Universität Wien, Fakultät für Mathematik

Die RSA-Verschlüsselung

Author:
Attila Keszeg

Supervisor:
Dr. Markus Fulmek

13. März 2016

Inhaltsverzeichnis

Kapitel 1

Kryptographie

In diesem Abschnitt wollen wir uns mit der *geheimen Kommunikation* beschäftigen, die sich in zwei von einander komplett unabhängige Disziplinen unterteilen lässt nämlich in die **Steganographie** und in die **Kryptographie**.

1.1 Steganographie

Verwendet man Steganographie zur Übermittlung von geheimen Texten , so versucht man die Botschaft so zu verdecken, dass ihre Existenz verborgen bleibt. Diese Art der Nachrichtenübermittlung reicht bis in die Antike zurück.
"Verschiedenste Quellen aus Griechenland, Persien und China berichten, wie Steganographie zur Übermittlung strategischer Pläne verwendet wurde. Da wir in dieser Arbeit den Schwerpunkt nicht auf die Steganographie legen wollen, geben wir nur ein Paar Beispiele, damit sich die LeserInnen dieses Artikels vorstellen können,was Steganographie bedeutet:

1. In der Antike verwendeten die Griechen Boten, deren Kopf sie rasierten und auf diesen eine Nachricht einbrannten oder tätowierten. Nachdem das Haar nachgewachsen war, schickte man sie zur Zielperson

2. In einem anderem Fall wurde von einer Schreibtafel berichtet, bei der das Wachs heruntergekratzt, die Mitteilung in das Holz geschrieben und später mit Wachs wieder überzogen wurde.

3. Die Chinesen schrieben ihre Botschaften auf sehr dünnes Papier oder Seide, rollten diese ein und tauchten sie in Wachs. Die Boten konnten die Kugeln in ihrer Kleidung verstecken oder verschluckten sie gar.

4. Im 1. Jahrhundert verwendeten die Römer die sogenannte Thithymallus- Pflanze um aus ihrem Saft eine unsichtbare Tinte herzustellen." [6]

1.2 Kryptographie

Die Wissenschaft der Verheimlichung von Nachrichten heißt Kryptographie[1]. Dieses Wort hört sich zwar recht altmodisch an, spielt aber in unserem Alltag eine ganz wichtige Rolle.Sie wird gebraucht um Informationen, wie zum Beispiel Chipkarten und Passwörter zu ver- bzw. entschlüsseln.
Wir können Verschlüsselungsverfahren in **Transpositionen** und **Substitutionen** unterteilen. Bei der Substitution müssen wir aber wieder eine Unterteilung machen. Die zwei wesentlichsten Arten der Substitution sind die **Codierung** und die **Chiffrierung**. Bei der Chiffrierung werden nur Buchstaben durch andere Buchstaben beziehungsweise Zahlen ersetzt, während bei der Codierung ganze Wörter ersetzt werden können.

[1] griechisch: geheimes Schreiben

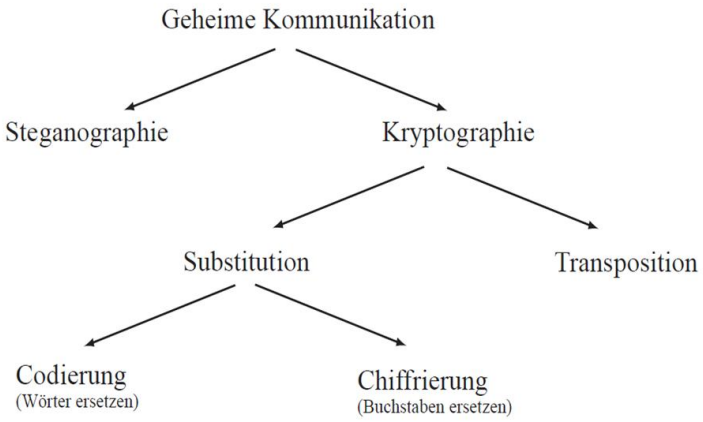

Abbildung 1.1: Geheime Kommunikation

In der Fachliteratur finden wir auch eine Unterteilung nach der Art der Verschlüsselung. Wir unterscheiden dabei zwischen symmetrischer und assymetrischer Verschlüsselung. Bei der symmetrischen Verschlüsselung vereinbaren die Kommunikationspartner einen gemeinsamen Schlüssel,dessen Übergabe vor einer geheimen Kommunikation stattfinden muss.

1.2.1 Transposition

Bei der Transposition werden Buchstaben eines Wortes/Satzes ersetzt. Dabei hat man immer $\boldsymbol{n!}$ viele Möglichkeiten, wobei $\boldsymbol{n}$ die Anzahl der Buchstaben im Wort/Satz angibt. Betrachten wir das Wort **Kommunikation**. Dieses besteht aus 13 Buchstaben, kann also auf $\mathbf{13! = 6227020800}$ verschiedene Arten Verschlüsselt werden. Der Leser/ die Leserin könnte sich denken, diese Methode wäre aufgrund der hohen Anzahl an Kombinationsmöglichkeiten sehr sicher. Das ‚Problem dabei ist, dass der Absender einer Nachricht nicht jede beliebige Permutation verwenden kann, da der Empfänger mit einer zufällig angeordneten Buchstabenreihe nicht wirklich was anfangen kann. Er kann natürlich mal Glück haben und erkennen, was die Botschaft ist. Das ist aber eher unwahrscheinlich. Das heißt, hinter jeder Vertauschung von Buchstaben muss eine im Vornhinein festgelegte Handlungsvorschrift (=Algorithmus) stehen. Nur so hat der Empfänger der Nachricht die Chance die Nachricht lesen zu können. Eine der ältesten Transpositionsmethoden wurde im 5. Jahrhundert vor Christus von den Spartianern benutzt. Die von ihnen verwendete Methode *(Skytale)* werden wir erst im nächsten Kapitel vorstellen.

1.2.2 Substitution

Bei dem Substitutionsverfahren wird der Klartext durch Austauschen der Buchstaben mit anderen Buchstaben, Ziffern oder Zeichen (Homophon) verschlüsselt. Ein klassisches Beispiel zur Substitution ist die Cäsar-Verschlüsselung, die wir ebenfalls im nächsten Abschnitt vorstellen wollen. Wie wir das sehen werden, es handelt sich bei der Cäsar Verschlüsselung um iene Verschiebung der Buchstaben des Alphabets. Verwenden wir ein Alphabet mit 26 Buchstaben ‚so muss derjenige, der unseren Code Knacken will, nur 25 verschiedene Verschiebungen testen. Wir können also einsehen, dass Verschlüsselung durch Verschiebung zu einer sehr unsicheren Verschlüsselung führt. Wir müssen uns also überlegen, wie dieses Problem gelöst werden könnte. Die am einfachsten erscheinende Idee wäre, dass wir ein allgemeines Geheimtextalphabet herstellen und dadurch die sogenannte monoalphabetische Substitution verwenden. Dafür gibt es bei einem Alphabet mit 26 Buchstaben genau $\mathbf{26!}$ Möglichkeiten, was eine enorm große Zahl ist in der Größenordnung von $\mathbf{4 \cdot 10^{26}}$. Bei so vielen Möglichkeiten könnten wir uns denken, wir hätten eine absolut sichere Methode gefunden. In der Wirklichkeit ist das aber nicht so. Mit der Methode der **Häufigkeitsanalyse** können wie oben verschlüsselte Nachrich-

ten ziemlich problemlos geknackt werden. Bei diesem Verfahren geht es nicht nur darum den Inhalt einer Nachricht zu studieren, sondern auch um die Häufigkeit der im Text auftauchenden Buchstaben, Wörter, Sätze. Im arabischen Raum hat man diese Methode bereits ab dem neunten Jahrhundert nach Christus zur Entschlüsselung von Nachrichten verwendet.
In allen Sprachen der Welt kommen alle Buchstaben mit einer bestimmten Häufigkeit vor.

Folgende Tabelle enthält die Häufigkeitsverteilung der einzelnen Buchstaben der deutschen Sprache:

A	**6,51%**	**F**	**1,66%**	**K**	**1,21%**	**P**	**0,79%**	**U**	**4,35%**	**Z**	**1,13%**
B	**1,89%**	**G**	**3,01%**	**L**	**3,44%**	**Q**	**0,02%**	**V**	**0,67%**		
C	**3,06%**	**H**	**4,76%**	**M**	**2,53%**	**R**	**7%**	**W**	**1,89%**		
D	**5,08%**	**I**	**7,55%**	**N**	**9,78%**	**S**	**7,27%**	**X**	**0,03%**		
E	**17,4%**	**J**	**0,27%**	**O**	**2,51%**	**T**	**6,15%**	**Y**	**0,04%**		

Eine der bekanntesten Substitutionsmethoden wurde im Jahr 1585 von **Blaise de Vigenére**[2]. In seinem Werk **Traicté des Chiffres** beschreibt er als erster eine polyalphabetische Verschlüsselung, die bis heute noch seinen Namen trägt. Die Vignére- Methode werden wir in diesem Artikel nocht vorstellen.

Bemerkung 1. *Sowohl Transpositionen als auch Substitutionen gehören zu den symmetrischen Verschlüsselungsverfahren. Bei diesen arbeiten die Kommunikationspartner mit einem einzigen Schlüssel, der bei der Ver- und Entschlüsselung vorhanden sein muss. Diese Verfahren sind zwar sehr schnell bieten aber nur dann Sicherheit , wenn der Schlüssel lang genug ist. Schwierig ist jedoch der Schlüsselaustausch. Die Datenübertragung erfolgt erst nachdem sich Kommunikationspartner auf einen gemeinsmen Schlüssel geeinigt haben und diesen auch ausgetauscht haben. Wenn der Schlüssel den selben Kommunikationsweg nimmt, wie die anschließend verschlüsselte Nachricht, besteht die Gefahr, dass ein Gegner beim Abhören der Kommunikation in den Besitzt des Schlüssels gelangt. Mit diesem könnte er natürlich die Botschaften entschlüsseln und somit wichtige Informationen erhalten. In einem einzigen Fall ist die Schlüsselübergabe sicher, nämlich wenn sich die Kommunikationspartner den Schlüssel persönlich austauschen, oder der Schlüssel auf einem Nebenkanal gesendet wird.*

1.2.3 Asymetrische Verschlüsselung

Im Gegensatz zur symmetrische Verschlüsselung arbeitet die asymetrische Verschlüsselung mit zwei Schlüsseln. Diese hat den Vorteil dass die Kommunikationspartner im Vornhinein den Schlüssel nicht austauschen müssen, wie das bei der symmetrischen Verschlüsselung der Fall ist.
Wollen zwei Kommunikationspartner ihre Botschaften untereinander austauschen, wobei die Botschaft asymmetrisch verschlüsselt ist, so muss der Empfänger der Nachricht einen für jeden zugänglichen öffentlichen Schlüssel ***Public-Key*** zu Verfügung stellen. Die Nachricht wird vom Absender mit diesem öffentlichen Schlüssel verschlüsselt und an den Empfänger geschickt. Der Empfänger kann diese Nachricht nur mit seinem anderen, privaten Schlüsse ***Private-Key*** entschlüsseln. Die Grundlagen der asymmetrische Verschlüsselung gehen auf Diffie und Hellman zurück. Zudem siehe [2] und [6]

Für genauere Beschreibung siehe [2] und [3]

1.3 Historischer Überblick

In diesem Abschnitt wollen wir einen kurzen, tabellarischen Überblick der wichtigsten Meilensteine der Kryptographie geben.

600 v.Chr.	Verschlüsselung mit **ATBaS**

[2] Blaise de Vigenére 1523 - 1596), französischer Diplomat und Kryptograph.

Abbildung 1.2: Skytale

ATBaS ist die erste bekannte Verschlüsselungsmethode. Sie beruht auf dem hebräischen Alphabet und ihr Name zeigt uns, wie man bei der Verschlüsselung fortgehen muss. Der erste Buchstabe *(Aleph)* Alphabets muss mit dem letzten Buchstaben *(Taw)* vertauscht werden , der zweite Buchstabe *(Beth)* mit dem vorletzten Buchstaben *(Sin)* usw.

500 v.Chr.	In Sparta werden militärische Nachrichten mit der **Skytale** *(siehe Abbildung 1.1.)* verschlüsselt

Das Wort **Skytale** (griech.: $\sigma\kappa\upsilon\tau\alpha\lambda\eta$) bedeutet Stab. Wollte man Nachrichten verfassen, so wickelte der Absender ein Pergamentband wendelförmig um einen Holzstab mit einem bestimmten Durchmesser , schrieb die Nachricht auf das Band *(längs des Stabs)* und wickelte es dann ab. Dem Empfänger wurde der Stab nicht überbracht. Der Empfänger konnte die Botschaft nur dann lesen, wenn er einer identischen Skytale im Besitze war.

Beispiel: Das ist ein Beispiel
Wir verschlüsseln diesen Text mit einer Skytale des Umfangs $U = 4$, indem wir den Text in 4 Spalten aufteilen

d	a	s	i
s	t	e	i
n	b	e	i
s	p	i	e
e			

Somit erhalten wir folgenden Text:

dsnseatbpseeiiiie

Dieser Art der Transposition wir Spaltentransposition genannt

100 v.Chr.	Der Römische Feldherr Julius Caesar*(100 v.Chr-44 v.Chr)* verschlüsselt Nachrichten mit dem nach ihm benannten **Caesar-Code**

Caesar-Code : Jede Buchstabe des Alphabets wird um eine bestimmte Anzahl an Positionen verschoben, das heißt das Grundprinzip ist eine Alphabetrotation.
Bemerkung: Bei dieser Art der Codierung werden Leerzeichen, Groß- und Kleinbuchstaben meist unberücksichtigt gelassen. Mit dem Caesar-Code verschlüsselte Nachrichten können mit den **Caesar-Rad** problemlos entschlüsselt werden *(siehe Abbildung 1.2.)*.

Beispiel: Verschiebung um 4 Positionen/ Geheimzahl=4

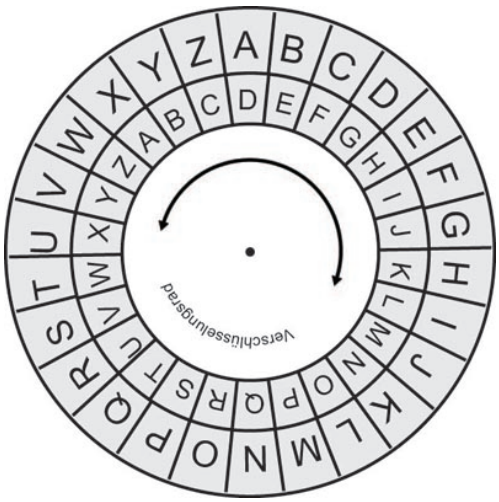

Abbildung 1.3: Caesar-Rad

Klartext	a	b	c	d	e	f	g	h	i	j	k	l	m
Geheimtext	e	f	g	h	i	j	k	l	m	n	o	p	q

Klartext	n	o	p	q	r	s	t	u	v	w	x	y	z
Geheimtext	r	s	t	u	v	w	x	y	z	a	b	c	d

Klartext	dasisteinbeispiel
Geheimtext	hewmwximrfimwtmip

Abbildung 1.4: Polygraphiae libri sex

855 n.Chr.	Erscheinung des ersten Buches über Kryptographie verfasst von: Abu ʻAbd al-Raham al-Khahil ibn Ahmad ibn'Amr ibn Tammam al Farahidi al-Zadi al Yahamadi
1412 n.Chr.	Erscheinung einer 14-bändigen arabische Enzyklopädie , in der kryptographische kryptographische Methoden beschrieben werden. In dieser Enzyklopädie wird zum ersten Mal die Methode der mehrfachen Substitutuin verwendet.
1518 n.Chr.	Erscheinung des von **Johannes Trithemius** *(1462 - 1516)* (eigentlich: Johannes Heldenberg aus Trittenheim) im Jahr 1508 verfassten Buches über Kryptographie, die **"Polygraphiae libri sex"** (Sechs Bücher zur Polygraphie).
1586 n.Chr.	Erscheinung des Buches ***Traicitié de Chiffre***, verfasst von dem französischen Diplomaten **Blaise de Vignière** *(1523-1596)*.Der nach ihm benannte Vignière-Code ist der bekannte polyalphabetische Algorithmus
1628 n.Chr.	Dem französischen Kryptoanalytiker **Antoine Rossignol***(1600-1682)* entschlüsselt eine fremde Botschaft und beendet dadurch die Belagerung der Stadt Réalmont durch die Hugenotten Seit diesem Zeitpunk werden Kryptoanalytiker bei dem Militär angestellt.
1863 n.Chr.	Der preußische Offizier **Friedrich Kasiski** *(1805-1881)* veröffentlicht sein Werk **"Die Geheimschriften und die Dechiffrierkunst"**. In diesem Werk wird auch ein Verfahren vorgestellt zur Lösung von polyalphabetischen Chiffren, die bis dahin als unlösbar galten.
1883 n.Chr.	**Jean Guillaume Auguste Victor François Hubert Kerckhoffs von Nieuwenhof** *(1835-1903)* beschreibt im Buch **"La cryptographie militaire"** die Prinzipien der strategischen Kryptologie
1917 n.Chr.	**Gilbert Sandfort Vernam** *(1890-1960)* verschlüsselt Nachrichten mit dem **One-Time-Pad**
1917 n.Chr.	Der deutsche Außenminister, Zimmermann, offriert den Mexikanern die verlorenen Gebiete Texas,New Mexico und Arizona, falls sie gegen die Amerikaner in den Krieg gehen. Da die Engländer diese Nachricht abfangen und entschlüsseln konnten und diese an den US- Präsidenten weitergeleitet haben, hat die USA Deutschland den Krieg erklärt.
1928 n.Chr.	Der deutschen Ingenieur Arthur Scherbius entwickelt die Rotormaschine **Enigma**.
1941 n.Chr.	Decodierung der japanischen Angriffsmeldung für den 2. Weltkrieg
1950 n.Chr.	Die US-Amerikaner **Martin Hellman** und **Whitfield Diffie** zeigen, dass **Public-Key-Verfahren** theoretisch möglich sind
1978 n.Chr.	Veröffentlichung des **RSA-Verfahrens**.

Für mehr Methoden und genauere Geschichte siehe [2] und [3].

Kapitel 2

Zahlentheoretische Grundlagen

Die RSA-Verschlüsselung ist eines der klassischen Anwendungsgebiete der elementaren Zahlentheorie. Aus diesem Grund werden wir versuchen in diesem Kapitel die wichtigsten Definitionen und Sätze der Zahlentheorie zu wiederholen.

2.1 Teilbarkeit

Vorbemerkungen:

1. $\mathbb{Z}$ bildet mit der Addition und Multiplikation einen kommutativen Ring, aber keinen Körper
2. $\mathbb{Z}$ ist nullteilerfrei $\iff$ $x, y \in \mathbb{Z}$ mit $xy = 0 \Rightarrow x = 0$ oder $y = 0$
3. $\mathbb{Z}$ ist total geordnet. Diese Ordnung ($\leq$) ist mit der Addition und Multiplikation verträglich

Definition 1. *(Teiler)*

Seien $m \neq 0$ und n ganze Zahlen. Man sagt: n ist durch m teilbar (anders: m ist ein Teiler von n), wenn es genau eine ganze Zahl d gibt mit $n = md$ ($\iff \exists d \in \mathbb{Z} : n = md$).
Notation: $m \mid n$

Bemerkung 1. *(Komplementärteiler)*
Seien $d, m, n \in \mathbb{Z}$ mit $m \neq 0$. d wird Komplementärteiler von n genannt, falls $n = md$ gilt.

Satz 1. *(Rechenregeln)*

1. $\forall n \neq 0 \in \mathbb{Z}$ *gilt:* $1 \mid n$, $n \mid n$ *und* $n \mid 0$
2. $m|n \Rightarrow (-m) \mid n$ *und* $m \mid (-n)$
3. *Aus* $m \mid n$ *und* $n \neq 0$ *folgt* $|m| \leq |n|$
4. *Für* $n \in \mathbb{Z}$ *mit* $n \mid 1$ *gilt:* $n \in \{1, -1\}$
5. *Aus* $m \mid n$ *und* $n \mid m$ *folgt entweder* $m = n$ *oder* $-m = n$
6. $l \mid m$ *und* $m \mid n \Rightarrow l \mid n$ *(*$l \in \mathbb{Z}$*)*
7. $m \mid n \;\Rightarrow\; (lm) \mid (ln) \; \forall \, l \in \mathbb{Z}$
8. $\forall l \in \mathbb{Z} \setminus \{0\}$ *gilt:* $(lm) \mid (ln) \Rightarrow m \mid n$
9. $m \mid n_i \; (1 \leq i \leq k) \;\Rightarrow\; m \mid \sum_{i=1}^{k} l_i n_i \;\; \forall l_i \in \mathbb{Z}$

10. $m_i \mid n_i \ (1 \leq i \leq k) \ \Rightarrow \ \prod_{i=1}^{k} m_i \mid \prod_{i=1}^{k} n_i \ \ \forall l_i \in \mathbb{Z}$

Wir werden exemplarisch die Beweise zu den Regeln (2),(5),(6) und (9) führen.

Beweis:. .

ad 2.)
Bei deiser Regel brauchen wir bloss die Definition von Teiler anwenden:
Nach Voraussetzung ist m ein Teiler von n. Das bedeutet, es existiert ein $d \in \mathbb{Z}$ mit $n = md$. Aus $n = md$ folgt sofort $n = (-m)(-d)$ und $(-n) = m(-d)$. Somit ist die Regel bewiesen.

ad 5.)
Für $n = m = 0$ ist die Aussage trivial.
Wegen $m \mid n$ gilt automatisch: $m \neq 0 \Rightarrow n \neq 0 \overset{Regel(3)}{\Rightarrow} |m| \leq |n|$
Wegen $n \mid m$ gilt automatisch: $n \neq 0 \Rightarrow m \neq 0 \overset{Regel(3)}{\Rightarrow} |n| \leq |m|$.
Kombinieren wir diese Aussagen, so erhalten wir $|m| = |n|$. Dies ist aber gleichbedeutend mit $n \in \{-m, m\}$, also unserer Behauptung.

ad 6.)

- $l \mid m \Rightarrow \exists d \in \mathbb{Z} : ld = m$
- $m \mid n \Rightarrow \exists k \in \mathbb{Z} : mk = n$

Die Kombination der "Aussagen" von oben ergibt die Behauptung: $n = mk \overset{m=ld}{\Rightarrow} n = ldk \Rightarrow l \mid n$

ad 9.)
$m \mid n_i \ (1 \leq i \leq k) \Rightarrow \ \forall i \in \{1, ..., k\} \ \exists d_i : n_i = md_i \Rightarrow \ \sum_{i=1}^{k} l_i n_i = \sum_{i=1}^{k} l_i (md_i) = m \sum_{i=1}^{k} l_i d_i$
$\Rightarrow | \sum_{i=1}^{k} l_i n_i$

□

Satz 2. *(Division mit Rest)*

Seien $\boldsymbol{m, n \in \mathbb{Z}, n > 0}$. *Dann gibt es eindeutig bestimmte* $\boldsymbol{k, r \in \mathbb{Z}}$ *mit* $\boldsymbol{0 \leq r < n}$ *sodass gilt:*

$$\boldsymbol{m = k \cdot n + r}$$

Beweis:
Ohne Beweis

□

Definition 2. *(Gemeinsamer Teiler, größter gemeinsamer Teiler)*
Seien $\boldsymbol{n_1, n_2, \ldots, n_k \in \mathbb{Z}}$. $\boldsymbol{m \in \mathbb{Z}}$ *wird gemeinsamer Teiler von* $\boldsymbol{n_i}$ *für* $\boldsymbol{i = 1, \ldots, k}$ *genannt, falls* $\boldsymbol{m \mid n_i}$ *für* $\boldsymbol{i = 1, \ldots, k}$.

Der größte gemeinsame Teiler der Zahlen $\boldsymbol{n_1, n_2, \ldots, n_k}$ *ist definiert als:*

$$\boldsymbol{ggT(n_1, n_2, \ldots, n_k) = max \{m \in \mathbb{Z} \ : \ m \mid n_i \text{ für } i = 1, \ldots, k\}}.$$

Satz 3. *(Euklidischer Algorithmus)*

Sind $a, b \in \mathbb{N}$ *und* $b \leq a$*. Führe wiederholt Division mit Rest durch:*

$$\begin{gathered} a = b \cdot k_0 + r_1 \text{ mit } 0 \leq r_1 < b \\ b = r_1 \cdot k_1 + r_2 \text{ mit } 0 \leq r_2 < r_1 \\ r_1 = r_2 \cdot k_2 + r_3 \text{ mit } 0 \leq r_3 < r_2 \\ r_2 = r_3 \cdot k_3 + r_4 \text{ mit } 0 \leq r_4 < r_3 \\ \vdots \\ r_{n-1} = r_n \cdot k_n + r_{n+1} \text{ mit } 0 \leq r_{n+1} < r_n \end{gathered}$$

Wegen $r_{n+1} < r_n < r_{n-1} < \ldots < r_1$ *gibt es ein kleinstes* n *mit* $r_{n+1} = 0$*. Es gilt:*

$$r_n = ggT(a, b)$$

Beweis:
Ohne Beweis

□

Beispiel: Berechne den größten gemeinsamen Teiler von 132 und 48!
Wir verwenden den euklidischen Algorithmus

$$\begin{gathered} 132 = 48 \cdot 2 + 36 \\ 48 = 36 \cdot 1 + \underline{\underline{12}} \\ 36 = 12 \cdot 3 + 0 \end{gathered}$$

Es gilt also : $ggT(132, 48) = 12$

Definition 3. *(Primzahl)*

Sei $p > 1 \in \mathbb{N}$*. Falls* p *eine Zahl ist, die nur die trivialen Teiler* 1 *und* p *als positive Teiler besitzt, heißt sie Primzahl.*

Definition 4. *(Kongruenz)*

Seien $a, b \in \mathbb{Z}$, $m \in \mathbb{N}$*. Man sagt,* a *und* b *seine kongruent modulo* m*, wenn* $m \mid (a - b)$
Notation: $a \equiv b \pmod{m}$*. Die Zahl* m *heißt Modul.*

Satz 4.
Seien $a, b \in \mathbb{Z}$, $m \in \mathbb{N}$*. Dann sind folgende Aussagen äquivalent:*

1. $a \equiv b \pmod{m}$
2. *Bei Division durch* m *haben* a *und* b *den selben Rest*

Beweis*:*

Sei $a = k_1 \cdot m + r_1$ und $b = k_2 \cdot m + r_2$ mit $0 \leq r_1, r_2 < m$

(1. $\Rightarrow$ 2.)

$a - b = (k_1 - k_2) \cdot m + r_1 - r_2$. $a \equiv b \pmod{m} \iff m \mid (a - b) \Rightarrow m \mid (k_1 - k_2) \cdot m + r_1 - r_2$
$m \mid (k_1 - k_2) \cdot m$ und $m \mid r_1 - r_2$. Es ist klar, dass $-m < r_1 - r_2 < m$ gilt. Daraus folgt $r_1 - r_2 = 0$,
was gleichbedeutend mit $r_1 = r_2$ ist.

$(2. \Rightarrow 1.)$

Da nach Voraussetzung $r_1 = r_2$ gilt, folgt $a - b = (k_1 - k_2) \cdot m + r_1 - r_2 = (k_1 - k_2) \cdot m \underbrace{\Rightarrow}_{\textit{per Definition}} a \equiv b \pmod{m}$ □

Satz 5.
Kongruent modulo $m \in \mathbb{N}$ zu sein ist eine Aequvalenzrelation.

Beweis:
Im ersten Schritt muss man die Reflexivität zeigen, das bedeutet wir müssen $a \equiv a \pmod{m}$ zeigen. Dies gilt aber trivialerweise da $a \equiv a \pmod{m} \iff m \mid (a-a) \iff m \mid 0$.

Um zu zeigen, dass es sich tatsächlich um eine Aequivalenzrelation handelt, müssen wir auch die Symmetrie zeigen. Zu zeigen ist also : $a \equiv b \pmod{m} \Rightarrow b \equiv a \pmod{m}$.
Dies gilt aber auch ,da :
$a \equiv b \pmod{m} \iff m \mid (a-b) \iff m \mid 0 \Rightarrow m \mid (-1)(a-b) \iff m \mid (b-a) \iff b \equiv a \pmod{m}$

Es fehlt uns nurmehr die Transitivität.Wir zeigen also :
$a \equiv b \pmod{m}, b \equiv c \pmod{m} \Rightarrow a \equiv c \pmod{m}$
Bei diesem abschließendem Teil des Beweises müssen wir einfach auf die Definition von Kongruenzen zurückgreifen.

$a \equiv b \pmod{m}, b \equiv c \pmod{m} \iff m \mid (a-b)\,,\ m \mid (b-c) \Rightarrow m \mid (a-b)+(b-c)$
$\Rightarrow m \mid (a-c) \iff a \equiv c \pmod{m}$

□

Satz 6. *Seien $a, b, c, d, k \in \mathbb{Z},\ k \neq 0$ und $m, n \in \mathbb{N}$. Dann gelten:*

1. $a \equiv b \pmod{m}$ *und* $c \equiv d \pmod{m} \Rightarrow a + c \equiv b + d \pmod{m}$
2. $a \equiv b \pmod{m}$ *und* $c \equiv d \pmod{m} \Rightarrow a \cdot c \equiv b \cdot d \pmod{m}$
3. $a \equiv b \pmod{m}$ *und* $k \mid m$, *so folgt* $a \equiv b \pmod{|k|}$
4. $a \equiv b \pmod{m} \Rightarrow k \cdot a \equiv k \cdot b \pmod{k \cdot m}$

Hier werden wir nur exemplarisch den Beweis zu (1.) führen, die anderen Beweise funktionieren analog

Beweis:

$a \equiv b \pmod{m} \iff m \mid (a-b)$ und $c \equiv d \pmod{m} \iff m \mid (c-d)$.
Aus diesen folgt: $m \mid (a-b)+(c-d) \iff m \mid (a+c)-(b+d) \iff a+c \equiv b+d \pmod{m}$

□

Definition 5. *(Diophantische Gleichung)*

Seien $a_1, a_2, \ldots, a_n, c \in \mathbb{Z}$. Eine Gleichung der Form $\sum_{i=1}^{n} a_i x_i = c$ heißt lineare diophantische Gleichung.

2.2 Der chinesische Restsatz

Die Lösung eines System der Form :

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ x &\equiv a_3 \pmod{m_3} \\ &\vdots \\ x &\equiv a_n \pmod{m_n} \end{aligned}$$

kann mit Hilfe des folgenden Satzes problemlos berechnet werden.

Satz 7. *(Der chinesische Restsatz)*

Seien $m_1, ..., m_n \in \mathbb{R}$ *paarweise relativ prim. Sei* $m := \prod_{i=1}^{n} m_i$ *und seien* $a_1, ..., a_n \in \mathbb{Z}$.
Dann $\exists! \, x \in \mathbb{Z}$ *mit* $0 \leq x < m$ *und* $x \equiv a_i \pmod{m_i}$ *für* $i = 1, ..., n$

Der Beweis zum obigen Satz besteht aus zwei Teilen. Im ersten Teil zeigen wir, dass x eindeutig ist, und im zweiten, dass die Lösung x überhaupt existiert.

Beweis:.

1. **Eindeutigkeit:**
   Seien $x, y \in \mathbb{Z}$, für die $x \equiv a_i \pmod{m_i}$ beziehungsweise $y \equiv a_i \pmod{m_i}$ für $i = 1, ..., n$ gilt. Dies ist gleichbedeutend mit folgender Aussage: $x \equiv y \pmod{m_i}$ für $i = 1, ..., n$. Wegen der Definition von Kongruenz muss $x - y$ also ein Vielfaches von m_i sein für $i = 1, ..., n$ und somit natürlich ein Vielfaches von $m := \prod_{i=1}^{n} m_i$.Da wir wissen, dass alle m_i paarweise teilerfremd sind, muss selbstverständlicherweise $x = y$ gelten. Somit ist die Eindeutigkeit der Lösung gezeigt.

2. **Existenz:**
   Sei $M_i := \frac{m}{m_i}$ und $N_i := M_i^{-1} \pmod{m_i}$ für $i = 1, ..., n$. Das Inverse von M_i ist wohldefiniert, weil $ggT(M_i, m_i) = 1$ für $i = 1, ..., n$ gilt. Dann ist

   $$x = \left(\sum_{i=1}^{n} a_i \cdot N_i \cdot M_i \right) \bmod m$$

   eine Lösung von $x \equiv a_i \pmod{m_i}$,da für $j = 1, ..., n$ gilt:

   $$\begin{aligned} x &\equiv \left(\sum_{i=1}^{n} a_i \cdot N_i \cdot M_i \right) \bmod m \pmod{m_j} \\ &\iff \\ x &\equiv \left(\sum_{i=1}^{n} a_i \cdot N_i \cdot M_i \right) \pmod{m_j} \\ &\iff \\ x &\equiv a_j \cdot \underbrace{N_j \cdot M_j}_{\equiv 1} + \left(\sum_{\substack{i=1, \\ i \neq j}}^{n} a_i \cdot N_i \cdot \underbrace{M_i}_{\equiv 0} \right) \pmod{m_j} \\ &\iff \\ x &\equiv a_j \pmod{m_j} \end{aligned}$$

Um Unklarheiten zu vermeiden,wollen wir nocheinmal erklären, wie wir vom vorletzten auf den letzten Schritt gekommen sind. Es muss klarerweise $N_j \cdot M_j \equiv 1 \pmod{m_j}$ gelten, da N_j als Inverses zu $M_j \pmod{m_j}$ definiert ist.
Für $i \neq j$ ist M_i ein Vielfaches von m_j daher gilt : $M_i \equiv 0 \pmod{m_j}$. Setzen wir diese "Teilergebnisse" zusammen, so erhalten wir das gewünschte Resultat.
Wegen der verwendeten modulo-Operation können wir uns auch sicher sein, dass x im gewünschten Bereich ist. Somit ist der zweite Teil des Beweises abgeschlossen.

□

2.2.1 Ein Beispiel

Ein burgenländischer Bauer möchte beim Golser Volksfest seine Pferde präsentieren. Werden die Pferde in 2-er -Reihen aufgestellt, so bleibt ein Pferd übrig. Stellt der Bauer seine Pferde in 3er- Reihen auf, so bleiben zwei Pferde übrig. Bei einer Aufstellung in 5er- Reihen bleiben drei Pferde übrig. Wie viele Pferde hat der Bauer mindestens?

Lösung:

1. Im ersten Schritt müssen wir den Text in die Sprache der Mathematik übersetzen. In diesem Beispiel bedeutet das, dass wir ein System von Kongruenzen aufstellen müssen. Bezeichne x die Anzahl der Pferde, die der Bauer hat. Wegen der Angabe erhalten wir:

$$\begin{aligned} x &\equiv 1 \pmod 2 \\ x &\equiv 2 \pmod 3 \\ x &\equiv 3 \pmod 5 \end{aligned}$$

2. Da $m_1 = 2,\ m_2 = 3,\ m_3 = 5$ paarweise relativ prim sind können wir den **chinesischen Restsatz** anwenden

3. Wie wir das bereits im Satz gesehen haben müssen wir ein gemeinsames Modulo m bilden. Für uns heißt das folgendes: $m := \prod_{i=1}^{3} m_i = 2 \cdot 3 \cdot 5 = 30$

4. In diesem Schritt bilden wir $M_i := \frac{m}{m_i}$ Wir gehen also bei der Rechnung analog vor, wie wie das beim Beweis getan haben.

   (a) $M_1 := \frac{m}{m_1} = \frac{30}{2} = 15$
   (b) $M_2 := \frac{m}{m_2} = \frac{30}{3} = 10$
   (c) $M_3 := \frac{m}{m_3} = \frac{30}{5} = 6$

5. Jetzt können wir neue Kongruenzen aufstellen und schauen, wann diese erfüllt sind:

$$\begin{aligned} 15 \cdot x_1 &\equiv 1 \pmod 2 && \text{erfüllt für } x_1 = 1 \\ 10 \cdot x_2 &\equiv 2 \pmod 3 && \text{erfüllt für } x_2 = 2 \\ 6 \cdot x_3 &\equiv 3 \pmod 5 && \text{erfüllt für } x_3 = 3 \end{aligned}$$

6. Eine Lösung x_0 kann wie folgt berechnet werden:

$$x_0 = \left(\sum_{i=1}^{3} x_i \cdot M_i \right) = 1 \cdot 15 + 2 \cdot 10 + 3 \cdot 6 = 53$$

7. Nach all diesen Vorbereitungen sind wir in der Lage eine Allgemeine Lösung anzugeben:

$$x = x_0 + k \cdot m = 53 + k \cdot 30 \quad \Longleftrightarrow \quad x = 23 + k \cdot 30 \text{ wobei } k \in \mathbb{Z}$$

Modulo Schreibweise:

$$x \equiv 23 \pmod{30}$$

8. Der Bauer hat also mindestens 23 Pferde.

2.3 Der Restklassenring $\mathbb{Z}_m$

Wir wissen bereits, dass kongruent modulo $m \in \mathbb{N}$ zu sein eine Aequivalenzrelation ist. Daher können wir die Aequivalenzklasse $\overline{a}$ von $a \in \mathbb{Z}$ konstruieren:

$\overline{a} = \{x \in \mathbb{Z} : x \equiv b \pmod{m}\} = \{x \in \mathbb{Z} : m \mid (x-a)\} = \{x \in \mathbb{Z} : \exists k \in \mathbb{Z} | x - a = k \cdot m\}$
$\{x \in \mathbb{Z} : \exists k \in \mathbb{Z} | x = a + k \cdot m\} = \{a + k \cdot m : k \in \mathbb{Z}\} = a + m \cdot \mathbb{Z}$

Definition 6. *(Restklasse modulo m)*

Für $m \in \mathbb{N}$ wird jede Aequivalenzklasse $\overline{a}$ als Restklasse modulo m bezeichnet.
Jedes x aus $\overline{a}$ wird als Repräsentant von $\overline{a}$ genannt.
Für die Menge der Restklassen modulo m schreibt man: $\mathbb{Z}_m$ oder $\mathbb{Z}/m\mathbb{Z}$
$\mathbb{Z}_m = \{\overline{a} : a \in \mathbb{Z}\}$ *bzw.* $\mathbb{Z}/m\mathbb{Z} = \{a + m\mathbb{Z} : a \in \mathbb{Z}\}$

2.4 Der Satz von Euler, der kleine Satz von Fermat

Definition 7. *(Die Eulersche φ-Funktion)*

Für $n \in \mathbb{N}$ ist die Eulersche φ-Funktion wie folgt definiert:

$$\varphi(n) := \mid \{m \in \{1, ..., n\} : ggT(m, n) = 1\} \mid$$

$\varphi(n)$ ist also die Anzahl der natürlichen, n nicht übersteigenden Zahlen, die zu m teilerfremd sind.

Satz 8. *Für eine Primzahl p gilt:*

1. $\varphi(p) = p - 1$
2. $\varphi(p^k) = p^k - p^{k-1}$ *für alle* $k \in \mathbb{N}$

Beweis:.

ad 1.) $\varphi(p) = p - 1$ ist trivial wegen der Definition für Primzahlen

ad 2.) Bei dem zweiten Teil des obigen Satzes haben wir mit den Zahlen $1, \ldots, p^k$ zu tun. Von diesen sind die Vielfachen von p nicht teilerfremd zu p^k. Das sind genau p^{k-1} Zahlen:

$$p \cdot 1, p \cdot 2, \ldots, p \cdot p^{k-1}$$

Daraus folgt die Behauptung. □

Lemma 1.
Seien $m, n \in \mathbb{N}$ *mit* $ggT(m,n) = 1$. *Dann gilt:*

$$\varphi(m \cdot n) = \varphi(m) \cdot \varphi(m)$$

ohne Beweis

Lemma 2.
Sei $m \in \mathbb{N}$ *und* $p_1^{\theta_1} \cdot p_2^{\theta_2} \cdot \ldots \cdot p_n^{\theta_n}$ *die Primfaktorzerlegung von* m. *Dann gilt:*

$$\varphi(m) = m \cdot \prod_{p|m} \left(1 - \tfrac{1}{p}\right)$$

Beweis:.

$$m = p_1^{\theta_1} \cdot p_2^{\theta_2} \cdot \ldots \cdot p_n^{\theta_n} \Rightarrow \varphi(m) = \varphi(p_1^{\theta_1} \cdot p_2^{\theta_2} \cdot \ldots \cdot p_n^{\theta_n}) \overbrace{=}^{\textit{Lemma 1.}} \varphi(p_1^{\theta_1}) \cdot \ldots \cdot \varphi(p_n^{\theta_n})$$

$$= (p_1^{\theta_1} - p_1^{\theta_1 - 1}) \cdot (p_2^{\theta_2} - p_2^{\theta_2 - 1}) \cdot \ldots \cdot (p_n^{\theta_n} - p_n^{\theta_n - 1}) = \prod_{i=1}^{n} (p_i^{\theta_i} - p_i^{\theta_i - 1})$$

$$= \underbrace{\left(\prod_{i=1}^{n} p_i^{\theta_i}\right)}_{m} \cdot \left(\prod_{i=1}^{n} (1 - \tfrac{1}{p_i})\right) = m \cdot \prod_{p|m} (1 - \tfrac{1}{p})$$

□

Satz 9. *(Der Satz von Euler-Fermat)*
Für $m \in \mathbb{N}$ *und* $a \in \mathbb{Z}$ *mit* $ggT(a,m) = 1$ *gilt:*

$$a^{\varphi(m)} \equiv 1 \pmod{m}$$

Beweis:.

Es gelte $ggT(a,m) = 1$. Die zu m teilerfremde Zahlen aus $\mathbb{Z}_m$ bezeichnen wir mit $k_1, k_2, \ldots, k_{\varphi(m)}$. Jetzt multiplizieren wir all diese Zahlen mit a und erhalten dadurch $a \cdot k_1, a \cdot k_2, \ldots, a \cdot k_{\varphi(m)}$. Da alle k_i und a teilerfremd zu m sind, ist auch deren Produkt teilerfremd zu m. Es gilt also: $ggT(a \cdot k_i, m) = 1$ für $i = 1, 2, \ldots, \varphi(m)$. Wenn wir a an alle k_i daranmultiplizieren, kommen wieder die selben k_i raus, nur in einer unterschiedlichen Reihenfolge .Daraus folgt:

$$a \cdot k_1 \cdot a \cdot k_2 \cdot \ldots a \cdot k_{\varphi(m)} = k_1 \cdot k_2 \cdot \ldots \cdot k_{\varphi(m)}$$

Das bedeutet:

$$a \cdot k_1 \cdot a \cdot k_2 \cdot \ldots a \cdot k_{\varphi(m)} \equiv k_1 \cdot k_2 \cdot \ldots \cdot k_{\varphi(m)} \pmod{m}$$

Jetzt können wir beide Seiten durch $k_1 \cdot k_2 \cdot \ldots \cdot k_{\varphi(m)}$ dividieren, weil alle k_i für $i = 1, \ldots, \varphi(m)$ teilerfremd zu m sind.Somit erhalten wir:

$$\underbrace{a \cdot a \cdot \ldots \cdot a}_{\varphi(m) - Mal} \equiv 1 \pmod{m}$$

$$\Longleftrightarrow$$

$$a^{\varphi(m)} \equiv 1 \pmod{m}$$

□

Der kleine Satz von Fermat kann als Spezialfall des Satzes von Euler verstanden werden

Satz 10. *(Der kleine Satz von Fermat)*
Für p Primzahl und $a \in \mathbb{Z}$ mit $ggT(a,p) = 1$ gilt:

$$a^{p-1} \equiv 1 \pmod{p}$$

Beweis:.

Für eine Primzahl p gilt: $\varphi(p) = p - 1$.Es gilt also:

$$a^{\varphi(p)} \equiv 1 \pmod{p} \iff a^{p-1} \equiv 1 \pmod{p}$$

Der Satz ist also tatsächlich ein Spezialfall des Satzes von Euler , zu dem wir den Beweis bereits geführt haben.

□

Satz 11. *(Der kleine Satz von Fermat; Spezialfall)*
Sei $n = p \cdot q$ ein Produkt zwei verschiedener Primzahlen und $a < n \in \mathbb{Z}$ und alle $k \in \mathbb{Z}$,so gilt:

$$a^{k \cdot \varphi(n)+1} \equiv a \pmod{n}$$

Ohne Beweis

Zum Abschluss dieses Kapitels wollen wir noch zwei , ziemlich abgeschwächte Definitionen geben, die fürs Arbeiten mit dem RSA-Algorithmus unerlässlich sind.

Definition 8. *(Einwegfunktion)*
Eine Funktion $f : X \to Y$ heißt Einwegfunktion (one-way function), wenn gilt:

1. *Es gibt ein effizientes Verfahren zur Berechnung von $y = f(x)$ für jedes $x \in X$. Anders ausgedruckt : $y = f(x)$ kann in Polinomialzeit berechnet werden.*
2. *Es gibt kein Effizientes Verfahren , um bei bekanntem y das $x := f^{-1}(y)$ zu berechnen für jedes $y \in Y$ bis auf vernachlässigbar viele.*

Wir können also problemlos $y = f(x)$ für jedes $x \in X$ berechnen, während es für alle $y \in Y$ *bis auf vernachlässigbar viele* schwer bis unmöglich ist $x := f^{-1}(y)$ zu berechnen. Ein Rechner kann also $y = f(x)$ in wenigen Sekunden bestimmen für jedes $x \in X$. Dagegen bräuchten selbst die besten Computer der Welt, Monate oder sogar Jahre um $y := f^{-1}(x)$ zu berechnen.
An dieser Stele wollen wir unbedingt betonen, dass wir uns nie sicher sein können, ob wir tatsächlich mit einer Einwegfunktion arbeiten oder nicht, da es nicht einmal bewiesen ist, dass Einwegfunktionen existieren. Wollen wir also richtig formulieren, so reden wir von **Kandidaten für Einwegfunktione**n und nicht von Einwegfunktionen.

Definition 9. *(Geheimtürfunktion[a])*
Eine Funktion $f : X \to Y$ heißt Geheimtürfunktion (trapdoor function), wenn gilt:

1. *Es gibt einen effizienten Algorithmus E zur Berechnung von $y = f(x)$ für jedes $x \in X$.*
2. *$x := f^{-1}(y)$ kann für jedes $y \in Y$ leicht mit einem effizienten Algorithmus D berechnet werden. Man kann also E nur dann umkehren, wenn Zusatzinformationen vorhanden sind.*

[a] auch als Falltürfunktion genannt

Die Sicherheit von dem RSA-Verfahren beruht auf Geheimtürfunktionen.

Kapitel 3

Theorie der RSA-Verschlüsselung

Im Jahr 1977 gelang es den damals jungen Mathematikern Ron Rivest, Adi Shamir und Leonard Adleman *(siehe Abbildung 3.1)* ein Verfahren zu entwickeln, mit dem man Daten sehr sicher verschlüsselt werden können, und das gleichzeitig für den alltäglichen Gebrauch geeignet ist. Die von ihnen entwickelte Methode wird heute als **RSA**-Verfahren[1] genannt.

3.1 Der RSA-Algorithmus

3.1.1 Herleitung

Zwei Personen **A** und **B** wollen untereineander geheime Nachrichten austauschen. Sei **A** der Sender der Nachricht und **B** der Empfänger.

1. Als erster Schritt muss sich **B** zwei große Primzahlen p und q ausdenken. Am besten sollen die beiden Primzahlen gleichviele Ziffer haben. Dann bildet der Empfänger

$$n = p \cdot q$$

2. Im zweiten Schritt muss der Empfänger (**B**) $\varphi(n)$ bestimmen, wobei φ die Eulersche φ-Funktion bezeichnet.

   Wegen *Lemma 1.* und *Satz* gilt :

$$\varphi(n) = \varphi(p \cdot q) = \varphi(p) \cdot \varphi(q) = (p-1)(q-1)$$

3. Jetzt sucht **B** eine zu n teilerfremde Zahl $e \in \mathbb{N}$ mit $1 < e < \varphi(n)$

4. Nach all diesen Schritten kann **B** das für die Verschlüsselung notwendige Zahlenpaar $\underbrace{(n; e)}_{\text{Öffentlicher Schlüssel}}$

   bekannt geben

5. Will der Absender eine Nachricht an **B** senden, so muss er den Text der geheimen Nachricht in eine Zifferfolge umwandeln, die aus gleichlangen Blöcken x besteht. Es muss aber $1 \leq x < n-1$ gelten.

6. Als Nächstes muss **A** den Rest m berechnen, der bei der Division von x^e **durch** n entsteht. In Formeln:

$$m \equiv x^e \pmod{n}$$

7. **A** muss die Zahl m an **B** senden [2].

[1] benannt nach den Erfindern

[2] für jeden Ziffernblock, x aus dem Klartext

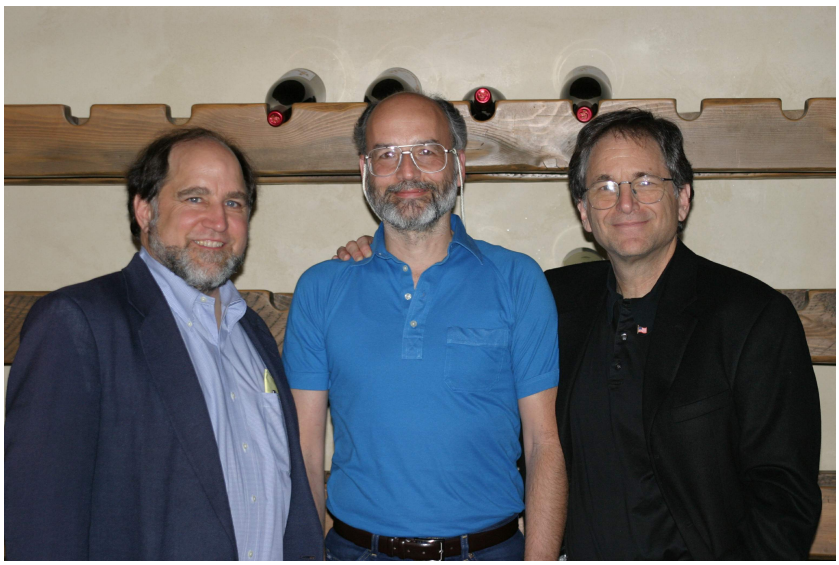

Abbildung 3.1: Ron Rivest, Adi Shamir und Leonard Adleman

8. Um die Nachricht m entschlüsseln zu können muss **B** die Lösung folgender linearen Kongruenz kennen:

$$d \cdot e \equiv 1 \pmod{\varphi(n)}$$

In obiger Gleichung ist $\underbrace{d}_{Privater\ Schlüssel}$ die Inverse zu $e \pmod{\varphi(n)}$ und gilt:

$$1 < d < \varphi(n)$$

Bemerkung:

d muss existieren, da e teilerfremd zu $\varphi(n)$ gewählt ist.

9. Für x gilt:

$$x \equiv m^d \pmod{n}$$

10. Nach all diesen Vorbereitungen kann **B** die Ziffernfolge in Text umwandeln

3.2 Primzahlerzeugung

Wie wir bereits beschrieben haben, sind Primzahlen zur Schlüsselerzeugung notwendig. Sie müssen also erzeigt werden. Wie das in der Praxis getan wird, werden wir in diesem kurzen Unterkapitel beschreiben.
Die Grundidee ist, dass es viel einfacher ist festzustellen , ob eine vorgegebene Zahl eine Primzahl ist, als ihre Primfaktoren anzugeben. Wollen wir Primzahlen erzeugen, so stehen uns zahlreiche Methoden zur Verfügung. An dieser Stelle wollen wir den wahrscheinlich häufigst gebräuchlichsten Algorithmus den **Rabin-Miller -Algorithmus** [3] vorstellen.

1. Wir wählen zufällig eine Zahl p aus, die wir testen werden
2. Wir berechnen d. Diese Zahl gibt an, wie oft $p - 1$ durch 2 geteilt werden kann. Das heißt:

[3] benannt nach den Entwicklern **Gary L. Miller** und **Michael O. Rabin**

$$d := \frac{p-1}{2^s}$$

3. Jetzt können wir den Test durchführen. Dieser besteht aus mehreren Schritten:

   (a) Wir wählen eine Zufallszahl a, mit $a < p$

   (b) In der ersten Teststufe überprüfen wir, ob $a^d \equiv 1 \pmod{p}$. Ist $a^d \equiv 1 \pmod{p}$,also der Test bestanden, so kann die nächste Zufallszahl berechnet werden, da wir ausschliessen können, dass a eine Pseudo-Primzahl ist[4].

   (c) Ist $a^d \not\equiv 1 \pmod{p}$, dann gehen wir in die zweite Teststufe.

   (d) Zweite Teststufe: Wir berechnen für alle $r \in \{0, \ldots, s-1\}$ ob $2^{r \cdot d} \equiv -1 \pmod{p}$. Fällt die Zahl durch alle Werte von r, so ist es klar, dass p keine Primzahl ist.

   (e) Wird in der zweiten Stufe auch nur ein Test bestanden, so muss die nächste Zufallszahl berechnet werden.

Wir müssen noch etwas erwähnen. Für jeden bestandenen Test ist die Wahrscheinlichkeit, dass dies ein Zufall war $\frac{1}{4}$. Daraus folgt, dass wir bereits nach 5 bestandenen Tests mit $99,902\%$ Wahrscheinlichkeit behauptet werden kann, dass p eine Primzahl ist.

3.2.1 Beispiel

Wir wollen Testen, ob **33487** eine Primzahl ist. Der Einfachkeit halber werden wir den Test nur 5-Mal wiederholen. Wir müssen uns aber klarmachen, dass mit steigender Anzahl an Wiederholungen sich die Warscheinlichkeit auch erhöht, dass die von uns gewählte Zahl tatsächlich eine Primzahl ist.

[4]Eine Pseudoprimzahl ist eine zusammengesetzte, natürliche Zahl, die gewisse Eigenschaften mit Primzahlen gemeinsam hat, selbst aber keine Primzahl ist.

Zu testende Zahl :$\mathbf{33487}$
Anzahl der Tests :$\mathbf{5}$

Ausgangsberechnung:
$\mathbf{33487}$
$s = 1$
$d = 16743$

Test 1:
Zufällig gewählt :$a = 23964$
$23964^{16743} \equiv 1 \pmod{33487} \rightarrow$ Bestanden

Test 2:
Zufällig gewählt: $a = 28962$
$28962^{16743} \equiv 1 \pmod{33487} \rightarrow$ Bestanden

Test 3:
Zufällig gewählt: $a = 11233$
$11233^{16743} \equiv 33486 \pmod{33487} \not\equiv 1 \pmod{33487} \rightarrow$ Durchgefallen
$11233^{16743} \equiv -1 \pmod{33487} \rightarrow$ Bestanden

Test 4:
Zufällig gewählt: $a = 10642$
$10642^{16743} \equiv 33486 \pmod{33487} \not\equiv 1 \pmod{33487} \rightarrow$ Durchgefallen
$10642^{16743} \equiv -1 \pmod{33487} \rightarrow$ Bestanden

Test 5:
Zufällig gewählt: $a = 1244$
$1244^{16743} \equiv 33486 \pmod{33487} \not\equiv 1 \pmod{33487} \rightarrow$ Durchgefallen
$1244^{16743} \equiv -1 \pmod{33487} \rightarrow$ Bestanden

$\mathbf{33487}$ ist also mit $1 - \left(\frac{1}{4}\right)^5 = 99,90234375\%$ Wahrscheinlichkeit eine Primzahl.

3.3 Korrektheit des RSA-Algorithmus

Satz 12. *(Korrektheit des RSA-Algorithmus)*
Es gelte:

1. $n = p \cdot q$ *für* p, q *Primzahlen*
2. $d \cdot e \equiv 1 \pmod{\varphi(n)}$
3. $m \equiv x^e \pmod{n}$

Dann ist der RSA-Algorithmus korrekt , das heißt:

$$m^d \equiv x \pmod{n}$$

Beweis:.

$d \cdot e \equiv 1 \pmod{\varphi(n)}$ ist gleichbedeutend damit, dass ein $k \in \mathbb{Z}$ existiert, mit $d \cdot e = 1 + k \cdot \varphi(n)$. Da nach Voraussetzung $m \equiv x^e \pmod{n}$ gilt, muss aufgrund der Rechenregeln $m^d \equiv (x^e)^d \pmod{n}$ gelten.

$$m^d \equiv (x^e)^d \pmod{n} \equiv x^{ed} \pmod{n} \equiv x^{1+k\cdot\varphi(n)} \pmod{n}$$

Es bleibt nurmehr zu zeigen, dass

$$x^{1+k\cdot\varphi(n)} \equiv x \pmod{n}$$

gilt. Bei diesem abschließenden Teil des Beweises müssen wir 4 Fälle betrachten:

1.Fall: $ggT(x,n) = 1$

$$x^{1+k\cdot\varphi(n)} \equiv x \cdot x^{k\cdot\varphi(n)} \pmod{n} \equiv x \cdot (x^{\varphi(n)})^k \pmod{n}$$

$$\underbrace{\Rightarrow}_{\text{Satz von Euler–Fermat}} \quad x^{1+k\cdot\varphi(n)} \equiv x \cdot 1 \pmod{n} \Rightarrow \text{Behauptung}$$

2.Fall: $ggT(x,n) = n = p \cdot q$

$$ggT(x,n) = n = p \cdot q \Rightarrow n \mid x$$

Nach Voraussetzung gilt: $0 \leq x \leq n-1 \Rightarrow x = 0$

$$\Longleftrightarrow$$

$$x \equiv 0 \pmod{n}$$

$$x^{1+k\cdot\varphi(n)} \equiv 0^{1+k\cdot\varphi(n)} \pmod{n} \equiv 0 \pmod{n} \equiv x \pmod{n}$$

3.Fall: $ggT(x,n) = p$ und $ggT(x,q) = 1$

$$ggT(x,q) = 1 \underbrace{\Rightarrow}_{\text{Satz von Euler}} x^{\varphi(q)} \equiv 1 \pmod{q}$$

$$\underbrace{\Rightarrow}_{q \text{ ist Primzahl}} x^{q-1} \equiv 1 \pmod{q} \Rightarrow x^{(q-1)(p-1)\cdot k} \equiv 1^{(p-1)\cdot k} \pmod{q}$$

$$\underbrace{\Rightarrow}_{\varphi(n)=(p-1)(q-1)} x^{\varphi(n)\cdot k} \equiv 1^{(p-1)\cdot k} \pmod{q} \Rightarrow x^{\varphi(n)\cdot k} \equiv 1 \pmod{q}$$

$$\underbrace{\Rightarrow}_{\cdot x} x^{\varphi(n)\cdot k+1} \equiv x \pmod{q}$$

Wegen $ggT(x,n) = p$ gilt: $x \equiv 0 \pmod{p} \Rightarrow x^{\varphi(n)\cdot k+1} \equiv 0 \pmod{p}$

$$\underbrace{\Rightarrow}_{x \equiv 0 \pmod{p}} x^{\varphi(n)\cdot k+1} \equiv x \pmod{p}$$

Mit $x^{\varphi(n)\cdot k+1} \equiv x \pmod{q}$ und $x^{\varphi(n)\cdot k+1} \equiv x \pmod{p}$ erhalten wir

$$x^{\varphi(n)\cdot k+1} \equiv x \pmod{n}$$

4.Fall: $ggT(x,n) = q$ und $ggT(x,p) = 1$

$$ggT(x,p) = 1 \underbrace{\Rightarrow}_{\text{Satz von Euler}} x^{\varphi(p)} \equiv 1 \pmod{p}$$

$$\underbrace{\Rightarrow}_{p \text{ ist Primzahl}} x^{p-1} \equiv 1 \pmod{p} \Rightarrow x^{(p-1)(q-1)\cdot k} \equiv 1^{(q-1)\cdot k} \pmod{p}$$

$$\underbrace{\Rightarrow}_{\varphi(n)=(p-1)(q-1)} x^{\varphi(n)\cdot k} \equiv 1^{(p-1)\cdot k} \pmod{p} \Rightarrow x^{\varphi(n)\cdot k} \equiv 1 \pmod{p}$$

$$\underbrace{\Rightarrow}_{\cdot x} x^{\varphi(n)\cdot k+1} \equiv x \pmod{p}$$

Wegen $ggT(x,n) = q$ gilt: $x \equiv 0 \pmod{q} \Rightarrow x^{\varphi(n)\cdot k+1} \equiv 0 \pmod{q}$

$$\underbrace{\Rightarrow}_{x \equiv 0 \pmod{q}} x^{\varphi(n)\cdot k+1} \equiv x \pmod{q}$$

Mit $x^{\varphi(n)\cdot k+1} \equiv x \pmod{p}$ und $x^{\varphi(n)\cdot k+1} \equiv x \pmod{q}$ erhalten wir

$$x^{\varphi(n)\cdot k+1} \equiv x \pmod{n}$$

Somit ist die Korrektheit gezeigt.

□

3.4 Beispiel

Attila und Markus wollen untereinander Informationene austauschen. Sei Markus der Empfänger, der von dem Absender eine Nachricht erwartet. Markus sucht zwei beliebige Primzahlen p und q aus und berechnet $n = p \cdot q$.
Markus wählt $p = 43$ und $q = 67$. Daraus folgt : $n = 43 \cdot 67 = 2881$.
Jetzt muss Markus $\varphi(n)$ berechnen:

$$\varphi(n) = \varphi(p \cdot q) \overbrace{=}^{p,q \text{ Primzahlen}} (p-1) \cdot (q-1) = 42 \cdot 66 = 2772$$

Markus muss noch eine, zu $\varphi(n)$ teilerfremde Zahl e finden:

Markus wählt $e = 115$

Um sicher gehen zu können, dass e tatsächlich teilerfremd zu $\varphi(n)$ ist, berechnet Markus den größten gemeinsamen Teiler der beiden Zahlen $\varphi(n)$ und $\varphi(e)$ mit Hilfe des Euklidischen Algorithmus:

$$ggT(\varphi(n), e) = ggT(2772, 115) = ?$$

$$\begin{aligned} 2772 &= 115 \cdot 24 + 12 \\ 115 &= 24 \cdot 4 + 19 \\ 24 &= 19 \cdot 1 + 5 \\ 19 &= 5 \cdot 3 + 4 \\ 5 &= 4 \cdot 1 + \underline{1} \\ 4 &= 4 \cdot 1 + 0 \end{aligned}$$

Es gilt :$ggT(\varphi(n), e) = ggT(2772, 115) = 1 \iff e$ und $\varphi(n)$ sind tatsächlich teilerfremd

Markus übermittelt jetzt den **öffentlichen Schlüssel** (n, e) an Attila :

$$n = 2881 \ , \ e = 115$$

Attila ist jetzt im Besitze des öffentlichen Schlüssel und will Markus die geheime Nachricht $x = 6$ zukommen lassen, die er durch $m \equiv x^e \pmod{n}$ verschlüsselt:

$$\begin{aligned} m &\equiv x^e \pmod{n} \\ m &\equiv 6^{115} \pmod{2881} \\ m &\equiv (6^5)^{23} \pmod{2881} \\ m \equiv (7776)^{23} \pmod{2881} &\equiv (\underbrace{7776}_{2 \cdot 2881 + 2014})^{23} \pmod{2881} \equiv (2014)^{23} \pmod{2881} \\ \equiv (2014^2)^{11} \cdot 2014 \pmod{2881} &\equiv (\underbrace{4056196}_{1407 \cdot 2881 + 2629})^{11} \cdot 2014 \pmod{2881} \equiv (2629)^{11} \cdot 2014 \pmod{2881} \\ \equiv (2629^2)^5 \cdot \underbrace{2629 \cdot 2014}_{1837 \cdot 2881 + 2409} \pmod{2881} &\equiv (\underbrace{6911641}_{2399 \cdot 2881 + 122})^5 \cdot 2409 \pmod{2881}) \equiv (122)^5 \cdot 2409 \pmod{2881} \\ \equiv (122^2)^2 \cdot \underbrace{122 \cdot 2409}_{=293898 = 102 \cdot 2881 + 36} \pmod{2881} &\equiv (\underbrace{14884}_{5 \cdot 2881 + 479})^2 \cdot 36 \pmod{2881} \equiv (479)^2 \cdot 36 \pmod{2881} \\ \equiv (\underbrace{229441}_{79 \cdot 2881 + 1842}) \cdot 36 \pmod{2881} &\equiv \underbrace{1842 \cdot 36}_{=66312 = 23 \cdot 2881 + 43} \pmod{2881} \underline{\underline{\equiv 49 \pmod{2881}}} \end{aligned}$$

Attila sendet jetzt die Nachricht $m = 49$ an Markus.
Markus muss jetzt die Nachricht entschlüsseln, dazu muss er aber zuerst die Kongruenz $d \cdot e \equiv 1 \pmod{\varphi(n)}$ lösen.

$$\begin{aligned} d \cdot e \equiv 1 \pmod{\varphi(n)} &\Rightarrow d \cdot 115 \equiv 1 \pmod{2772} \\ &\iff \\ \exists\, k \in \mathbb{Z} \text{ mit } 115 \cdot d - 1 = 2772 \cdot k &\Rightarrow 115 \cdot d - 2772 \cdot k = 1 \end{aligned}$$

Diese Gleichung ist eine lineare Diopantische Gleichung in zwei Unbekannten, die genau dann gelöst werden kann, wenn $ggT(115, 2772) = 1$ gilt. Wir müssen also erneut mit dem Euklidischen Algorithmus nachrechnen, ob die beiden Zahlen tatsächlich teilerfremd sind:

$$ggT(115, 2772) =?$$
$$2772 = 24 \cdot 115 + 12$$
$$115 = 9 \cdot 12 + 7$$
$$12 = 1 \cdot 7 + 5$$
$$7 = 1 \cdot 5 + 2$$
$$5 = 2 \cdot 2 + \underline{1}$$
$$2 = 2 \cdot 1 + 0$$

$ggT(115, 2772) = 1$,d.h. 115 und 2772 sind wirklich teilerfremd.

Jetzt muss Markus rückwärts rechnen:

$$1 = 5 - 2 \cdot 2 = 5 - (7 - 5) \cdot 2 = 5 - 2 \cdot 7 + 2 \cdot 5 = -2 \cdot 7 + 3 \cdot 5 = -2 \cdot 7 + 3 \cdot (12 - 7) = 3 \cdot 12 - 5 \cdot 7$$
$$= 3 \cdot 12 - 5 \cdot (115 - 9 \cdot 12) = 48 \cdot 12 - 5 \cdot 115 = 48 \cdot (2772 - 24 \cdot 115) - 5 \cdot 115 = (48) \cdot 2772 - (1157) \cdot 115$$

Somit erhält Markus als Lösung :

$$d_0 = -1157 \text{ und } k_0 = 48$$

Nach dieser Rechnerei ist Markus immernoch nicht fertig. Er muss noch die allgemeine Form der Lösung bestimmen:

$$d_t = 1157 + 2772 \cdot t$$
$$k_t = 48 + 115 \cdot t$$

Es muss aber auch $1 < d < \varphi(n) = (p - 1) \cdot (q - 1)$ gelten. Deswegen wählt Markus $t = 1$:

$$d_t = -1157 + 2772 \cdot t$$
$$k_t = 48 + 115 \cdot t$$

Somit erhält er:

$$d_1 = -1157 + 2772 \cdot 1 = 1615$$
$$k_1 = 48 + 115 \cdot 1 = 163$$

Nach all diesen Vorbereitungen ist Markus endlich in der Lage die Nachricht zu entschlüsseln. Dabei verwendet er die Formel:

$$m^d \equiv x \pmod{n} \Rightarrow 49^{1165} \equiv x \pmod{2881} \Rightarrow x = 49^{1165} \pmod{2881} \Rightarrow \underline{\underline{x = 6}}$$

3.5 Modulares Potenzieren

Wie wir oben gesehen haben ,ist die Berechnung von $\mathbf{6^{115} \equiv \pmod{2881}}$ nicht all zu einfach . In der Zahlentheorie ist modulares Potenzieren, also das Erheben einer Zahl durch in eine Potenz modulo einer anderen Zahl eine häufig vorkommende Operation. Bei dieser suchen wir einen effizienten Weg zur Berechnung von $\boldsymbol{a^b \pmod{n}}$.
Sei $\boldsymbol{(b_k,\ b-k-1,\ \ldots, b_1,\ b_0)}$ die Binärdarstellung von $\boldsymbol{b}$. Es ist ja offensichtlich, dass die Binärdarstellung dieser Zahl $\boldsymbol{k+1}$ Bit lang ist , und $\boldsymbol{b_k}$ der höchstwertige- und $\boldsymbol{b_0}$ der niedrigstwertige Bit ist. Durch folgenden Algorithmus kann $\boldsymbol{a^c \pmod{n}}$ berechnet werden, wobei $\boldsymbol{c}$ durch Verdopplungen und Inkrementieren von 0 auf b erhöht wird:

1. $\boldsymbol{c = 0}$
2. $\boldsymbol{d = 1}$
3. $\boldsymbol{(b_k,\ b-k-1,\ \ldots, b_1,\ b_0)}$ die Binärdarstellung von $\boldsymbol{b}$
4. For i=k downto 0
5. $\quad \boldsymbol{c = 2c}$
6. $\quad \boldsymbol{d = (d \cdot d) \pmod{n}}$
7. $\quad$ **iff** $\boldsymbol{b_i == 1}$
8. $\quad\quad \boldsymbol{c = c + 1}$
9. $\quad\quad \boldsymbol{d = (d \cdot a) \pmod{n}}$
10. return d

Für genauere Details siehe [7]

3.5.1 Beispiel

Berechne $\mathbf{6^{115} \equiv \pmod{2881}}$ mit der Methode des modularen Potenzierens !

Vorgehensweise

1. Stelle den Exponenten in Binärform dar!
2. Erster Faktor ist gleich der Basis
3. Die weiteren Faktoren ergeben sich jeweils durch Quadrieren des vorhergehenden Faktors
4. Bit im Exponenten ist 1 $\Rightarrow$ das Ergebnis soll mit dem jeweiligen Faktor multipliziert werden
5. Bit im Exponenten ist 0 $\Rightarrow$ das Ergebnis wird nicht verändert

Nach jedem Quadrieren und jeder Multiplikation wird eine Modulo-Operation ausgeführt. Diese Operation hat keinen Einfluß auf das Endergebnis, sie dient nur dazu, die Größe der Zwischenergebnisse zu reduzieren.

Berechnung:

(mit Programm)

Darstellung des Exponenten in Binärform

$$115 = 1110011$$

Initialisierung

$ergebnis_0$=1
faktor=6
m=2881

Schleife

Iteration 1: Bit im Exponenten gesetzt
$faktor_1$ = 6
$ergebnis_1$ = $ergebnis_0$ * $faktor_1$ % m = 1 * 6% 2881 = 6

Iteration 2: Bit im Exponenten gesetzt

$faktor_2$ = $faktor_1^2$% m = 62 % 2881 = 36
$ergebnis_2$ = $ergebnisr_1$ * $faktorr_2$ % m = 6 * 36 % 2881 = 216

Iteration 3: Bit im Exponenten nicht gesetzt
$faktor_3$ = $faktor_2^2$ % m = 362 % 2881 = 1296
$ergebnisr_3$ = $ergebnis_2$ = 216

Iteration 4: Bit im Exponenten nicht gesetzt
$faktor_4$ = $faktor_3^2$ % m = 12962 % 2881 = 2874
$ergebnis_4$ = $ergebnis_3$ = 216

Iteration 5: Bit im Exponenten gesetzt
$faktor_5$ = $faktor_4^2$ % m = 28742 % 2881 = 49
$ergebnis_5$ = $ergebnis_4$ * $faktor_5$ % m = 216 * 49 % 2881 = 1941

Iteration 6: Bit im Exponenten gesetzt
$faktor_6$ = $faktor_5^2$ % m = 492 % 2881 = 2401
$ergebnis_6$ = $ergebnis_5$ * $faktor_6$ % m = 1941 * 2401 % 2881 = 1764

Iteration $_7$: Bit im Exponenten gesetzt
$faktor_7$ = faktor62
$ergebnis_7$ = $ergebnis_6$ * $faktor_7$% m = 1764 * 2801 % 2881 = 49

Ergebnis:

$$6^{115} \equiv \pmod{2881} = 49$$

Für mehr Details siehe [7]

3.6 Die Sicherheit von RSA

Die Sicherheit des RSA-Algorithmus basiert auf der Tatsache, dass die Zerlegung von großen Dezimalzahlen in Primfaktoren sehr schwer ist. Nach dem heutigen Stand der Wissenschaft kennen wir keinen Algorithmus der in der Lage wäre dieses Problem zu lösen. Sogar mit den besten Algorithmen übersteigt die benötigte Zeit 100 Jahre. Solange das Problem der Primfaktorzerlegung großer Dezimalzahlen nicht gelöst ist, gilt RSA als eines der sichersten Verschlüsselungsverfahren.

3.6.1 Die Wahl der Primfaktoren p und q

Die Sicherheit der Verschlüsselung hängt selbstverständlich auch von den Primfaktoren $\boldsymbol{p}$ und $\boldsymbol{q}$ ab. Wollen wir die Faktorisierung des Produktes $\boldsymbol{p \cdot q}$ möglichst schwer machen, so sollen wir versuchen $\boldsymbol{p}$ und $\boldsymbol{q}$ so zu wählen, dass sie (ungefähr) gleich groß sind . Bei einem 2048 Bit Modul sollen wir versuchen die Wahl von $\boldsymbol{p}$ und $\boldsymbol{q}$ so zu treffen, dass beide 1024 Bit Zahlen werden.

3.6.2 Die Wahl von e und d

Bei der Wahl von $\boldsymbol{e}$ müssen wir auch vorsichtig vortgehen. Sie muss so gewählt werden, dass eine effiziente Verschlüsselung möglich ist. Dass $\mathbf{3}$ der minimale Wert von $\boldsymbol{e}$ ist bedarf keiner Erklärung. Wählt man $\boldsymbol{e = 3}$, so ist selbstverständlicherweise nur eine Quadrierung und eine Multplikation notwendig:

$$m^3 \pmod n = ((m^2 \pmod n) \cdot m) \pmod n$$

Wird $\boldsymbol{e}$ zu klein gewählt, so besteht die Gefahr einer ***Low-Exponent-Attacke***.

3.7 Beschleunigung mit dem Chinesischen Restsatz

Mit dem im 2. Kapitel vorgestellten chinesischen Restsatz können wir bei dem RSA-Verfahren Vorberechnungen durchführen und somit die benötigte Zeit halbieren. Im Falle des RSA-Verfarens haben wir einen Schlüsseltext $\boldsymbol{s}$, den privaten Schlüssel $\boldsymbol{d}$ und den ursprünglichen Klartext $\boldsymbol{m}$ und berechnen:

$$m_p = s^d \pmod p$$
$$m_q = s^d \pmod q$$

Als nächstes lösen wir die simultane Kongruenz:

$$m \equiv m_p \pmod p$$
$$m \equiv m_q \pmod q$$

Es ist klar, dass wir beim Lösen dieser Kongruenzen den erweiterten Euklidischen Algorithmus brauchen. Mit ihm berechnen wir $\boldsymbol{x_p}$ und $\boldsymbol{x_q}$ mit $\boldsymbol{x_p p + x_q q = 1}$. Die Entschlüsselung erfolgt dann mit:

$$m = (m_p x_q q + m_q x_p p) \pmod{p \cdot q}$$

Da die Zahlen $\boldsymbol{x_p p}$ und $\boldsymbol{x_q q}$ unabhängig von der zu entschlüsselnden Nachricht sind, können sie auch vorberechnet werden.

3.7.1 Beispiel

Betrachten wir unser Beispiel von oben:

$\boldsymbol{m_p = 6^{115} \pmod{43} = 6}$

$\boldsymbol{m_q = 6^{115} \pmod{67} = 49}$

$\boldsymbol{x_p = -14}$

$\boldsymbol{x_q = 9}$

$\boldsymbol{m = (6 \cdot 67 \cdot 9 - 49 \cdot 14 \cdot 43) \pmod{2881} = 49}$

Kapitel 4

Digitale Signatur

In unserer globalisierten, digitalen Welt spielen Briefe in der Alltagskommunikation fast keine Rolle mehr. Sie werden immer mehr durch E-Mails abgelöst. Es bedarf keiner Erklärung , welche Vorteile dieses Medium hat. E-Mails können natürlich auch für wichtige Dokumente ,wie zum Beispiel Kaufverträge und Rechnungen eingesetzt werden. Dank der digitalen Signatur können E-Mails für alle offiziellen Dokumente verwendet werden, da die Authentifikation [1] des Absenders in Form der Unterschrift problemlos ist. Diese Technik ist sehr verbreitet und ermöglicht den Geschäftsverkehr auch per Mail tätigen zu können.

Definition 10. *(Digitale Signatur)*
Eine digitale Signatur ist ein 5-Tupel $\boldsymbol{P, A, K, S, V}$

1. $\boldsymbol{P}$ *ist eine endliche Menge von Nachrichten*
2. $\boldsymbol{A}$ *ist eine endliche Menge von Signaturen*
3. $\boldsymbol{K}$ *ist der Schlüsselraum, d.h. eine endliche Menge möglicher Schlüssel*
4. $\boldsymbol{S}$ *ist die Menge der möglichen Signierfunktionen, sodass für alle* $\boldsymbol{k \in K}$ *ein* $\boldsymbol{sig_k \in S}$ *gibt , mit:*

$$\boldsymbol{sig_k : P \to A}$$

5. $\boldsymbol{V}$ *ist die Menge der Verifikationsfunktionen, sodasses für alle* $\boldsymbol{k \in K}$ *ein* $\boldsymbol{verify_k \in V}$ *gibt mit :*

$$\boldsymbol{verify_k : P \times A \to \{richtig\ ,\ falsch\}}$$

Für jede Nachricht $\boldsymbol{x \in P}$ *und für jede Signatur* $\boldsymbol{y \in A}$ *gilt:*

$$verify(x, y) = \begin{cases} richtig & falls \quad y = sig(x) \\ falsch & falls \quad y \neq sig(y) \end{cases}$$

Ein Paar $\boldsymbol{(x, y)}$ *mit* $\boldsymbol{x \in P}$ *und* $\boldsymbol{y \in A}$ *wird Signatur Nachricht genannt.*

Eine digitale Signatur muss folgende Merkmale erfüllen:

- Die Signatur wurde absichtlich unter das Dokument gesetzt
- Die Signatur kann nicht gefälscht werden

[1] aus dem Griechischen, bedeutet Urheber, Täter

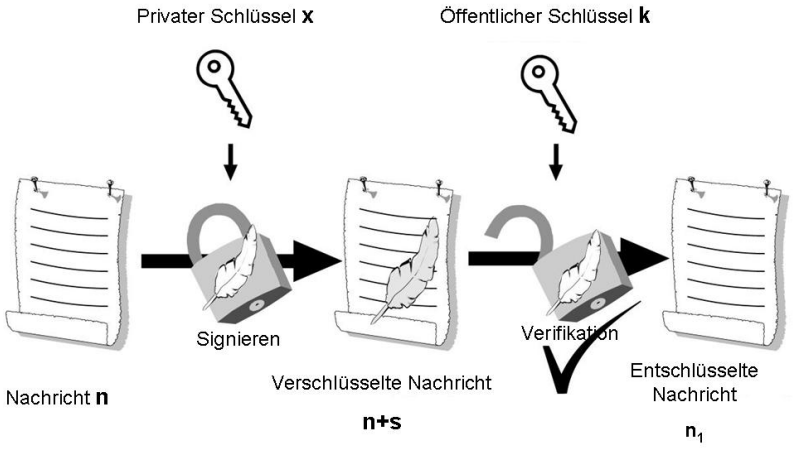

Abbildung 4.1: Digitale Unterschrift

- Die Unterschrift kann nicht auf andere Dokumente übertragen werden
- Nachträgliche Aenderungen im Dokument sind nicht möglich
- Die Unterschrift kann nachträglich nicht geleugnet werden

Würden wir versuchen mit Hilfe symmetrischer Verfahren digitale Signaturen zu erzeugen, so wären obige Bedingungen/Merkmale nicht erfüllt, da das Dokument im Nachhinein wieder verändert werden könnte. Damit sich bei der Schlüsselerzeugung symmetrische Verfahren natürlich ausgeschlossen.

"Unterschriften mit Tinte auf Papier erfüllen keine dieser Aussagen vollständig. Eine digitalen Signatur ist eine (recht große) Zahl, die im Zusammenhang mit einem digitalen Dokument ähnliche Eigenschaften aufweist wie eine Unterschrift von Hand auf einem Dokument aus Papier. Sie erfüllt alle bis auf die erste Aussage mit einer sehr hohen Sicherheit. Die erste Aussage kann prinzipiell nicht garantiert werden. Der zweite Punkt wird dadurch gelöst, dass ein Passwort (privater Schlüssel) in die digitale Signatur eingeht. Damit kann jeder seine eigene, eindeutige digitale Unterschrift generieren, sofern niemand sonst den Schlüssel kennt. Die Tücke bei der Sache ist jedoch, dass es auch ohne Kenntnis des Schlüssels möglich sein muss, die Echtheit der Unterschrift zu überprüfen. Dazu wird ein öffentlicher Schlüssel verwendet".

Überlegen wir uns, wie wir dies mathematisch formulieren können! Sei eine Nachricht $\boldsymbol{n}$ gegeben, die wir unterschreiben/signieren wollen. Dazu müssen wir einen öffentlich bekannten Schlüssel $\boldsymbol{k}$ und einen geheimen Schlüssel $\boldsymbol{x}$ besitzen. In diesem Fall entspricht das Unterschreiben der Berechnung einer Funkton $\boldsymbol{f(n,x)}$. Die Unterschrift bezeichnen wir mit $\boldsymbol{s = f(n,x)}$. Es ist selbstverständlich, dass der Absender die einzige Person ist, die den geheimen Schlüssel $\boldsymbol{x}$ kennt, so ist er auch der einzige der $\boldsymbol{s = f(n,x)}$ berechnen kann. Der Empfänger der Nachricht muss überprüfen, ob die erhaltene Nachricht echt ist. Um dies zu tun, muss der Empfänger eine Funktion "Verifikationsfunktion" $\boldsymbol{v}$ verwenden. Diese Funktion $\boldsymbol{v}$ ist klarerweise von $\boldsymbol{k}$ und $\boldsymbol{s}$ abhängig. Der Empfänger der Nachricht $\boldsymbol{n}$ berechnet also $\boldsymbol{n_1 := v(k,\ s)}$. Wenn am Ende der Rechnung $\boldsymbol{n_1 = n}$ herauskommt, ist die Unterschrift echt. Für Genaueres siehe [1] und [6]

Literaturverzeichnis

[1] Ertel W., 2007, *Angewandte Kryptographie*: Carl-Hanser Verlag

[2] J.Buchmann,2003*Einführung in die Kryptographie*:Springer Verlag

[3] K.Schmeh,2013*Kryptographie*:dpunkt.verlag

[4] A.Beutelspacher,H.B .Neumann, 2009*Kryptographie in der Theorie und Praxis*:Vieweg-Teubner Verlag

[5] P.Bundschuh,2002*Einführung in die Zahlentheorie*:Springer Verlag

[6] M.Zeppmeisel, 2006*Grundlagen der Quantenkryptographie*:Hausarbeit an der Ludwig-Maximilians-Universität München

[7] Dr.Christof Paar,2013 *Vorlesungsaufzeichnung zu Einführung in die Kryptographie und Datensicherheit*

[8] Thomas H. Cormen, Charles E. Leiserson, R.Rivest, 2010 *Algorithmen-Eine Einführung*:Oldenburg -Verlag

Abbildungsverzeichnis