciphertext-Only-Angriff einfach durchzuführen ist, ist die Cäsar-Verschlüsselung, da der Schlüsselraum lediglich 26 Elemente besitzt.

Aber auch bei moderneren Verschlüsselungsverfahren, wie beim DES-Algorithmus, funktioniert ein Ciphertext-Only-Angriff. Der Schlüsselraum des DES besitzt die Größe 2^{56}. 1998 wurde der DES-Algorithmus von der Electronic Frontier Foundaitin binnen 56 Stunden mithilfe des Ciphertext-Only-Angriffes entschlüsselt. Mittlerweile kann der DES-Algorithmus von jedem handelsüblichen PC gebrochen werden.

Laut Moorsche'm Gesetz verdoppelt sich die Rechenleistung eines Computer alle 18 Monate. Daher wird es immer wichtiger, einen entprechend großen Schlüsselraum zu verwenden. Die Mindestgröße des Schlüsselraums in den nächsten Jahren kann der folgenden Grafik entnommen werden.

Schutz bis	Mindestgröße
2020	2^{96}
2030	2^{112}
2040	2^{128}
für absehbare Zukunft	2^{256}

Abbildung 6.1: Mindestgröße des Schlüsselraums

Eine andere Art des Ciphertext-Only-Angriffs ist das zu Nutzen machen der statistischen Häufigkeiten der einzelnen Buchstaben einer Sprache. Bei einer Verschiebungschiffre wird jedes Klartextzeichen immer durch dasselbe Schlüsseltextzeichen ersetzt. Somit entspricht das häufigste Zeichen des Schlüsseltextes dem häufigsten Zeichen des Klartextes.

6.1.2 Known-Plaintext-Angriff

Bei einem Known-Plaintext-Angriff verfügt der Angreifer sowohl über Wissen betreffend den Klartext und den Chiffretext. Dieses Wissen macht sich der Angreifer zu nutzen um Informationen über die Verschlüsselungsmethode und den Schlüssel zu erhalten.

Ein bekanntes Beispiel für einen Known-Plaintext-Angriff fand im zweiten Weltkrieg statt. Deutschland verwendete zum Verschlüsseln die Enigma-Maschine. Die Nachrichten die gesendet wurden, hatten meist denselben Aufbau. Weiters wurde der Wetterbericht jeden Morgen zur selben Zeit verschlüsselt gesendet. Da der Wetterbericht jedoch auch öffentlich war, konnten so die Klartexte zu den Chiffretexten zugeordnet werden.

6.1.3 Chosen-Plaintext-Angriff

Bei einem Chosen-Plaintext-Angriff besitzt der Angreifer keine Kenntnis über den Schlüssel. Jedoch hat der Angreifer die Möglichkeit, verschlüsselte Nachrichten zu senden, bei denen er auch den Klartext kennt. Mithilfe des Klartextes und des Chiffretextes kann der Angreifer den geheimen Schlüssel bestimmen. Diese Art der Angriffe ist gerade bei determinitischen Public-Key-Verfahren gefährlich.

6.1.4 Chosen-Ciphertext-Angriff

Bei einem Chosen-Ciphertext-Angriff besitzt der Angreifer Kenntnise über den Schlüssel und kann so verschlüsselte Nachrichten in Klartexte verwandeln. [8]

7 Schluss

Der Schluss dieser Arbeit fasst die wichtigsten Erkenntnisse der verschiedenen Verschlüsselungsverfahren zusammen. Verschlüsselungsverfahren haben eine enorme Entwicklung durchgemacht. Da die ersten Verschlüsselungsmehtoden bereits in den frühen Hochkulturen entwickelt wurden, war es damals wichtig, dass die Verschlüsselung leicht zu berechnen war. Die historischen Verschlüsselungen sind daher symmetrische Verschlüsselungen. Es war daher umso wichtiger, den Schlüssel vor Fremden geheim zu halten.

Die Skytale zählt zu den ältesten Verschlüsselungsmehtoden und beruht auf dem Prinzip der Permutation. Dabei wird ein Klartext auf eine Schriftrolle geschrieben, welche um einen Stab mit bestimmtem Durchmesser gewickelt wird. Dabei ist der Durchmesser der Schlüssel. Die Skytale zählt zu den Transpositionsverfahren. Eine weitere Verschlüsselung, welche ein Transpositionsverfahren ist, ist die Fleissner-Schablone. Die Fleissner-Schablone ist weiter entwickelt, als die Skytale. Sie baut ebenfalls auf dem Prinzip der Permutation auf. Weiters kann die Schablone als eine Form linearer Transformation aufgefasst werden. Die Analyse der möglichen Permutationen kann mithilfe der Kombinatorik durchgeführt werden.

Neben der Transpositionverfahren, teilt sich die symmetrische Verschlüsselung in die Substitutionsverfahren. Dazu zählt die Cäsar-Verschlüsselung. Sie ist zwar eine einfache Methode der Verschlüsselung, dient aber als Ausgangspunkt für weitere Verschlüsselungsverfahren. Die Cäsar-Verschlüsselung ist eine Verschiebungschiffre. Mathematische Voraussetzung für dieses Verfahren ist die modulo Rechnung. Die modulo-Operation findet in verschiedenen Bereichen der Mathematik Anwendung. Ein weiterer Vertreter des Substitutionsverfahrens ist die Vignère-Verschlüsselung. Die Vignère-Verschlüsselung zählt ebenfalls zu den Verschiebechiffres. Dazu wird Verschiebung öfters hintereinander durchgeführt. Daher ist ebenfalls die modulo-Operation ein wichtiger Bestandteil.

Moderne symmetrische Verschlüsselung ist die Feistelchiffre, AES und DES. Bei der Feistelchiffre wird der zu verschlüsselnde Text über mehrere Runden hinweg mit einer bestimmten Funktion verschlüsselt. Die Entschlüsselung erfolgt in umgekehrter Reihenfolge. Der AES wird ebenfalls über mehrere Runden hinweg verschlüsselt. Mithilfe von S-Boxen werden mehrere Permutationen hintereinander durchgeführt. Der DES-Algorithmus wendet die Feistel-Chiffre an und verwendet daher das Prinzip der Permutation und der Substitution.

Das Diffie-Hellman-Verfahren bietet eine Möglichkeit einen Schlüssel sicher auszutauschen. Es macht sich das Problem des diskreten-Logarithmus zu Nutzen. Eine Verschlüsselungs-

methode, die ebenfalls auf dieser Problematik aufbaut, ist die El-Gamal-Verschlüsselung. Solange der diskrete Logarithmus nicht effetkiv berechnet werden kann, ist sowohl das Diffie-Hellman-Verfahren als auch die El-Gamal-Verschlüsselung sicher. Die RSA-Verschlüsselung beruht auf dem Schwierigkeitsgrad des Fakorisierungsproblems, während das Rabin-Verfahren auf den mathematischen Prinzipien der quadratischen Restklassen und der Eigenschaften von Quadraten in der modularen Arithmetik basiert.

Zusammengefasst kann gesagt werden, dass eine symmetrische Verschlüsselung solange sicher ist, solange der Schlüssel geheim ist. Gibt es jedoch zu wenige Möglichkeiten, kann die Entschlüsselung durch alleiniges Ausprobieren durchgeführt werden. Bei der asymmetrischen Verschlüsselung ist es wichtig, dass das Verfahren nur in eine Richtung effizient berechenbar ist und die Rückrechnung sehr schwer und ineffizient ist.

Literaturverzeichnis

[1] Hintergrundwissen zur kryptografie. Eingesehen am 15.04.2021.

[2] Dr. Reimund Albers. Stellenwerte. Vorlesungsskript.

[3] Peter Reichl Andreas Janecek. Technische grundlagen und systemsoftware. Vorlesungsskript.

[4] Peter Baeumle-Courth, Stefan Nieland, and Hinrich Schröder. Symmetrische verfahren (private key). In *Wirtschaftsinformatik*. Walter de Gruyter GmbH, United States, 2004.

[5] A. Beutelspacher, H.B. Neumann, and T. Schwarzpaul. *Kryptografie in Theorie und Praxis: Mathematische Grundlagen für Internetsicherheit, Mobilfunk und elektronisches Geld*. Vieweg Studium. Vieweg+Teubner Verlag, 2010.

[6] Albrecht Beutelspacher. *Kryptologie : Eine Einführung in die Wissenschaft vom Verschlüsseln, Verbergen und Verheimlichen*. Springer Fachmedien Wiesbaden, Wiesbaden, 10., aktualisierte aufl. 2015 edition, 2015.

[7] Thomas Borys. *Codierung und Kryptologie : Facetten einer anwendungsorientierten Mathematik im Bildungsprozess /*. Vieweg+Teubner Research. Vieweg+Teubner Verlag : Imprint: Vieweg+Teubner Verlag„ Wiesbaden :, 1st ed. 2011. edition, 2011.

[8] Johannes Buchmann. *Einführung in die Kryptographie*. Springer-Lehrbuch. Springer Berlin Heidelberg, Berlin, Heidelberg, 6., überarb. aufl. 2016 edition.

[9] Sebastian Dworatschek. *2. Zahlensysteme*, pages 88–129. De Gruyter, Berlin, Boston, 1969.

[10] Claudia Eckert. *IT-Sicherheit, 10th Edition*. De Gruyter Oldenbourg, 2018.

[11] Christoph Engemann. Exkurs: Asymmetrische verschlüsselung und die digitale signatur. In *Electronic Government - vom User zum Bürger*, pages 75–86. transcript Verlag, Bielefeld, 2015.

[12] Roland Hellmann. Symmetrische verschlüsselung. In *IT-Sicherheit*, pages 7–32. Walter de Gruyter GmbH, Germany, 2022.

[13] Gernot Koller. ausgewählte primzahltests. 2015.

[14] Stefan Krauss. Quod erat docendum? In *Quod erat knobelandum*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[15] Burkhard Lenze. *Basiswissen Angewandte Mathematik – Numerik, Grafik, Kryptik: Eine Einführung mit Aufgaben, Lösungen, Selbsttests und interaktivem Online-Tool.* Springer Fachmedien Wiesbaden, Wiesbaden, 2. aufl. 2020 edition.

[16] Olaf Manz. *Verschlüsseln, Signieren, Angreifen : Eine kompakte Einführung in die Kryptografie.* Springer Berlin Heidelberg Imprint: Springer Spektrum, Berlin Heidelberg, 1st ed. 2019 edition, 2019.

[17] W. Scherer. *Mathematik der Quanteninformatik: Eine Einführung.* Springer Berlin Heidelberg, 2016.

[18] H. Schichl and R. Steinbauer. *Einführung in das mathematische Arbeiten.* Springer Berlin Heidelberg, 2018.

[19] Fleissner von Wostrowitz. *Handbuch der Kryptographie- Anleitung zum Chiffrieren und Dechiffrieren von Geheimschriften.* K. k. Hofbuchdruckerei Carl, Wien, 1881.

[20] K.U. Witt. *Algebraische und zahlentheoretische Grundlagen für die Informatik: Gruppen, Ringe, Körper, Primzahltests, Verschlüsselung.* SpringerLink : Bücher. Springer Fachmedien Wiesbaden, 2014.

[21] Stefan Witzel. Elementare geometrie. Vorlesungsskript.