Einführung in die Programmierung für Studierende der Naturwissenschaften
SS 2021

https://aam.uni-freiburg.de/agdo/lehre/ss21/prog/index.html

UNI FREIBURG

Codebreaker für Caesar-Verschlüsselung mit Häufigkeitsanalyse

von Daniel Rath und Theresa Maurer

Verschlüsselung und Entschlüsselung

Die Caesar-Verschlüsselung ist eine einfache, monoalphabetische Verschlüsselung. Die Verschlüsselung der Buchstaben a-z entspricht einer Verschiebung des Alphabets:

Klartext	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
Chiffre:	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	C

Abbildung 1: Beispiel für eine Caesar-Verschlüsselung mit Verschiebung um 3 Buchstaben

Die Verschlüsselung lässt sich leichter darstellen, wenn jeder Buchstabe in seiner ASCII-Darstellung betrachtet wird. Jeder Buchstabe wird so als Zahl gestellt und die Verschiebung entspricht dann einer Addition.
Jedem Buchstaben des Alphabets ist eine Zahl im ASCII-Code zugeordnet:

Character	ASCII
a	97
b	98
c	99
d	100
e	101
f	102
g	103
h	104
i	105
j	106
k	107
l	108
m	109

Character	ASCII
n	110
o	111
p	112
q	113
r	114
s	115
t	116
u	117
v	118
w	119
x	120
y	121
z	122

Character	ASCII
A	65
B	66
C	67
D	68
E	69
F	70
G	71
H	72
I	73
J	74
K	75
L	76
M	77

Character	ASCII
N	78
O	79
P	80
Q	81
R	82
S	83
T	84
U	85
V	86
W	87
X	88
Y	89
Z	90

Character	ASCII
0	48
1	49
2	50
3	51
4	52
5	53
6	54
7	55
8	56
9	57

Abbildung 2: Groß-/Kleinbuchstaben und Zahlen mit dem entsprechenden ASCII-Code

Der Einfachheit halber beschränken wir uns hier auf die Kleinbuchstaben von a-z, also die ASCII-Zeichen 97-122. Ein Kleinbuchstabe x in ASCII-Darstellung kann nun durch eine additive Funktion

$$C(x) = ((x - 97) + c) \bmod 26) + 97$$

verschlüsselt werden, für ein $c \in \mathbb{Z}$. Das c entspricht der Verschiebung des Alphabets.
Der verschlüsselte Buchstabe ist dann wieder eine Zahl zwischen 97 und 122 und somit auch ein Kleinbuchstabe.
Die zugehörige Dechiffrierfunktion ist dann

$$D(x) = ((x - 97) - c) \bmod 26) + 97,$$

da

$$\begin{aligned} D(C(x)) =& ((((x - 97) + c) \bmod 26) + 97 - 97) - c) \bmod 26) + 97 = \\ & (((x - 97) + c) - c) \bmod 26) + 97 = \\ & (((x - 97) \bmod 26) + 97 = x, \text{ für } 97 \leq x \leq 122 \end{aligned}$$

Für ein c, das die Funktionen C(x) und D(x) eindeutig bestimmt, bewirkt jede Zahl c' mit der Eigenschaft $c = c' \bmod 26$ dieselbe Verschiebung des Alphabets. Daher gibt es für die Chiffrier-/Dechiffrierfunktion nur 26 verschiedene Möglichkeiten und Caesar-Verschlüssungen sind sehr einfach zu knacken.

Entschlüsselung eines Caesar-chiffrierten Textes durch Häufigkeitsanalyse

Um einen Caesar-verschlüsselten Text zu entschlüsseln, muss man die Dechiffrierfunktion D(x) finden und diese auf jeden Buchstaben anwenden. Dazu ist ausreichend, die Verschiebung c zu finden, da durch das c die Funktion D(x) eindeutig bestimmt wird.
In der deutschen Sprache kommt das e eindeutig am häufigsten vor. Durch eine Häufigkeitsanalyse der Buchstaben des verschlüsselten Textes, lässt sich so der Buchstabe y (in Ascii-Darstellung), auf den das e abgebildet wurde, ermitteln. Es gilt $D(y) = ((y - 97) - c) \bmod 26) + 97 = 101$ und somit $c = y - 101$.
Dabei ist zu Beachten, dass die Häufigkeitsanalyse nur bei ausreichend langen Texten funktioniert.

Buchstabe	Häufigkeit in %	Buchstabe	Häufigkeit in %
a	6,51	n	9,78
b	1,89	o	2,51
c	3,06	p	0,79
d	5,08	q	0,02
e	17,40	r	7,00
f	1,66	s	7,27
g	3,01	t	6,15
h	4,76	u	4,35
i	7,55	v	0,67
j	0,27	w	1,89
k	1,21	x	0,03
l	3,44	y	0,04
m	2,53	z	1,13

Abbildung 3: Häufigkeit der Buchstaben in der deutschen Sprache

C++-Programm zur Ent-/Verschlüsselung

Die Verschlüsselungsfunktion: encrypt_text()

Die Funktion fordert als erstes die Eingabe eines zu verschlüsselnden Textes, wobei der eingegebene Text mindestens 25 Zeichen enthalten muss, damit sich der Text auch wieder mit einer Häufigkeitsanalyse entschlüsseln lässt. Dann muss noch der Kleinbuchstabe eingegeben werden, auf den das a bei der Verschlüsselung abgebildet werden soll, um die Differenz c zu bestimmen.
Anschließend werden alle Großbuchstaben des Textes in Kleinbuchstaben umgewandelt. Dann wird über den ganzen Text iteriert und auf jeden Buchstaben die Verschlüsselungsfunktion C(x) angewandt, wobei Sonderzeichen nicht verschlüsselt werden, sondern einfach an der richtigen Stelle im Text wieder eingefügt werden.
Der verschlüsselte Text wird dann als string zurückgegeben.

Die Entschlüsselungsfunktion: decrypt_text()

Die Funktion nimmt einen Caesar-verschlüsselten Text als Eingabe, der mindestens die Länge 2 hat (und keine Großbuchstaben enthält). Dann wird die Häufigkeit jedes Buchstaben im Text gezählt und in eine Priority Queue eingefügt, wobei der key die absolute Häufigkeit als integer ist und der value der entsprechende Buchstabe.
Der am häufigsten verwendete Buchstabe im verschlüsselten Text entspricht dann dem value des obersten Elements in der Priority Queue und kann einfach ausgelesen werden. Daraus kann die Differenz c wie oben beschrieben berechnet werden und anschließend die Funktion D(x) auf den gesamten Text angewendet werden.
Am Ende werden die ersten zwanzig Buchstaben des entschlüsselten Textes ausgegeben und überprüft, ob das wirklich der entschlüsselte Text ist. Falls nicht, wird angenommen, dass der nächst häufigste Buchstabe das e war und der Text wird damit entschlüsselt. Das wird dreimal wiederholt, wenn dann kein richtiger Text gefunden wurde, kann angenommen werden, dass der eingegebene Text gar nicht Caesar-verschlüsselt war.
Entsprechen die ersten zwanzig Buchstaben einem korrekten Text, wird der ganze entschlüsselte Text als string zurückgegeben.

main_program()

Quellen:

23.06.21, Beispiel c=3:
https://www.ionos.de/digitalguide/fileadmin/DigitalGuide/Screenshots/caesar-verschluessel
png
27.07.21, Ascii Alphabet:
https://www.braingle.com/brainteasers/codes/images/ascii.png
23.06.21, Tabelle:
https://www.seo-woman.de/bilder/2012/buchstaben-haeufigkeit-deutsches-alphabet.jpg