Harald Löwe

Piraten, Mr. X und verschlüsselte Botschaften

Zusammenfassung: Der vorliegende Artikel ist aus einer Reihe von Projekten entstanden, die der Verfasser einige Male mit verschiedenen 9. Klassen, aber auch mit Oberstufenkursen durchgeführt hat, und die den Schülerinnen und Schülern eins der modernen kryptographischen Verfahren, nämlich der DIFFIE-HELLMANN-Schlüsselvereinbarung, näherbringen sollten.

Zu Beginn wird mit der betagten VIGÈNERE-Verschlüsselung die Unsicherheit der älteren kryptographischen Verfahren beleuchtet: Der KASISKI-Test zur Ermittlung der Schlüssellänge benötigt zwar als mathematische Fertigkeit lediglich die Berechnung des größten gemeinsamen Teilers natürlicher Zahlen, ist aber als Angriff auf die VIGÈNERE-Verschlüsselung höchst effektiv.

Nach dem so erbrachten Nachweis der Anfälligkeit althergebrachter Chiffren wird das Rechnen mit Restklassen als Grundlage für die einfacheren modernen Verfahren eingehend behandelt. Im Anschluss an die Diskussion der diskreten Exponentialfunktion kann dann das DIFFIE-HELLMANN-Verfahren erläutert werden, das eine Möglichkeit zum sicheren Vereinbarung eines Schlüssels über eine ungesicherte Leitung bietet. Der Nachweis der Sicherheit gegen einen brute-force-Angriff gehört hierbei ebenso zum Programm wie die Erläuterung eines schnellen Algorithmus zum Potenzieren, ohne den die praktische Umsetzung des DIFFIE-HELLMANN-Verfahrens nicht möglich wäre.

1. Die Vigenère-Verschlüsselung

BLAISE DE VIGENÈRE (1523-1596) war ein französischer Diplomat und Kryptograph, der in seinem Buch „Traicte de Chiffres“ die nach ihm benannte Verschlüsselungsmethode beschrieben hat. Nahezu 300 Jahre galt diese Verschlüsselung als so gut wie unknackbar; erst CHARLES BABBAGE (1791-1871) fand eine Methode zu einem systematischen Angriff. Da Babbage seine Erkenntnisse nicht veröffentlichte, gilt die vom preußischen Major FRIEDRICH WILHELM KASISKI (1805-1881) in seinem Buch „Die Geheimschriften und die Dechiffrir-Kunst“ publizierte Methode als erste Veröffentlichung. Wie wir sehen werden, war die VIGÈNERE-Verschlüsselung damit angreifbar und mit kurzen Schlüsselwörtern nicht mehr als ernsthaftes Verfahren zum Wahren von Geheimnissen verwendbar.

1.1 Ver- und Entschlüsseln mit VIGÈNERE

Piratenkapitän K. ist wieder mal auf große Beute aus: Er möchte in der Gegend der Insel Borneo mehrere Städte überfallen und plündern. Da seine eigenen Leute für solch ein Unternehmen zahlenmäßig nicht ausreichen, soll eine Nachricht mit der Bitte um Verstärkung an die anderen Piratenkapitäne geschickt werden. Diese Nachricht enthält Pläne und Treffpunkte – ein Abfangen durch den Gegner hätte daher fatale Folgen. Dies ist auch Kapitän K. klar, und so verschlüsselt er die Nachricht mit dem VIGÈNERE-Verfahren.

Hierzu benötigt man ein geheimes Schlüsselwort, das nicht zu kurz sein sollte, etwa „`MatheLok`“[1]. Anschließend wird dieses über den Klartext geschrieben, wobei das Schlüsselwort zu wiederholen ist. Für die Nachricht „`Piraten und Spione`“ erhalten wir damit

[1] „Mathe-Lok“ ist der Name des Mathematiklehrerfortbildungszentrums der Technischen Universität Braunschweig.

Schlüssel: `MATHELO KMA THELOK`
Klartext: `piraten und spione;`

wobei wir zur Verdeutlichung das Schlüsselwort in Großbuchstaben und den Klartext in Kleinbuchstaben geschrieben haben. Jetzt ist der Zeitpunkt für den Auftritt des „VIGÈNERE-Quadrats“ gekommen:

	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
A	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
B	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	a
C	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
D	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	x
E	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
F	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
G	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
H	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	x	d	e	f	g
I	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
J	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
K	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
L	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	x	d	e	f	g	h	i	j	k
M	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
N	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
O	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
P	p	q	r	s	t	u	v	w	x	y	z	a	b	x	d	e	f	g	h	i	j	k	l	m	n	o
Q	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
R	r	s	t	u	v	w	x	y	z	a	b	x	d	e	f	g	h	i	j	k	l	m	n	o	p	q
S	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
T	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
U	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
V	v	w	x	y	z	a	b	x	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
W	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
X	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
Y	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
Z	z	a	b	x	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y

Die eigentliche Verschlüsselung erfolgt mit Hilfe dieses VIGÈNERE-Quadrates buchstabenweise. Der erste Klartextbuchstabe des obigen Beispiels etwa ist ein „p“; über diesem steht der zuständige Schlüsselbuchstabe, ein „M“. Im VIGÈNERE-Quadrat tritt nun in Zeile „M“ (Schlüssel) und Spalte „p“ (Klartext) ein „b“ – und eben dieses „b“ ist der erste Buchstabe des Geheimtextes. Für jeden weiteren Buchstaben des Klartextes erhält man den zugehörigen Buchstaben des Geheimtextes auf die gleiche Weise.

Aufgabe 1: Führen Sie die Verschlüsselung im oben beschriebenen Beispiel vollständig durch.

Schlüssel: `MATHELO KMA THELOK`
Klartext: `piraten und spione`
Geheimtext: `b_____ ___ ______`

Die Entschlüsselung ist bei Kenntnis des Schlüssels kein Problem. Findet man zum Beispiel zu einem „s“ im Geheimtext als zugehörigen Schlüsselbuchstaben ein „M“ vor, so muss man nur in der Zeile „M“ des VIGÈNERE-Quadrates das „s“ aufsuchen. Dieses steht in der mit „g“ überschriebenen Spalte – und das ist natürlich der zugehörige Klartextbuchstabe.

Aufgabe 2: Entschlüsseln Sie den folgenden Geheimtext:

Schlüssel: `MATHELO KMATHELOK`
Geheimtext: `sealmxs xmcaymnvd`
Klartext: `g_____ _________`

1.2 Der KASISKI-Test

Pirat K. schickt die folgende verschlüsselte Botschaft los:

```
lqixm vrunv nqyah fmokb qtzoi gwraf aaoek mezrq jnwzv nqeah kylbi fhmed
qhvtf znliy mjfrg ibmie xqhbg bonxl ztvow bvkuz zsvlj pajek bnjno xbqkp
vjcap hhaou xfrgn rleia mezrq hvtfu rxmkx vjeer kcwvr wpvrp ovhrk qwinx
skmlz eqieq mltxx rcfjc aerih vxuvv covnq avgfd oxkyx kysmb epath qdhmc
esxvr bfleu pvrpv nqzrg akevo kxuob xvzjl rdkiv tkbxq mpwfd oxgui monhg
umlrr tysmp znpiv hmexw efhfz mqfrg mzcab qtzoi gwraf aaoek tqeiz ikhky
iyjrm lvsqk bjdvr gihga ovhrk xvuog ifxme xzifx kytxx klwid xrkhq edqbn
aafxt yseij sxrkm uwg
```

Diese wird auch prompt von einem Schiff des Gouverneurs von Indonesien abgefangen. Davon kann unser Piratenkapitän zwar nichts ahnen, aber er rechnet trotzdem zumindest mit der Möglichkeit. Mit Blick auf die von ihm für sicher befundene Verschlüsselung allerdings fühlt er sich vollkommen sicher und läuft wohlgemut zum Treffen mit den Kollegen aus.

Im Hauptquartier der Flotte von Indonesien herrscht dagegen große Aufregung. Endlich einmal konnte eine der Nachrichten von K. abgefangen werden – sollte es nun möglich werden, diesen Gegner unschädlich zu machen? Insbesondere wird im Stab des Admirals der junge Leutnant H. mit der Aufgabe betraut, den abgefangenen Text möglichst rasch zu entschlüsseln. Dieser hat sich mit K. lange Zeit beschäftigt und weiß, dass dieser für einen Pirat recht höflich ist und seine Briefe grundsätzlich mit dem Kürzel „MfG“ beendet[2]. Außerdem weiß der Leutnant, dass es sich um eine VIGÈNERE-Verschlüsselung handelt[3]. Damit aber kann er die letzten drei Buchstaben „uwq“ des abgefangenen Textes dem Klartext „mfg“ zuordnen. Der hierbei verwendete Teil des Schlüsselwortes muss „ira“ sein – denn nur „ira“ macht aus „mfg“ den Geheimtext „uwq“! Hätte man nun noch die Länge des Schlüsselwortes, so könnte man wenigstens einige Teile des Geheimtextes dechiffrieren und dann sehen, ob man weitere Informationen gewinnen kann. Aber natürlich war K. nicht so freundlich, die Schlüssellänge bekannt zu geben!

Glücklicherweise erinnert sich H. an das Buch von KASISKI: Wenn in einem VIGÈNERE-chiffrierten Text zweimal die gleiche Buchstabenkombination auftritt, dann spricht einiges dafür, dass hier der gleiche Klartext mit dem gleichen Teil des Schlüsselwortes behandelt wurde. Der Abstand zwischen den beiden Buchstabenkombinationen muss dann ein Vielfaches der Schlüssellänge sein. Leutnant H. sucht daher Folgen gleicher Buchstaben und markiert diese:

```
lqixm vrunv nqyah fmokb qtzoi gwraf aaoek mezrq jnwzv nqeah kylbi fhmed
qhvtf znliy mjfrg ibmie xqhbg bonxl ztvow bvkuz zsvlj pajek bnjno xbqkp
vjcap hhaou xfrgn rleia mezrq hvtfu rxmkx vjeer kcwvr wpvrp ovhrk qwinx
skmlz eqieq mltxx rcfjc aerih vxuvv covnq avgfd oxkyx kysmb epath qdhmc
esxvr bfleu pvrpv nqzrg akevo kxuob xvzjl rdkiv tkbxq mpwfd oxgui monhg
umlrr tysmp znpiv hmexw efhfz mqfrg mzcab qtzoi gwraf aaoek tqeiz ikhky
iyjrm lvsqk bjdvr gihga ovhrk xvuog ifxme xzifx kytxx klwid xrkhq edqbn
aafxt yseij sxrkm uwg
```

Durch Nachzählen ergibt sich:

- Abstand zwischen den *kursiven* Wörtern: $105 = 3 \cdot 5 \cdot 7$
- Abstand zwischen den **fett gedruckten** Wörtern: $315 = 3^2 \cdot 5 \cdot 7$

[2] Wer meint, dass so etwas nur ungebildeten Piraten passieren kann: auch im 2. Weltkrieg haben die englischen Kryptoanalytiker in Bletchley stets wiederkehrende Anreden („An F.d.U.West“) oder stereotypische Standardtexte („Keine besonderen Vorkommnisse“) zum Kryptoangriff gegen die Enigma erfolgreich genutzt.

[3] Ein Credo der Kryptographie lautet, man solle stets annehmen, dass der Gegner das Verfahren kennt und ihm nur das Schlüsselwort fehlt.

- Abstände zwischen den unterstrichenen Wörtern: $35 = 5 \cdot 7$
  $168 = 2^3 \cdot 3 \cdot 7$
  $42 = 2 \cdot 3 \cdot 7.$

Nun könnten zwei gleiche Dreiergruppen durchaus noch auf Zufall beruhen. Daher nimmt H. nun an, dass die Schlüssellänge 3, 7 oder 21 ist. Obwohl ihm die Länge 3 doch verdächtig kurz vorkommt, probiert er diesen Fall kurz aus: Der Text ist $438 = 3 \cdot 146$ Buchstaben lang, so dass über „uwg“ das Schlüsselwort „ira“ (und nicht etwa „rai“ oder „air“) steht. Entschlüsseln mit „ira“ liefert

dzipv vjdnn wqqjh xvock qlioa pwjjf sjowt mwiri snoiv fzesq *etc.*

als Anfang des Klartextes – damit scheidet 3 als Schlüssellänge aus, und die 7 ist die nächste Möglichkeit.

1.3 Entschlüsseln durch Raten

Bei einer Schlüsselwortlänge von 7 muss der Text in 7er-Blöcke aufgeteilt werden. Wegen $438 = 7 \cdot 62 + 4$ sind die letzten vier Buchstaben „muwg“ der Beginn des letzten (und nicht mehr vollständigen) 7er-Blocks. Daher lautet das Schlüsselwort „⋆ira⋆⋆⋆“, wobei „⋆“ für einen unbekannten Buchstaben steht. Der in 7er-Blöcke unterteilte Geheimtext und der mit den drei bekannten Buchstaben des Schlüsselwortes wenigstens teilweise dechiffrierte Text sehen jetzt so aus:

Klartext:

```
⋆irx⋆⋆⋆ ⋆fen⋆⋆⋆ ⋆xvo⋆⋆⋆ ⋆rxi⋆⋆⋆ ⋆xja⋆⋆⋆ ⋆wir⋆⋆⋆ ⋆ren⋆⋆⋆ ⋆chl⋆⋆⋆ ⋆end⋆⋆⋆
⋆xin⋆⋆⋆ ⋆bor⋆⋆⋆ ⋆anx⋆⋆⋆ ⋆txn⋆⋆⋆ ⋆nxw⋆⋆⋆ ⋆ris⋆⋆⋆ ⋆sse⋆⋆⋆ ⋆fxx⋆⋆⋆ ⋆nsc⋆⋆⋆
⋆sxu⋆⋆⋆ ⋆fal⋆⋆⋆ ⋆wir⋆⋆⋆ ⋆xdr⋆⋆⋆ ⋆nse⋆⋆⋆ ⋆oer⋆⋆⋆ ⋆hxv⋆⋆⋆ ⋆orn⋆⋆⋆ ⋆die⋆⋆⋆
⋆eut⋆⋆⋆ ⋆xsc⋆⋆⋆ ⋆zex⋆⋆⋆ ⋆gen⋆⋆⋆ ⋆xmo⋆⋆⋆ ⋆chs⋆⋆⋆ ⋆schr⋆⋆ ⋆uns⋆⋆⋆ ⋆xue⋆⋆⋆
⋆hen⋆⋆⋆ ⋆ste⋆⋆⋆ ⋆mxb⋆⋆⋆ ⋆dad⋆⋆⋆ ⋆ckx⋆⋆⋆ ⋆xmo⋆⋆⋆ ⋆exn⋆⋆⋆ ⋆dar⋆⋆⋆ ⋆hin⋆⋆⋆
⋆enx⋆⋆⋆ ⋆xim⋆⋆⋆ ⋆eic⋆⋆⋆ ⋆rxi⋆⋆⋆ ⋆xja⋆⋆⋆ ⋆ini⋆⋆⋆ ⋆chi⋆⋆⋆ ⋆des⋆⋆⋆ ⋆ver⋆⋆⋆
⋆sxv⋆⋆⋆ ⋆ndo⋆⋆⋆ ⋆enx⋆⋆⋆ ⋆cht⋆⋆⋆ ⋆ord⋆⋆⋆ ⋆ind⋆⋆⋆ ⋆sox⋆⋆⋆ ⋆ass⋆⋆⋆ ⋆mfg
```

Das sähe ja schon ganz gut aus, wäre da nicht das häufige Auftreten des Buchstaben x, das den Leutnant stutzig werden lässt. Er überlegt, dass diese x im Klartext einfach für ein Leerzeichen stehen könnte. So ein Leerzeichen steht bestimmt auch vor der Grußformel „mfg“. Folglich wird der Klartext „x“ durch den ersten Buchstaben des Schlüsselwortes in den Geheimtext „m“ übertragen – dieser erste Buchstabe muss folglich ein „p“ sein. Außerdem kann in der fett markierten Gruppe „⋆xsc⋆⋆⋆“ des Klartextes auf das „c“ nur ein „h“ folgen, womit wir „t“ als fünften Buchstaben des Schlüsselwortes erraten hätten.

Bislang hat H. „pirat⋆⋆“ als Schlüsselwort. Nach erneuter Dechiffrierung der bekannten Teile des Textes, Auflösen der 7er-Blocks sowie Ersetzen jedes „x“ durch ein Leerzeichen erhält er

```
wir t⋆⋆ffen ⋆⋆s vor⋆⋆er in⋆⋆l jav⋆⋆ wir ⋆⋆hren ⋆⋆schli⋆⋆send ⋆e ins⋆⋆
born⋆⋆ an ⋆⋆rt ne⋆⋆en wi⋆⋆frisc⋆⋆asser⋆⋆uf i⋆⋆ansch⋆⋆ss ue⋆⋆rfall⋆⋆ wir ⋆⋆e
dre⋆⋆insel⋆⋆noerd⋆⋆ch vo⋆⋆borne⋆⋆ die ⋆beute⋆⋆n sch⋆⋆tze b⋆⋆ngen ⋆⋆r
moe⋆⋆ichst⋆⋆asch ⋆⋆ unse⋆⋆m ueb⋆⋆chen ⋆⋆rstec⋆⋆im be⋆⋆udadr⋆⋆eck ⋆⋆h
moe⋆⋆te no⋆⋆ dara⋆⋆ hinw⋆⋆sen d⋆⋆s im ⋆⋆reich⋆⋆er in⋆⋆l
jav⋆⋆einig⋆⋆schif⋆⋆ des ⋆uvern⋆⋆rs vo⋆⋆indon⋆⋆ien
g⋆⋆ichte⋆⋆worde⋆⋆sind⋆⋆lso a⋆⋆passe⋆⋆ mfg
```

Hier muss man nicht einmal sehr lange nachdenken, um bei Kenntnis der deutschen Sprache auf den Inhalt der Nachricht zu kommen. Schließlich benötigen wir nur noch zwei Buchstaben des Schlüsselwortes!

Aufgabe 3: Ermitteln Sie die fehlenden Buchstaben des Schlüsselwortes und dechiffrieren Sie die Nachricht vollständig. Ein „xx“ im Klartext steht übrigens für das Ende eines Satzes.

Durch die stereotype Endfloskel und ein im Verhältnis zur Länge des Textes viel zu kurzes Schlüsselwort konnte also die Nachricht des Piratenkapitäns recht rasch entschlüsselt werden – ein Fehler, der sich für K. fatal auswirken wird…. Aber auch ohne Kenntnisse des Schlusses sind nach dem Erraten der Schlüssellänge und einer Häufigkeitsanalyse derjenigen Buchstaben, die jeweils mit dem gleichen Buchstaben des Schlüsselworts verschlüsselt werden, einem Kryptoangriff gut zugänglich, wenn nur die Nachrichtenlänge recht groß ist. In diesem Fall ist es – gerade durch den heute möglichen Einsatz von Rechnern[4] – mit der Sicherheit des VIGÈNERE-Verfahrens nicht weit her! Die Wahl etwa eines ganzen Buchs als Schlüsselwort bringt ebenfalls keinen großen Gewinn, schlagen doch Regelmäßigkeiten der Sprache auf den Geheimtext durch (so wird etwa ein x im Klartext nur selten auf ein y im Schlüsselwort stoßen, und die Buchstabenkombination „dfgh“ wird wohl nie vorkommen) und macht das Verfahren gegen statistische Methoden anfällig.

Abhilfe schafft die Verwendung einer zufälligen Buchstabenfolge als Schlüsselwort, wenn diese (mindestens) die Länge des Klartextes hat. Dann aber hat man Probleme, diesen Schlüssel an den Empfänger sicher zu übermitteln. Für bessere Methoden benötigt man ein wenig Mathematik, die wir im folgenden Abschnitt beschreiben wollen.

2. Rechnen mit Resten

Viele moderne Methoden der Kryptographie verwenden das Rechnen mit Resten, das wir jetzt vorstellen wollen. Wir beginnen mit einigen Schreibweisen: Bekanntlich hinterlässt 15 bei Division durch 4 den Rest 3; wir schreiben für diesen Sachverhalt kurz $15 \equiv 3 \bmod 4$, wobei wir dies als „15 ist kongruent zu 3 modulo 4“ lesen wollen. Allgemein schreiben wir für eine natürliche Zahl $n \geq 2$ und zwei ganze Zahlen a, b kurz

$$a \equiv b \bmod n,$$

wenn a und b bei Division durch n den gleichen Rest besitzen. Dies ist offenbar gleichbedeutend mit der Aussage, dass die Differenz $a - b$ durch n teilbar ist. Dies wiederum ist gleichwertig zu: es gibt ein ganzzahliges k mit der Eigenschaft $a - b = k \cdot n$ bzw. $a = b + k \cdot n$. Wir halten daher fest

Satz 1: Genau dann gilt $a \equiv b \bmod n$, wenn es eine ganze Zahl k mit $a = b + k \cdot n$ gibt.

So ist etwa $-13 \equiv -13 + 3 \cdot 6 = 5 \bmod 6$, wobei wir Satz 1 soeben verwendet haben. Weitere nützliche Rechenregeln sind im folgenden Satz gesammelt:

Satz 2: Es gelte $a \equiv u \bmod n$ und $b \equiv v \bmod n$. Weiterhin sei m eine natürliche Zahl. Dann folgt:

a. $a + b \equiv u + v \bmod n$.
b. $a - b \equiv u - v \bmod n$.
c. $a \cdot b \equiv u \cdot v \bmod n$.
d. $a^m \equiv u^m \; mod \; n$.

Beweis von a.: Aus den Voraussetzungen folgt die Existenz ganzer Zahlen k, l mit $a = u + k \cdot n$ und $b = v + l \cdot n$. Daher ist $a + b = (u + k \cdot n) + (v + l \cdot n) = (u + v) + (k + l) \cdot n \equiv u + v \bmod n$, wie behauptet.

Aufgabe 4: Weisen Sie die Behauptungen b., c. und d. aus Satz 2 nach.

[4] Verfahren zum automatisierten Verschlüsseln und Entschlüsseln nach der VIGÈNERE-Verschlüsselung findet man im Internet, z.B. unter http://gc.de/gc/vigenere/. Auch für den Kryptoangriff gibt es überraschend gute elektronische Helferlein, siehe etwa http://weddige.eu/tools/kryptix/.

Da wir vor allem an den Resten interessiert sind, legen wir als Abkürzung fest: $[a]_n$ bezeichne diejenige natürliche Zahl[5] zwischen 0 und $n-1$, für die $a \equiv [a]_n \bmod n$ gilt; d.h. $[a]_n$ ist der Rest von a bei Division durch n. So ist zum Beispiel $[-13]_6 = 5$, denn $-13 \equiv 5 \bmod 6$. Und natürlich sind jetzt $a \equiv b \bmod n$ und $[a]_n = [b]_n$ nur zwei verschiedene Möglichkeiten, den gleichen Sachverhalt mitzuteilen: a und b besitzen bei Division durch n den gleichen Rest.

Wir halten kurz inne, um über die Notation $[a]_n$ für den Rest etwas nachdenken. Natürlich kann man $[\cdot]_n$ als Funktion auffassen, die der ganzen Zahl a ihren Rest $[a]_n$ bei Division durch n zuordnet. Sinnvoller ist es allerdings, die Zahl $[a]_n$ als *Stellvertreter* für *sämtliche* Zahlen mit Rest $[a]_n$ zu betrachten. So steht die $5 = [5]_6 = [17]_6 = [-13]_6 = \cdots$ für *alle* Zahlen mit Rest 5 bei Division durch 6. Zur Vermeidung von Missverständnissen werden wir aus diesem Grund die Notation „$[5]_6$“ anstelle von „5“ vorziehen, womit hoffentlich das jetzt zu behandelnde Rechnen mit Resten klarer wird. Wir definieren Summen, Produkte und Potenzen von Resten durch

- $[a]_n + [b]_n = [a+b]_n$; $[a]_n - [b]_n = [a-b]_n$
- $[a]_n \cdot [b]_n = [a \cdot b]_n$ und
- $[a]_n^x = [a^x]_n$ (für natürliche Zahlen x).

So ist beispielsweise $[3]_{13} + [2]_{13} = [5]_{13}$ und $[5]_6 \cdot [4]_6 = [20]_6 = [2]_6$. Wegen $[5]_6 = [-13]_6$ und $[4]_6 = [10]_6$ könnten wir die letzte Gleichung auch als $[-13]_6 \cdot [10]_6 = [-130]_6 = [2]_6$ lesen. Diese Gleichung besagt daher viel mehr als es auf den ersten Blick sichtbar, nämlich: Das Produkt einer Zahl a mit Rest 5 und einer Zahl b mit Rest 4 bei Division durch 6 hat immer den Rest 2. Glücklicherweise haben wir dies schon gelernt: Ist $a \equiv 5 \bmod 6$ sowie $b \equiv 4 \bmod 6$, so gilt nach Satz 2c.

$$a \cdot b \equiv 5 \cdot 4 = 20 \equiv 2 \bmod 6.$$

Allgemein besagt Satz 2c. in der neuen Notation: Aus $[a]_n = [u]_n$ und $[b]_n = [v]_n$ folgt $[a \cdot b]_n = [u \cdot v]_n$. Da haben wir Glück gehabt! Das Ergebnis der Rechnung $[a]_n \cdot [b]_n = [a+b]_n$ hängt also *nicht* davon ab, welche Vertreter u und v aller Zahlen mit Rest $[a]_n$ bzw. $[b]_n$ zur Berechnung genommen werden. Zu dieser „Unabhängigkeit von der Wahl der Vertreter“ sagt der Mathematiker kurz „Die Festlegung $[a]_n \cdot [b]_n = [a \cdot b]_n$ ist *wohldefiniert*“. Natürlich gilt das soeben Gesagte auch, wenn man „$\cdot$“ durch „$+$“ ersetzt.

Aufgabe 5: Berechnen Sie:

a.	$[2]_5 + [4]_5$	$[6]_{17} \cdot [3]_{17}$	$[3]_7 + [4]_7$	$[2]_6 \cdot [3]_6$	$[3]_7^4$
b.	$[729 + 865]_5$	$[697 \cdot 528]_5$	$[17^{10}]_5$	$[14^{1.000}]_5$	

Aufgabe 6: Was bedeutet „Die Festlegung $[a]_n^x = [a^x]_n$ ist wohldefiniert“? Wie zeigt man das?

Aufgabe 7: Bekanntlich steht der Bruch $\frac{a}{b}$ stellvertretend für alle Lösungen x der Gleichung $b \cdot x = a$. Wieso ist es keine gute Idee, die „Schüleraddition“ $\frac{a}{b} \oplus \frac{u}{v} = \frac{a+u}{b+v}$ von Brüchen einzuführen?

Zum Abschluss dieses Abschnitts steht noch der Satz von EULER auf dem Programm. Hierzu benötigen wir die EULERsche φ-Funktion, die einer natürlichen Zahl $n \geq 1$ die Anzahl $\varphi(n)$ der zu n teilerfremden Zahlen zwischen 1 und $n-1$ zuordnet. Es ist daher

- $\varphi(5) = 4$, denn es gibt genau 4 zu 5 teilerfremde Zahlen unterhalb 5, nämlich $1, 2, 3, 4$.
- $\varphi(10) = 4$, denn es gibt genau 4 zu 10 teilerfremde Zahlen unterhalb 10, nämlich $1, 3, 7, 9$.
- $\varphi(16) = 8$, denn es gibt genau 8 zu 16 teilerfremde Zahlen unterhalb 16, nämlich $1, 3, 5, 7, 9, 11, 13, 15$

Für eine Primzahl n ist jede kleinere Zahl teilerfremd zu n; wir erhalten daher

Satz 3: Ist n eine Primzahl, so gilt $\varphi(n) = n - 1$.

[5] Zugegeben: Es wäre mathematisch deutliche sauberer, die übliche Definition der Restklasse $[a]_n$ als die Menge derjenigen ganzen Zahlen b mit $b \equiv a \bmod n$ einzuführen. Bitte vergessen Sie aber nicht, dass die Zielgruppe des Projekts Schülerinnen und Schüler der Mittelstufe sind.

Aufgabe 8: Vorgelegt sind verschiedene Primzahlen p und q. Bestimmen Sie $\varphi(p \cdot q)$.

Damit haben wir die notwendigen Grundlagen zum Verständnis des folgenden, überaus nützlichen Satzes gelegt.

Satz von EULER: Sind b und n teilerfremd, so gilt $b^{\varphi(n)} \equiv 1 \bmod n$ bzw. $[b]_n^{\varphi(n)} = [1]_n$.

Den *Beweis* zerlegen wir in einige mundgerechte Häppchen:

a) Gilt $[b]_n^u = [b]_n^v$ bzw. $b^u \equiv b^v \bmod n$ für Exponenten $1 \le u < v$, so gilt $b^{v-u} \equiv 1 \bmod n$.
Aus der Voraussetzung erhalten wir nämlich die Teilbarkeit von $b^v - b^u = b^u \cdot (b^{v-u} - 1)$ durch n. Da b und n teilerfremd sind, muss n ein Teiler von $b^{v-u} - 1$ sein, woraus die Behauptung bereits folgt.

b) Es gibt ein $x \ge 1$ mit $b^x \equiv 1 \bmod n$.
Für jedes der unendlichen vielen $x \ge 1$ ist $[b^x]_n$ einer der endlich vielen Reste $1, \dots, n-1$. Daher gibt es sicherlich verschiedene Zahlen $v > u \ge 1$ mit $[b]_n^u = [b]_n^v$ bzw. $b^u \equiv b^v \bmod n$. Mit Teil a) folgt $b^x \equiv 1 \bmod n$ für $x = v - u$.

Es sei x_b die kleinste Zahl $x \ge 1$ mit $b^x \equiv 1 \bmod n$. Es gibt zwischen 1 und $n-1$ insgesamt $\varphi(n)$ viele Zahlen, die zu n teilerfremd sind. Wie die Schülerinnen und Schüler einer Jahrgangsstufe werden wir diese Zahlen jetzt in „Klassen“ einteilen: Schüler $[k]_n$ kommt zusammen mit $[k \cdot b]_n, [k \cdot b^2]_n, \dots, [k \cdot b^{x_b-1}]_n$ in die Klasse mit der Bezeichnung $\mathbf{K}([k]_n)$. Aus $[k \cdot b^{x_b-1}]_n \cdot [b]_n = [k]_n \cdot [b^{x_b}]_n = [k]_n$ entnehmen wir, dass sich die Klasse $\mathbf{K}([k]_n)$ nicht verändert, wenn man jedes ihrer Elemente mit derselben Potenz $[b^u]_n$ multipliziert.

c) Jede der Klassen hat genau x_b Elemente.
Ist nämlich $[k \cdot b^u]_n = [k \cdot b^v]_n$ mit $0 \le u \le v \le x_b - 1$, so ist n ein Teiler von $k \cdot b^v - k \cdot b^u = k \cdot b^u \cdot (b^{v-u} - 1)$. Sowohl k als auch b sind zu n teilerfremd, so dass n ein Teiler von $b^{v-u} - 1$ bzw. $b^{v-u} \equiv 1 \bmod n$ sein muss. Wir beachten $0 \le v - u < x_b$ und erhalten aus der Minimalität von x_b sofort $v = u$, was die Behauptung zeigt.

d) Zwei verschiedene Klassen besitzen keine gemeinsamen Elemente.
Ist nun $[l]_n$ in der Klasse $\mathbf{K}([k]_n)$, so ist auch $[l \cdot b^u]_n$ in dieser Klasse, die sich ja bei Multiplikation ihrer Elemente mit $[b^u]_n$ nicht verändert. Damit ist $\mathbf{K}([l]_n)$ eine Teilmenge von $\mathbf{K}([k]_n)$. Beide Klassen haben x_b Elemente und müssen daher übereinstimmen.

e) Die Zahl x_b ist ein Teiler von $\varphi(n)$.
Jede der $\varphi(n)$ zu n teilerfremden Zahlen zwischen 1 und $n-1$ liegt nach d) in genau einer der Klassen, von denen es insgesamt d Stück geben möge. Nach c) besitzt jede der Klassen genau x_b Elements, woraus $\varphi(n) = d \cdot x_b$ folgt. Das zeigt aber unmittelbar die Teilbarkeit von $\varphi(n)$ durch x_b.

f) Es gilt $[b]_n^{\varphi(n)} = [1]_n$.
Nach e) ist $\varphi(n) = d \cdot x_b$ und folglich $[b]_n^{\varphi(n)} = \left([b]_n^{x_b}\right)^d = [1]_n^d = [1]_n$. Damit ist alles gezeigt.

Der soeben eingeführte minimale Exponent $x_b \ge 1$ mit $[b]_n^x = [1]_n$ wird übrigens als die *Ordnung* von $[b]_n$ bezeichnet; die Ordnung von $[b]_n$ teilt also $\varphi(n)$. Eine Zahl $[b]_n$ mit der maximal möglichen Ordnung $x_b = \varphi(n)$ heißt auch *primitives Element modulo n.* Solche primitiven Elemente gibt es außer für $n = 2$ und $n = 4$ genau dann, wenn n eine ungerade Primzahlpotenz oder das Doppelte einer solchen ist. Es ist aber nicht klar, wie man sich – außer durch Probieren – solche primitiven Elemente beschaffen kann. So ist es ein immer noch ungelöstes Problem, ob $[2]_n$ für unendlich viele Primzahlen n eine primitives Element ist.

Aufgabe 9: Finden Sie ein primitives Element modulo 25, und zeigen Sie, dass es keine primitiven Elemente modulo 15 oder modulo 16 gibt.

Für Primzahlen n geht der Satz von EULER in den *kleinen Satz von FERMAT* über: Für eine Primzahl n und einer Zahl $1 \le b \le n-1$ gilt stets $b^{n-1} \equiv 1 \bmod n$.

Aufgabe 10: Berechnen Sie $[17^{2.015}]_5$.

3. Das DIFFIE-HELLMANN-Verfahren

Das Ziel des DIFFIE-HELLMANN-Verfahrens klingt zunächst recht einfach: Alice und Bob (so heißen die beiden entfernt voneinander wohnenden Rechnerbenutzer in der Kryptographie mittlerweile fast immer) möchten über eine ungesicherte Leitung eine Zahl als Schlüssel vereinbaren. Moment – was heißt hier „ungesichert“? Nichts anderes als dass Mr. X, der an den Schlüssel gelangen möchte, *sämtliche* Nachrichten von Alice und Bob mithören kann! Das Übermitteln des Schlüssels über die Leitung verbietet sich daher von selbst. Aber auch mit dem Verschlüsseln des Schlüssels ist nichts gewonnen, denn wie sollte der hierzu nötige Schlüssel übertragen werden!

3.1 Die diskrete Exponentialfunktion

Auch für das Rechnen mit Resten kann man eine Exponentialfunktion betrachten: Für eine natürliche Zahl n und einen der zugehörigen, zu n teilerfremden Reste b, $2 \leq b \leq n-1$, betrachtet man die durch $f(x) = [b^x]_n$ beschriebene Funktion, die x (mit $0 \leq x \leq n-1$) auf den Rest von b^x bei Division durch n abbildet. Im Gegensatz zur reellen Exponentialfunktion erhält man allerdings hierbei einen im Allgemeinen recht wüst aussehenden Gesellen. Ein Beispiel für eine derartige Exponentialfunktion mit $n = 2003$ und $b = 3$ ist in Abbildung 1a zu sehen.

Allerdings gibt es durchaus neben den offensichtlichen Basen $b = 1$ und $b = [-1]_n = n-1$ auch Wahlen von b, für die die diskrete Exponentialfunktion ein nicht ganz so chaotisches Aussehen besitzt. So erhält man für $n = 2003$ und $b = 2$ die in Abbildung 1b wiedergegebene, recht regelmäßig aussehende Punktreihe. Das ist auch kein Wunder: Wegen $2^{286} \equiv 1 \bmod 2003$ wiederholt sich $[2^x]_{2003}$ ab $x = 286$ immer wieder. Im Hinblick auf die Diskussion des letzten Abschnitts wird man daher möglichst primitives Element modulo einer großen Primzahl n oder wenigstens solche b mit recht großer Ordnung nehmen.

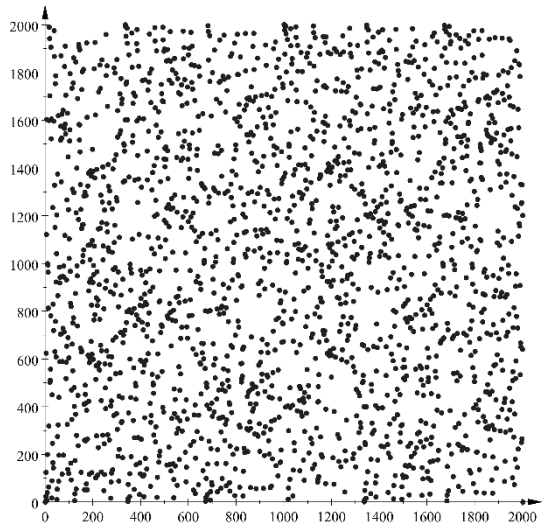


Abbildung 1a: Exponentialfunktion $[3^x]_{2003}$

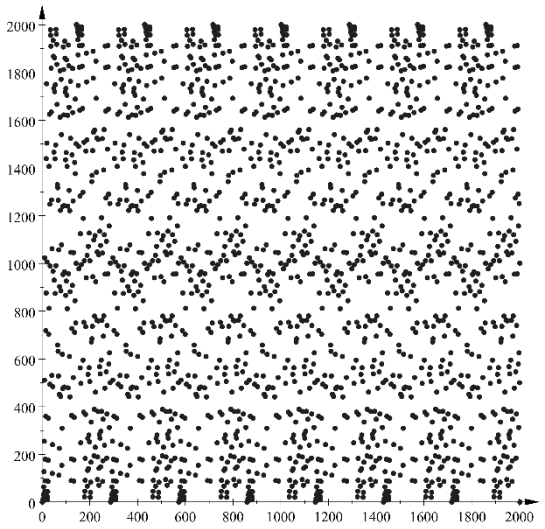


Abbildung 1b: Exponentialfunktion $[2^x]_{2003}$

In der Tat sind die zu den diskreten Exponentialfunktionen gehörenden „diskreten Logarithmen“ auch bislang nicht in den Griff zu kriegen – noch gibt es keinen schnellen Algorithmus, um die Gleichung $b^x \equiv u \bmod n$ effektiv zu lösen. In Spezialfällen (etwa wenn $\varphi(n)$ nur relativ kleine Primteiler besitzt) kann der Aufwand erheblich reduziert werden (und für das Lösen der Gleichung $2^x \equiv 8 \bmod 2003$ braucht man sich überhaupt nicht anzustrengen. Allgemein aber ist man auf das Raten einer Lösung angewiesen.

Das (derzeitige) Fehlen eines schnellen Algorithmus zur Lösen der Gleichung $b^x \equiv u \bmod n$ nach x veranlasste die beiden Kryptographen Whitfield DIFFIE und Martin HELLMANN die im folgenden Abschnitt geschilderten Lösung für die Schlüsselvereinbarung in einem 1976 publizierten Artikel vorzuschlagen.

3.2 Schlüsselvereinbarung nach DIFFIE und HELLMANN

Alice und Bob vereinbaren eine ausreichend große Primzahl n sowie eine weitere Zahl b über den ungesicherten Kanal. Unser feindlicher Lauscher, Mr. X, kennt daher beide Zahlen. Anschließend wählt Alice eine geheime Zahl x und berechnet den Rest $u = [b^x]_n$. Bob wählt eine weitere, nur ihm bekannte Zahl y und berechnet $v = [b^y]_n$. Anschließend werden die – vermutlich verschiedenen – Zahlen u und v ausgetauscht. Wir beachten

$$[u^y]_n = [(b^x)^y]_n = [b^{x\cdot y}]_n = [b^{y\cdot x}]_n = [(b^y)^x]_n = [v^x]_n$$

und erkennen, dass sowohl Alice durch die Rechnung $S = [v^x]_n$ als auch Bob mit Hilfe von $S = [u^y]_n$ sich dieselbe Zahl verschaffen können, ohne dass die beiden geheimen Zahlen x und y ausgetauscht werden müssen. Die Zahl S aber kann als gemeinsamer Schlüssel verwendet werden.

Ein kleines Beispiel: Alice und Bob wählen $b = 3$ und $n = 17$. Alice geheime Zahl ist $x = 4$; sie überträgt daher $u = [3^4]_{17} = [81]_{17} = 13$ an Bob. Dieser besitzt die Geheimzahl $y = 10$ und sendet folglich $v = [3^{10}]_{17} = [59.049]_{17} = 8$ an Alice. Außerdem berechnet Bob den Schlüssel $S = [u^y]_{17} = [13^{10}]_{17} = 16$. Die Rechnung von Alice lautet $S = [v^x]_{17} = [8^4]_{17} = 16$ und führt damit auf den gleichen Schlüssel.

Aufgabe: Wählen Sie (kleine) Zahlen und führen Sie das Verfahren durch.

3.3 Sicherheit des Verfahrens

Der Angreifer Mr. X kennt die Zahlen b, n, $u = [b^x]_n$ sowie $v = [b^y]_n$, nicht aber die geheimen Zahlen x und y, da diese nicht übertragen wurden. Zur Bestimmung von $S = [v^x]_n$ muss aber x bekannt sein. Die einzige Möglichkeit für Mr. X besteht in der Lösung der Gleichung $b^x \equiv u \bmod n$ (oder alternativ $b^y \equiv v \bmod n$). Die gesamte Sicherheit des Verfahrens hängt daher am Unvermögen des Herrn X, diese Gleichung tatsächlich zu lösen.

Wir nehmen an dieser Stelle an, dass Herr X die Gleichung $b^x \equiv u \bmod n$ nur durch Probieren lösen kann. Als Zahlen b und n wählen wir natürlich keine zweistelligen Zahlen, sondern 300-stellige; z.B.[6]

$$n = 467598293376684547350947367647078834228133877919179249590039375120953930062836$$
$$344301131374608653800586266491307481365622064384244384413190575456567207535839113 5$$
$$537108795991638155474452610874309742867231360502542308382199053675592825240788613 9$$
$$918985672771168817937493408077283357953943012616050620835 61 \approx 4.7 \cdot 10^{300}.$$

Gestehen wir Herrn X außerdem noch eine gehörige Portion Glück zu und vereinbaren, dass er für das Lösen einer der Gleichungen weniger als ein Promille der möglichen Zahlen, nämlich nur 10^{297} Möglichkeiten durchprobieren muss, und dies auf einer Milliarde Rechner parallel durchführen kann. In diesem Fall hat jeder Rechner noch 10^{288} Rechnungen durchzuführen, wozu er – wir sind weiterhin großzügig – lediglich 10^{275} Sekunden benötigt. Das aber sind rund $3 \cdot 10^{267}$ Jahre – und seit dem Urknall sind gerade einmal $1{,}4 \cdot 10^{10}$ (alias 14 Milliarden) Jahre vergangen.

Auch ein Abspeichern der kompletten Exponentialfunktion auf Festplatten ist unmöglich: Die etwa 2^{998} Werte würden selbst dann etwa $2^{998-40} = 2^{958}$ Terrabyte belegen, wenn wir jede der Zahlen in ein einzelnes Byte stopfen könnten. Rechnet man nun $2^{958} \approx 10^{288}$ Festplatten zu je einem Gramm (wir sind immer noch sehr großzügig), so erhält man mit 10^{285} kg einen Rechner, der unvorstellbar viel mehr als das Universum (Schätzung 10^{54} kg) wiegt.

Von einem „Brute-Force-Angriff“ (so nennt man die Methode, den Schlüssel durch Erraten zu finden) hat das DIFFIE-HELLMANN-Verfahren daher nur wenig zu fürchten. Nach wie vor weiß aber niemand, ob es nicht doch einen schnellen Algorithmus zur Lösung der Gleichung $b^x \equiv u \bmod n$ gibt. Ein solcher Algorithmus wäre natürlich das sofortige Ende des hier vorgestellten Verfahrens.

[6] Die Fragestellung, wie man sich solch großen Primzahlen zum Beispiel mit dem sogenannten Rabin-Miller-Verfahren beschafft und wie man eine geeignete Basis b findet, wird hier nicht behandelt.

3.4 Schnelles Potenzieren

Nachdem wir die Sicherheit des DIFFIE-HELLMANN-Verfahrens festgestellt haben, wollen wir als nächstes zur Implementierung übergehen, da wohl keiner die Zeit und die Lust hat, $[b^x]_n$ für 300-stellige Zahlen per Hand auszurechnen. Doch halt: Wie soll das eigentlich ein Rechner machen? Die naive Methode, zunächst b^x auszurechnen, produziert eine Zahl mit $3 \cdot 10^{302}$ Stellen, was wir sofort als undurchführbar verwerfen. Aber wie sollen überhaupt 10^{300} Multiplikationen ausgeführt werden?

Glücklicherweise führt ein wenig Nachdenken auf eine Lösung, die wir anhand eines Beispiels demonstrieren wollen: Gesucht wird $[2]_{2003}^{505}$. Wir entwickeln den Exponenten des Terms in die Binärdarstellung

$$505 = 2^8 + 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 1$$

und schließen hieraus

$$[2]_{2003}^{505} = [2]_{2003}^{2^8} \cdot [2]_{2003}^{2^7} \cdot [2]_{2003}^{2^6} \cdot [2]_{2003}^{2^5} \cdot [2]_{2003}^{2^4} \cdot [2]_{2003}^{2^3} \cdot [2]_{2003}^{1}.$$

Unter Beachtung der Gleichung $[2]_{2003}^{2^{n+1}} = \left([2]_{2003}^{2^n}\right)^2$ erhalten wir die Faktoren der rechten Seite der letzten Gleichung durch fortgesetztes Quadrieren:

- $[2]_{2003}^{2^0} = [2]_{2003}$
- $[2]_{2003}^{2^1} = \left([2]_{2003}^{2^0}\right)^2 = [2]_{2003}^2 = [4]_{2003}$
- $[2]_{2003}^{2^2} = \left([2]_{2003}^{2^1}\right)^2 = [4]_{2003}^2 = [16]_{2003}$
- $[2]_{2003}^{2^3} = \left([2]_{2003}^{2^2}\right)^2 = [16]_{2003}^2 = [256]_{2003}$
- $[2]_{2003}^{2^4} = \left([2]_{2003}^{2^3}\right)^2 = [256]_{2003}^2 = [65.536]_{2003} = [1.440]_{2003}$
- $[2]_{2003}^{2^5} = \left([2]_{2003}^{2^4}\right)^2 = [1440]_{2003}^2 = [2.073.600]_{2003} = [495]_{2003}$
- $[2]_{2003}^{2^6} = \left([2]_{2003}^{2^5}\right)^2 = [495]_{2003}^2 = [245.025]_{2003} = [659]_{2003}$
- $[2]_{2003}^{2^7} = \left([2]_{2003}^{2^6}\right)^2 = [659]_{2003}^2 = [434.281]_{2003} = [1.633]_{2003}$
- $[2]_{2003}^{2^8} = \left([2]_{2003}^{2^7}\right)^2 = [1.633]_{2003}^2 = [2.666.689]_{2003} = [696]_{2003}$

Das Ergebnis erhalten wir nunmehr aus der relativ kurzen Rechnung:

$$[2]_{2003}^{505} = [696]_{2003} \cdot [1.633]_{2003} \cdot [659]_{2003} \cdot [495]_{2003} \cdot [1.440]_{2003} \cdot [256]_{2003} \cdot [2]_{2003} = [780]_{2003}.$$

Das bekannte Verfahren zur Bestimmung der Binärdarstellung einer Zahl durch fortgesetzte Division durch 2 führt jetzt zu einem extrem schnellen Algorithmus für das Berechnen beliebiger Potenzen:

```
01    % Algorithmus zum schnellen Potenzieren
02    % mod(x,n) berechnet den Rest von x bei Division durch n
03    Input: ganze Zahl b; natuerliche Zahlen x, n; n>1
04    Lokal Variable: a, bloc, xloc, z
05    % Ausgabe: a = [b^x]_n
06
07    bloc := mod(b,n)
08    xloc := x
09    a:=1
10    while xloc<>0 do
11          z:=mod(xloc,2)
12          a:=mod(a*bloc^z,n)
13          xloc:=(xloc-z)/2
14          bloc:=mod(bloc^2,n)
15    end_while
16    Output: a
```

Es sei nun $x = z_h z_{h-1} \dots z_2 z_1 z_0$ die Binärdarstellung mit[7] $z_h = 0$ und $z_{h-1} = 1$. Per vollständiger Induktion weist man nun nach, dass die Variablen nach k-maligem Durchlaufen der while-Schleife wie folgt belegt sind:

$$b_{\text{loc}} = \left[b^{2^k}\right]_n; \quad x_{\text{loc}} = z_h z_{h-1} \dots z_k; \quad a = \left[b^{z_0 2^0} \cdot b^{z_1 2^1} \cdot b^{z_2 2^2} \cdot \dots \cdot b^{z_{k-1} 2^{k-1}}\right]_n.$$

Für $k = 0$ steht das bereits weiter oben. Für den Induktionsschritt $k \to k+1$ erhalten wir $z = z_k$ in Zeile 11, da nach Induktionsvoraussetzung x_{loc} die Binärdarstellung $x_{\text{loc}} = z_h z_{h-1} \dots z_k$ hat und folglich den Rest z_k bei Division durch 2 besitzt. Damit wird a in Zeile 12 der Wert

$$a \cdot b_{\text{loc}}^z = \left[b^{z_0 2^0} \cdot b^{z_1 2^1} \cdot b^{z_2 2^2} \cdot \dots \cdot b^{z_{k-1} 2^{k-1}} \cdot \left(b^{2^k}\right)^{z_k}\right]_n$$

zugewiesen – wie behauptet. In Zeile 13 führt die Division von $x_{\text{loc}} - z = (z_h \dots z_{k+1} z_k) - z_k$ durch zwei offensichtlich zur Binärdarstellung $z_h \dots z_{k+1}$ des neuen Wertes für x_{loc}. Schließlich wird aus $b_{\text{loc}} = \left[b^{2^k}\right]_n$ durch Quadrieren in Zeile 14 zu $b_{\text{loc}} = \left[b^{2^{k+1}}\right]_n$. Damit ist alles gezeigt.

Für $k < h$ ist der Wert von x_{loc} mit $z_h z_{h-1} \dots z_k = 01 z_{h-2} \dots z_k$ von 0 verschieden; die Schleife wird daher nochmals durchlaufen. Nach $k = h$ Durchläufen der while-Schleife erhalten wir mit $x_{\text{loc}} = z_h = 0$ die Abbruchbedingung. Wie gewünscht wird jetzt in Zeile 16 der Wert

$$a = \left[b^{z_0 2^0} \cdot b^{z_1 2^1} \cdot b^{z_2 2^2} \cdot \dots \cdot b^{z_{h-1} 2^{h-1}}\right]_n = \left[b^{z_0 2^0 + z_1 2^1 + z_2 2^2 + \dots + z_{h-1} 2^{h-1}}\right]_n = [b^x]_n$$

ausgegeben; man beachte hierzu nochmals $z_h = 0$.

Für das bereits oben durchgerechnete Beispiel ($b = 2, x = 505, n = 2003$) erhalten wir den folgenden Ablauf des Algorithmus; notiert sind die Werte der angegebenen Variablen nach dem k-ten Durchlaufen der while-Schleife:

Anzahl Schleifendurchläufe k	0	1	2	3	4	5	6	7	8	9
$a \coloneqq \left[a \cdot b_{\text{loc}}^{\text{mod}(x_{\text{loc}},2)}\right]_{2003}$	1	2	2	2	512	176	991	91	381	780
$b_{\text{loc}} \coloneqq [b_{\text{loc}}]_{2003}^2$	2	4	16	256	1440	495	659	1633	696	1693
$x_{\text{loc}} \coloneqq (x_{\text{loc}} - \text{mod}(x_{\text{loc}}, 2))/2$	505	252	126	63	31	15	7	3	1	0

Wieder erhalten wir das gewünschte Ergebnis $[2^{505}]_{2003} = 780$. Allerdings erkennen wir auch, dass das letzte Quadrieren von b_{loc} überflüssig war – ein Schönheitsfehler des Programms, der leicht behoben werden kann.

Aufgabe 11: Berechnen Sie 2^{50} ohne Verwendung eines Taschenrechners.

Aufgabe 12: Berechnen Sie $[7^{50}]_{103}$ ohne die Verwendung eines Taschenrechners.

Aufgabe 13: Implementieren Sie den Algorithmus zum schnellen Potenzieren.

4. Lösung der Aufgaben

Aufgabe 1: Durchführen des Verfahrens liefert den Geheimtext `bikhxpb ezd lwmzbo`.

Aufgabe 2: Der Klartext lautet `geheime nachricht`.

[7] Diese etwas seltsame Festlegung dient zur besseren Beschreibung des Algorithmus. Für $h = 0$ ist dabei auch $x = 0$ in die Betrachtung eingeschlossen.

Aufgabe 3: Das Schlüsselwort lautet `piraten` und der Klartext ist:

```
wir treffen uns vor der insel java.
wir fahren anschliessend die insel borneo an.
dort nehmen wir frischwasser auf.
im anschluss ueberfallen wir die drei inseln noerdlich von
borneo.
die erbeuteten schaetze bringen wir moeglichst rasch zu
unserem ueblichen versteck im bermudadreieck.
ich moechte noch darauf hinweisen dass im bereich der insel
java einige schiffe des gouverneurs von indonesien gesichtet
worden sind.
also aufpassen.
mfg
```

Aufgabe 4: Die Teile a) und b) lassen sich wörtlich wie Teil c) beweisen. Als Alternative kann man auch wie folgt argumentieren: Nach Voraussetzung werden die Differenzen $a-u$ und $b-v$ durch n geteilt. Daher ist n auch ein Teiler von $(a-u)\pm(b-v)=(a\pm b)-(u\pm v)$, was aber schon die Behauptung $a\pm b\equiv u\pm v \bmod m$ zeigt. Für Teil c) erhält man aus der Gleichung $a^m-u^m=(a-u)\cdot(a^{m-1}+a^{m-2}u+\cdots+au^{m-2}+u^{m-1})$, dass mit $a-u$ auch a^m-u^m durch n teilbar ist.

Aufgabe 5:

a. $[2]_5+[4]_5=[6]_5=[1]_5$ $\quad$ $[6]_{17}\cdot[3]_{17}=[18]_{17}=[1]_{17}$ $\quad$ $[3]_7+[4]_7=[7]_7=[0]_7$
$[2]_6\cdot[3]_6=[6]_6=[0]_6$ $\quad$ $[3]_7^4=[81]_7=[4]_7$

b. $[729+865]_5=[729]_5+[865]_5=[4]_5+[0]_5=[4]_5$
$[667\cdot 528]_5=[667]_5\cdot[528]_5=[2]_5\cdot[3]_5=[6]_5=\{1\}_5$
$[17^{10}]_5=[17]_5^{10}=[2]_5^{10}=[1024]_5=[4]_5$
$[14^{1.000}]_5=[4]_5^{1.000}=[-1]_5^{1.000}=[1]_5$

Aufgabe 6: Die direkte Übersetzung: Aus $[a]_n=[b]_n$ folgt $[a^x]_n=[b^x]_n$; dies steht in Satz 2.d.

Aufgabe 7: Es ist $\frac{1}{1}=\frac{2}{2}$ und $\frac{3}{5}=\frac{12}{20}$, aber $\frac{1}{1}\oplus\frac{3}{5}=\frac{4}{6}=\frac{2}{3}$ ist von $\frac{2}{2}\oplus\frac{12}{20}=\frac{14}{22}=\frac{7}{11}$ verschieden – die „Schüleraddition $\oplus$“ ist daher nicht wohldefiniert.

Aufgabe 8: Nicht teilerfremd zu $p\cdot q$ sind die Zahlen $p, 2p, \dots, (q-1)p$ sowie $q, 2q, \dots, (p-1)q, pq$. Diese $p+q-1$ Zahlen sind paarweise verschieden, da p und q verschiedene Primzahlen sind. Damit bleiben $pq-p-q+1=(p-1)\cdot(q-1)$ zu $p\cdot q$ teilerfremde Zahlen unterhalb $p\cdot q$ übrig. Wir erhalten daher

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

Aufgabe 9: Ist die Ordnung x_b von $[b]_{25}$ echt kleiner $\varphi(25)=20=2^2\cdot 5$, so ist x_b ein echter Teiler von 25 (Satz von EULER) und damit ein Teiler von $4=20/5$ oder von $10=20/2$; es gilt daher $b^4\equiv 1 \bmod 25$ oder $b^{10}\equiv 1 \bmod 25$. Wir testen für $b=2$:

$$2^4\equiv 16 \bmod 25 \text{ und } 2^{10}\equiv 24 \bmod 25.$$

Folglich besitzt $[b]_{25}$ die Ordnung $\varphi(25)=20$ – wir haben daher eine primitives Element modulo 25 gefunden.

Für $n=15$ beachte: Sind b und 15 teilerfremd, so auch b und 3 sowie b und 5. Wegen $\varphi(3)=2$ und $\varphi(5)=4$ erhalten wir aus dem Satz von EULER $b^4\equiv 1 \bmod 3$ und $b^4\equiv 1 \bmod 5$, d.h. b^4-1 wird sowohl durch 3 als auch durch 5 geteilt. Dies zeigt die Teilbarkeit von b^4-1 durch 15 und damit $b^4\equiv 1 \bmod 15$. Die Ordnung von $[b]_{15}$ ist daher höchstens 4 und somit verschieden von $\varphi(15)=8$. Also gibt es kein primitives Element modulo 15.

Für ungerades b ist jede der Zahlen $b\pm 1$ und b^2+1 gerade; eine der Zahlen $b\pm 1$ ist sogar durch 4 teilbar. Damit ist $b^4-1=(b-1)\cdot(b+1)\cdot(b^2+1)$ durch 16 teilbar, d.h. $b^4\equiv 1 \bmod 16$. Also ist die Ordnung von b höchstens 4 und damit echt kleiner als $\varphi(16)=8$, so dass b kein primitives Element modulo 16 sein kann.

Aufgabe 10: Beachte $\varphi(5) = 4$ und wende den Satz von EULER an. Dann erhalte $[17^{2.015}]_5 = [2]_5^{4\cdot 503+3} = ([2]_5^4)^{503} \cdot [2]_5^3 = [1]_5^{503} \cdot [8]_5 = [3]_5$.

Aufgabe 11: Wir schreiben $2^{50} = 2^{40} \cdot 2^{10}$. Wir kennen $2^{10} = 1.024$ und berechnen durch Quadrieren $2^{20} = 1.024^2 = (1.000 + 24)^2 = \cdots = 1.048.576$ sowie $2^{40} = 1.048.576^2 = (1.050.000 - 1.424)^2 = \cdots = 1.099.511.627.776$. Damit erhalten wir $2^{50} = 1.024 \cdot 1.099.511.627.776$ und endlich

$$2^{50} = 1.125.899.906.842.624.$$

Aufgabe 12: Wir sehen $7^2 = 49$ und $7^4 = (50-1)^2 = 2.501 - 100 \equiv 29 + 3 = 32 \bmod 103$. Wegen $7 \cdot 32 = 224 \equiv 18 \bmod 103$ folgt jetzt $[7]_{103}^5 = [18]_{103}$ und durch Quadrieren $[7]_{103}^{10} = [324]_{103} = [15]_{103}$. Durch nochmaliges Quadrieren bestimmen wir $[7]_{103}^{20} = [225]_{103} = [19]_{103}$. Es ist $[7]_{103}^{25} = [7]_{103}^{20} \cdot [7]_{103}^5 = [18 \cdot 19]_{103} = [342]_{103} = [33]_{103}$. Wegen $33^2 = 9 \cdot 121 \equiv 9 \cdot 18 = 162 \equiv 59 \bmod 103$ haben wir

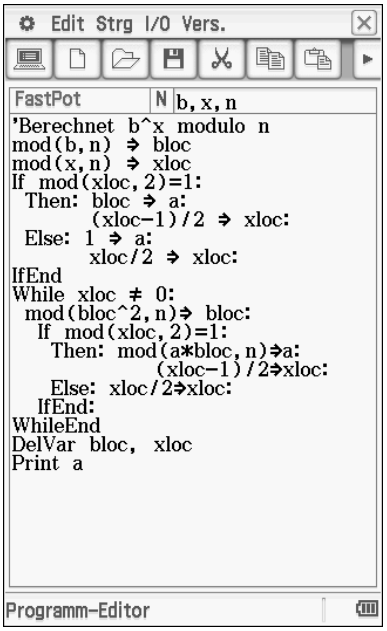
$$[7^{50}]_{103} = [59]_{103}.$$

Aufgabe 13: Auf der rechten Seite befindet sich eine Implementierung des Algorithmus für den CAS-Taschenrechner ClassPad II. Die if-Klausel vor der while-Schleife verhindert, dass b_{loc} die einmal zuviel quadriert wird. Auf dem ClassPad II bedeutet „$a \Rightarrow b$“ die Zuweisung des Wertes a an die Variable b.

Tests des Programms per ClassPad-Manager auf einem PC zeigen, dass die Algorithmus auch für Rechnungen der Art

$$3219898354^{6667466446} \equiv 623019857126874593 \bmod 6546546598413527$$

keine wahrnehmbare Rechenzeit benötigt.



5. Literatur

Einige weiterführende Werke zur Zahlentheorie und Kryptographie sind hier zusammengetragen:

J. Buchmann	Einführung in die Kryptographie, Springer 2010
A. Beutelspacher	Kryptologie, 10. Aufl., Springer 2015
M. Welschenbach	Kryptographie in C und C++, 2. Aufl., Springer Xpert.press 2001
C. Karpfinger und H. Kiechle	Kryptologie, Vieweg+Teubner 2010
B. Schneier	Angewandte Kryptographie, Addison-Wesley 1996

Anschrift des Autors:

Professor Dr. Harald Löwe
Technische Universität Braunschweig
Institut Computational Mathematics
Pockelsstraße 14, 38106 Braunschweig
h.loewe@tu-bs.de