Teil 2 Tauschchiffre, modulare Arithmetik

Wir wiederholen unsere Buchstaben-Zahl Codierung:

Buchstabe

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Kodierung

Bei der Verschiebechiffre (Cäsar-Chiffrierung) mit Schlüssel s wird zu jedem Klartextbuchstaben (genauer: zum Zahlencode des Buchstabens) die Zahl s addiert.
Falls man dabei über 25 hinauskommt, so rechnet man wieder bei 0 weiter: 26 entspricht 0, 27 entspricht 1 und so weiter.
Mathematisch gesagt: Man reduziert *modulo 26* .

Statt jeden Buchstaben um s Stellen (additiv) zu verschieben, könnte man ihn auch multiplikativ um einen Faktor t verändern.

Probieren wir es mit t=2:

Klarbuchstabe	:	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
Geheimbuchstabe	:	A	C	E	G	I	K	M	O	Q	S	U	W	Y	A	C	E	G	I	K	M	O	Q	S	U	W	Y
		?													?												

Man sieht, daß es so nicht geht: Unterschiedliche Klarbuchstaben (beispielsweise a und n) transformieren sich auf denselben Geheimbuchstaben.
Dies widerspricht der selbstverständlichen Forderung, daß eine in beiden Richtungen eindeutige Beziehung zwischen Menge der Klarbuchstaben und Menge der Geheimbuchstaben bestehen muß.

Wir versuchen es erneut mit Faktor t=3:

Klarbuchstabe	:	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
Geheimbuchstabe	:	A	D	G	J	M	P	S	V	Y	B	E	H	K	N	Q	T	W	Z	C	F	I	L	O	R	U	X

Diesmal war der Versuch erfolgreich: Jeder Klarbuchstaben entspricht einem und auch nur einem Geheimbuchstaben.

Durch Probieren stellt man fest: Legitime **multiplikative Schlüssel** sind die Zahlen:

1 , 3 , 5 , 7 , 9 , 11 , 15 , 17 , 19 , 21 , 23 , 25

Mögliche multiplikative Schlüssel gibt es also (nur) 12.

Warum es gerade diese Zahlen sind und wie man sie einfacher bestimmen kann als durch Ausprobieren, werden wir später klären.

Tauschchiffre
Die Tauschchiffre entsteht als Kombination von Verschiebechiffre mit anschließender multiplikativer Chiffrierung. Der Schlüssel einer solchen Tauschchiffre ist ein Paar (s,t), das folgendermaßen benutzt wird: Zunächst wendet Alice auf den Klartext die Verschiebechiffre mit Schlüssel s an. Anschließend wendet sie auf das Zwischenresultat die multiplikative Chiffre mit Schlüssel t an (wobei t natürlich eine der oben angegebenen 12 Zahlen sein muß).

Man beachte: Die Anzahl möglicher Tauschchiffren ist 26*12 = 312. Es wird nicht mehr so leicht sein, diese Chiffre *von Hand* zu brechen. Mit Rechnereinsatz und Alphabeten mit nur 26 Buchstaben ist es natürlich immer noch kein Problem.

Der Adressat Bob muß beim Entschlüsseln zunächst die Multiplikation mit t und dann die Addition mit s zurückrechnen. Das letztere, die Rückrechnung der Verschiebung s ist simpel: Sie wird neutralisiert durch nochmalige Verschiebung mit -s. Um die Multiplikation mit t zurückzurechnen, ist es das naheliegendste, die obige Multiplikationstabelle umgekehrt, also von unten nach oben zu benutzen: Man sucht unten den Geheimbuchstaben und findet oben den zugehörigen Klartextbuchstaben. Am besten würde man dazu die Multiplikationstabelle zunächst alphabetisch nach den Geheimbuchstaben sortieren.

Aber zu dieser Methode gibt es eine elegante und effiziente Alternative. Sei beispielsweise s=7 und t=3: Die Multiplikation mit t=3 neutralisiert Bob durch nochmalige Multiplikation, und zwar mit 9. - Letzteres ist leicht einzusehen:

$$27 \equiv 1 \bmod 26$$

also

$$(x*3)*9 = x*27 \equiv x*1 = x \bmod 26$$

Für jedes beliebige x liefert Multiplikation mit 3 und nochmalige Multiplikation mit 9 das gegebene x zurück!

Beispiel (für die Tauschchiffre)
Additiver Schlüssel s = 7
Multiplikativer Schlüssel t = 3

	Klartext	:	s	c	h	r	i	f	t
Alice ver-									
schlüsselt:	Zahlenwert	:	18	2	7	17	8	5	19
	+ s (d.h. + 7)	:	25	9	14	24	15	12	0
	* t (d.h. * 3)	:	23	1	16	20	19	10	0
	Geheimtext	:	X	B	Q	U	T	K	A
	schickt Alice an Bob								

Bob ent-									
schlüsselt:	Zahlenwert	:	23	1	16	20	19	10	0	|*9
	* inv(t) (d.h. *9)	:	25	9	14	24	15	12	0	|-7
	- s (d.h. - 7)	:	18	2	7	17	8	5	19	
	Klartext	:	s	c	h	r	i	f	t	

Die Tatsache, daß zwei modulare Multiplikationen sich gegenseitig aufheben können, ist von so großer Bedeutung, daß wir ihr einen Namen geben. Die obige Rechnung

$$3*9\ (= 27) \equiv 1 \bmod 26$$

ist ein Muster für die sogenannte *modulare Inverse*:

Modulare Inverse: $t * inv(t) \equiv 1 \bmod m$ (oben: m = 26)

Multiplikative Inverse gibt es nicht nur bei den reellen Zahlen (3*1/3 = 1), sondern überall dort, wo die für Inverse charakteristische Gleichung

$$x*x^{-1} = 1$$

gültig ist.
In der modularen Arithmetik tritt modulare Kongruenz an die Stelle der Gleichheit. Die modulare Inverse von 3 mod 26 ist 9. Die modulare Inverse einer ganzen Zahl ist stets ebenfalls eine ganze Zahl, die kleiner als der Modul ist.

Ein weiterer Punkt ist wichtig: Bei den reellen Zahlen gibt es genau eine Zahl, die keine Inverse hat, nämlich die 0 (*durch 0 kann man nicht teilen*). In modularer Arithmetik gibt es normalerweise noch andere Zahlen, die keine Inverse besitzen:

Beispiel: Die 2 hat keine Inverse mod 26 (warum nicht?)

Wir halten fest: Die 3 war legitimer multiplikativer Schlüssel für die Tauschchiffre und hatte eine modulare Inverse (die 9). Die Zahl 2 war *kein* legitimer multiplikativer Schlüssel und hatte *keine* modulare Inverse.

Die letztere Koinzidenz war kein Zufall. Später werden wir zeigen: Genau die legitimen multiplikativen Schlüssel der Tauschchiffre haben eine modulare Inverse.

Die folgende Tabelle gibt die modulo 26 berechneten Inversen der legitimen multiplikativen Schlüssel:

t	: 1	3	5	7	9	11	15	17	19	21	23	25
inv(t)	: 1	9	21	15	3	19	7	23	11	5	17	25

Mathematische Grundlagen der Kongruenzrechnung

Bisher war unsere modulare Rechnerei rein intuitiv. Es ist an der Zeit, die Dinge zu systematisieren. Für das Folgende sei eine natürliche Zahl m, der **Modul**, zu Grunde gelegt.

Definition (modulare Kongruenz)
Zwei ganze Zahlen a und b heißen **kongruent** modulo m , wenn die Differenz a-b ohne Rest durch m teilbar ist:

$$a \equiv b \bmod m$$

Oder anders gesagt: $a-b = \alpha * m$ für passendes ganzzahliges α

Durch die Kongruenzrelation wird eine Zerlegung der Ganzen Zahlen $\mathbb{Z}$ in das System der m **Restklassen** $\mathbb{Z}_m$ induziert.

Beispiel m = 26
Zu $\mathbb{Z}_{26}$ gehören 26 Restklassen.
Die zu 0 gehörende Restklasse enthält die Zahlen
{0,26,-26,52,-52,..}
Die zu 1 gehörende Restklasse enthält die Zahlen
{1,27,-25,53,-51,..}
und so weiter.

Ein Element einer Restklasse ist ein **Vertreter** dieser Restklasse. Vertreter der zu 0 gehörenden Restklasse im Beispiel sind 0 oder 26 oder -52 etc.
Jeder Vertreter einer Restklasse ist *genauso gut* wie ein anderer Vertreter derselben Restklasse. Es ist allerdings praktisch, kleine Vertreter oder Vertreter im Bereich 0,1,...,m-1 zu wählen.

Mit Restklassen kann man rechnen. Will man Restklassen addieren, so wähle man sich Vertreter aus ihnen, addiere letztere zur Summe S, suche diejenige Restklasse, die S enthält und deklariere diese Restklasse als das Ergebnis.
Ganz entsprechend macht man es bei der Multiplikation.

Beispiel m = 26

$$3 + 24 = 27 \equiv 1 \bmod 26$$
$$3 + 50 = 53 \equiv 1 \bmod 26$$

$$4 * 8 = 32 \equiv 6 \bmod 26$$
$$4 * (-18) = -72 \equiv 6 \bmod 26$$
$$\text{oder } -22 * (-18) = 396 \equiv 6 \bmod 26$$

Diese Beispiele deuten auf ein Problem. Da jeder Vertreter einer Restklasse genauso gut ist wie ein anderer, ist die Frage, ob das Resultat beim Rechnen mit Restklassen etwa von der Wahl der Vertreter abhängt. - In diesem Falle wäre das Rechnen mit Restklassen so gut wie wertlos!
In den Beispielen war das Ergebnis immer das gleiche. In der Tat ist das immer so, es gilt:

Satz Das Ergebnis bei Addition und Multiplikation von Restklassen ist unabhängig von der Wahl der Vertreter.

Beweis:
Addition:
Seien a und a' unterschiedliche Vertreter der einen Restklasse, also $a-a' = \alpha*m$
Seien b und b' unterschiedliche Vertreter der anderen Restklasse, also $b-b' = \beta*m$
Dann ist zu zeigen: $a+b \equiv a'+b' \bmod m$
oder $(a+b)-(a'+b') = const*m$
In der Tat ist

$$(a+b)-(a'+b') = (a-a')+(b-b') = \alpha*m+\beta*m = (\alpha+\beta)m$$

also Vielfaches von m.

Multiplikation:
Seien a und a' Vertreter der einen Restklasse und b und b' Vertreter der anderen Restklasse.
Dann ist zu zeigen: $a*b \equiv a'*b' \bmod m$
Es ist

$$\begin{aligned} a*b - a'*b' &= a*b - a'*b + a'*b - a'*b' = \\ &= (a-a')*b + a'*(b-b') \\ &= \alpha*m*b + a'*\beta*m \\ &= (\alpha*b + a'*\beta)*m \text{ , also Vielfaches von m //} \end{aligned}$$

Man beachte: Später werden wir modulare Zahlen häufig zu *potenzieren* haben. Man kann sich fragen, ob man ähnlich wie bei Summand und Faktor auch beim Exponenten manipulieren darf. Den Exponenten einfach modular reduzieren geht nicht, sei z.B. m=10.
Dann ist $2^{12} = 2^{10}*2^2 = 1024*4 \equiv 4*4 = 16 \equiv 6 \bmod 10$
aber $2^2 = 4 \not\equiv 6 \bmod 10$

Auf modulares Potenzieren werden wir im Rahmen der Public-Key-Kryptografie noch genauer zurückkommen.

Die oben offen gebliebene Frage war, welche Zahlen t sich als multiplikative Schlüssel für die Tauschchiffre eignen. Beim Modul m=26 waren es die Zahlen

1 , 3 , 5 , 7 , 9 , 11 , 15 , 17 , 19 , 21 , 23 , 25

Das sind nur ungerade Zahlen, aber nicht alle: die 13 fehlt. Wir wollen zeigen: Die tauglichen Zahlen sind genau diejenigen, die zum Modul teilerfremd sind.

Zwei Zahlen sind teilerfremd, wenn ihr **größter gemeinsamer Teiler** 1 ist (ggt=1). Wir brauchen ein effizientes Verfahren zur Bestimmung des größten gemeinsamen Teilers zweier Zahlen. Ein solches Verfahren ist seit langem bekannt, der *euklidische Algorithmus*.

Euklidischer Algorithmus
Seien gegeben zwei natürliche Zahlen a und b mit a≥b.
Man setzt zunächst $r_0=a$ und $r_1=b$ und führt dann fortwährende Division mit Rest durch, solange bis der Rest 0 ist:

$$\begin{aligned} r_0 &= q_1 * r_1 + r_2 \qquad \text{mit} \quad 0 < r_2 < r_1 \\ r_1 &= q_2 * r_2 + r_3 \qquad \text{mit} \quad 0 < r_3 < r_2 \\ r_2 &= q_3 * r_3 + r_4 \qquad \text{mit} \quad 0 < r_4 < r_3 \\ &\ldots \end{aligned}$$

und so weiter, bis die Division ohne Rest aufgeht.

Es gilt: Der letzte von 0 verschiedene Rest ist der **größte gemeinsame Teiler** ggT(a,b) von a und b.

Beispiel Sei a=792 und b=75

$$\begin{aligned} 792 &= 10*75 + 42 \\ 75 &= 1*42 + 33 \\ 42 &= 1*33 + 9 \\ 33 &= 3*9 + 6 \\ 9 &= 1*6 + 3 \\ 6 &= 2*3 \quad \text{(ohne Rest)} \end{aligned}$$

Also der letzte Rest (=3) ist der größte gemeinsame Teiler der beiden Zahlen:

$$ggT(792,75) = 3$$

Der euklidische Algorithmus kann folgendermaßen berechnet werden (Pseudocode):

```
read a,b
while b≠0
   r := a mod b
   a := b
   b := r
Return a
```

Wir brauchen noch eine weitere Tatsachen aus der Zahlentheorie:

Satz Ist d der größte gemeinsame Teiler der Zahlen a und b, so gibt es ganze Zahlen x und y mit

$$x*a + y*b = d = ggT(a,b)$$

Anders gesagt: Der größte gemeinsame Teiler von a und b läßt sich ganzzahlig aus a und b kombinieren.

Diese Tatsache leuchtet ein, wenn man die Rechnung des obigen Beispiels von *unten* nach *oben* durchgeht:

```
3 =      9 -  1*6      =       9 - 1*( 33 -  3* 9)
3 = -1* 33 +  4*9      =   -1*33 + 4*( 42 -  1*33)
3 =  4* 42 -  5*33     =    4*42 - 5*( 75 -  1*42)
3 = -5* 75 +  9*42     =   -5*75 + 9*(792 - 10*75)
3 =  9*792 - 95*75
```

Also

$$3 = 9*792 - 95*75$$

Auf der Basis dieses grundlegenden euklidischen Algorithmus können nun die offenen Fragen hinsichtlich Invertierbarkeit modulo m und Legitimität des multiplikativen Tauschschlüssels vollständig geklärt werden:

Satz Sei gegeben ein Modul m. Dann sind die folgenden drei Aussagen äquivalent:

i) Der größte gemeinsame Teiler von a und m ist 1 :
$$ggT(a,m) = 1$$

ii) a hat eine modulare Inverse inv_a : $a*inv\_a \equiv 1 \bmod m$

iii) Die **Kürzungsregel** der Kongruenzrechnung:
$$a*b \equiv a*c \bmod m \quad => \quad b \equiv c \bmod m$$

Aussage (iii) ist gleichbedeutend damit, daß die Zahl a legitimer multiplikativer Schlüssel modulo m ist: Wenn die Zahlen nach Multiplikation mit a übereinstimmen, dann haben sie auch vorher schon übereingestimmt (Übereinstimmung jeweils im Sinne der modularen Kongruenz).
Man beachte, daß die Kürzungsregel keineswegs immer gilt:

$13*4 \equiv 13*2 \bmod 26$ und dennoch $4 \not\equiv 2 \bmod 26$

Mit Blick auf die oben untersuchten Tauschchiffren bedeutet der Satz: Eine Zahl a ist genau dann legitimer multiplikativer Schlüssel modulo m, wenn ggT(a,m)=1.

Beweis des Satzes:
(i)=>(ii) Aus ggT(a,m)=1 folgt
$$x*a + y*m = 1$$
für passende ganze Zahlen x und y. Liest man diese Gleichung modulo m, so erhält man
$$x*a + 0 \equiv 1 \bmod m$$
oder
$$x*a \equiv 1 \bmod m$$

Also ist x die modulare Inverse von a.

(ii)=>(iii)
Aus $a*b = a*c \bmod m$
folgt $inv\_a*a*b = inv\_a*a*c \bmod m$
also $b \equiv c \bmod m$

(iii)=>(i)
Der Modul m läßt sich schreiben in der Form:

$m = ggT(a,m)*Rest$ (z.B. $10 = ggT(8,10)*Rest = 2*5$)

Ist nun (indirekter Beweis) $ggT(a,m)>1$, so folgt
$Rest \not\equiv 0 \bmod m$ (*)

Andererseits ist (m =) $ggT(a,m)*Rest$ ein Teiler von $a*Rest$
oder: $a*Rest$ ist Vielfaches von m
Also gilt $a * Rest \equiv 0 \bmod m$
oder $a * Rest \equiv a*0 \bmod m$
Kürzen durch a: $Rest \equiv 0 \bmod m$ im Widerspruch zu (*) //

Aus dem Beweis des Satzes geht auch hervor, wie man die modulare Inverse von t bekommt: Mit dem euklidischen Algorithmus bestimme man Zahlen x und y mit

$$x*t + y*m = 1$$

Die Zahl x ist dann die modulare Inverse von t.

Beispiel (von früher) m=26 und t=3
Vorwärtsrechnung:
$26 = 8*3 + 2$
$3 = 1*2 + 1$
$2 = 2*1$ (ohne Rest)

Also ggt(26,3)=1, die Zahlen 26 und 3 sind in der Tat teilerfremd.

Um die Inverse von 3 zu bekommen, rechnet man rückwärts:

$1 = 3 - 1*2$
$1 = 3 - 1*(26 - 8*3)$
$1 = -26 + 9*3$

Modulo 26 gelesen ist das $1 \equiv 9*3 \bmod 26$

Also ist 9 die modulare Inverse zu 3 : $inv(3) = 9 \bmod 26$

Die folgenden Tatsachen sollen für spätere Zwecke noch einmal betont werden:
- Der euklidische Algorithmus kann auf dem Rechner effizient implementiert werden (logarithmischer Rechenaufwand, siehe Ergänzungen).
- Der größte gemeinsame Teiler zweier ganzer Zahlen kann (mit

dem euklidischen Algorithmus) effizient ausgerechnet werden.
Insbesondere kann leicht entschieden werden, ob zwei Zahlen teilerfremd sind.
- Die modulare Inverse einer invertierbaren Zahl kann (ebenfalls unter Zuhilfenahme des euklidischen Algorithmus) effizient ausgerechnet werden.

Ergänzungen zur Kryptologie, Teil 2
==========

1) Man berechne log m, m^2, 2^m und m! für m=1,10,100 und 1000 und trage die Ergebnisse in eine Tabelle ein (für die großen m nur näherungsweise).

2) Sei m=792.
Ist 89 bezüglich m modular invertierbar?
Ist 123 bezüglich m modular invertierbar?
Gegebenenfalls bestimme man die Inverse.

3) Wie viele legitime multiplikative Schlüssel besitzt ein Alphabet mit 25 bzw. 27 Buchstaben?

4) Der Rechenaufwand beim Euklidischen Algorithmus ist
$$O(\log_2 m)$$
wenn m die größere der beiden beteiligten Zahlen ist.
Um dies einzusehen, mache man sich klar: Die Reste halbieren sich spätestens bei jedem zweiten Schritt.

5a) Vom Praktikum weiß man, daß die Tauschchiffre einem Known-Plaintext-Angriff kaum standhält. Man könnte nun versuchen, die Tauschchiffre dadurch sicherer zu machen, daß man sie doppelt anwendet: Man chiffriert zunächst mit einem Schlüsselpaar (s_1,t_1) und das Ergebnis noch einmal mit einem zweiten Schlüsselpaar (s_2,t_2).
Kommentieren Sie diesen Vorschlag, machen Sie sich die Verhältnisse an einem Beispiel klar.

(b) Bei der Tauschchiffre wird zunächst mit s addiert und dann mit t multipliziert. Man könnte es auch umgekehrt machen.
Kommentieren Sie auch diesen Vorschlag!