Kryptologie

Mono- und polyalphabetische Chiffrierungen

Mathias Weigert
Miro Ljubicic

Kurs „Softwareprojekt 2“
Frühlingssemester 2012

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
Lagerstrasse 41
8004 Zürich

15. März 2012
aktualisierte Version vom 4. April 2012

Inhaltsverzeichnis

Einleitung

Motivation

Wir interessieren uns beide stark für das Thema Kryptologie und hatten bis jetzt keine Möglichkeit, uns auf diesem Gebiet sinnvoll zu vertiefen.

Da wir momentan für das Hauptstudium hin zu den Themen „Informationssicherheit und Kryptografie“ und „Netzwerktechnik“ tendieren, möchten wir die Chance nutzen, uns schon im Vorfeld intensiv mit den Grundlagen der Informationssicherheit zu befassen.

Der Einstieg in die Grundlagen der Kryptologie und deren Anwendung auf ein fundamentale Fallbeispiele erachten wir als gute Basis.

Typographische Konventionen

Um die Lesbarkeit des Textes zu erhöhen, werden folgende Konventionen verwendet:

`dies ist der klartext`	Klartext	Courier New, klein
`derschluessel`	Schlüssel	Courier New, klein, kursiv
`ERTS SDF ORT ZIFGHOIP`	Chiffrierter Text	Courier New, gross
$z \rightarrow (z+k) mod(n)$	Mathematische Formeln	Cambria Math, kursiv

Theoretische Grundlagen

Monoalphabetische Chiffrierung

Caesar-Chiffre (Verschiebe-Chiffre)

Der Caesar-Chiffre ist eines der ältesten Verschlüsslungsverfahren. Es zeichnet sich dadurch aus, das man ein Buchstabe des Alphabetes als Schlüssel nimmt und das Alphabet dann um X Stellen verschiebt.

Dieser Chiffre ist extrem einfach und ohne grossen Aufwand zu entschlüsseln. Da nur 26 Schlüssel existieren (Anzahl der Buchstaben im Alphabet).

Beispiel

```
yippie ya yeah schweinebacke (anderer Text bitte :-)
KUBBUQ KM KQMT EOTIQUZQNMOWQ
```

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

Der maximale Anzahl von Versuchen um den Caesar-Chiffre zu entschlüsseln beträgt 25 (entspricht der Anzahl Verschiebungen).

Umsetzung im Programm

Für die Verschlüsslung mit dem Caesar-Chiffre haben wir folgenden Algorithmus verwendet.

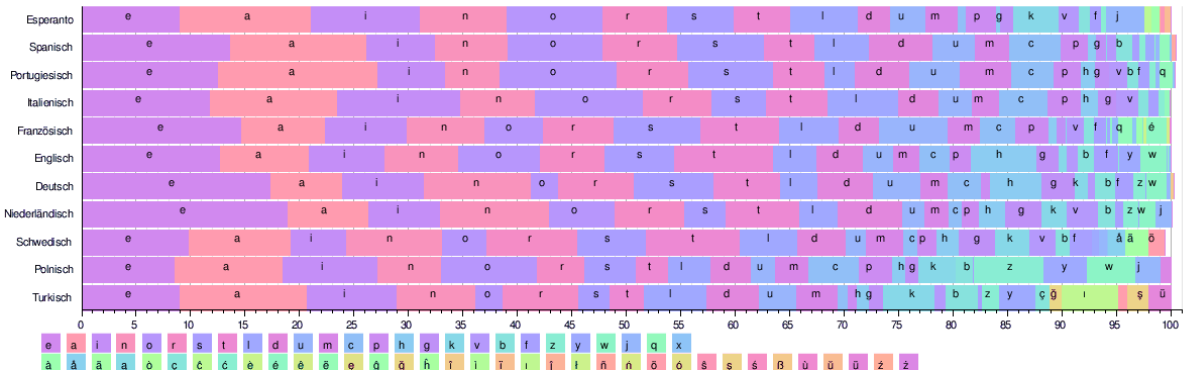
$$z \rightarrow (z + k) mod(n)$$

Wobei z das zu verschlüsselnde Zeichen, aus einem Alphabet von 0 bis n-1, ist und k der Wert des Schlüssels.

Zum Dechiffrieren verwenden wir Bruteforce, was bei einer maximalen Schlüsselmenge von 26 und einem Alphabet mit 26 Zeichen am meisten Sinn macht. Die durchschnittliche Dauer bei unserer Implementierung ist ca. 10 ms

Substitutions-Chiffre

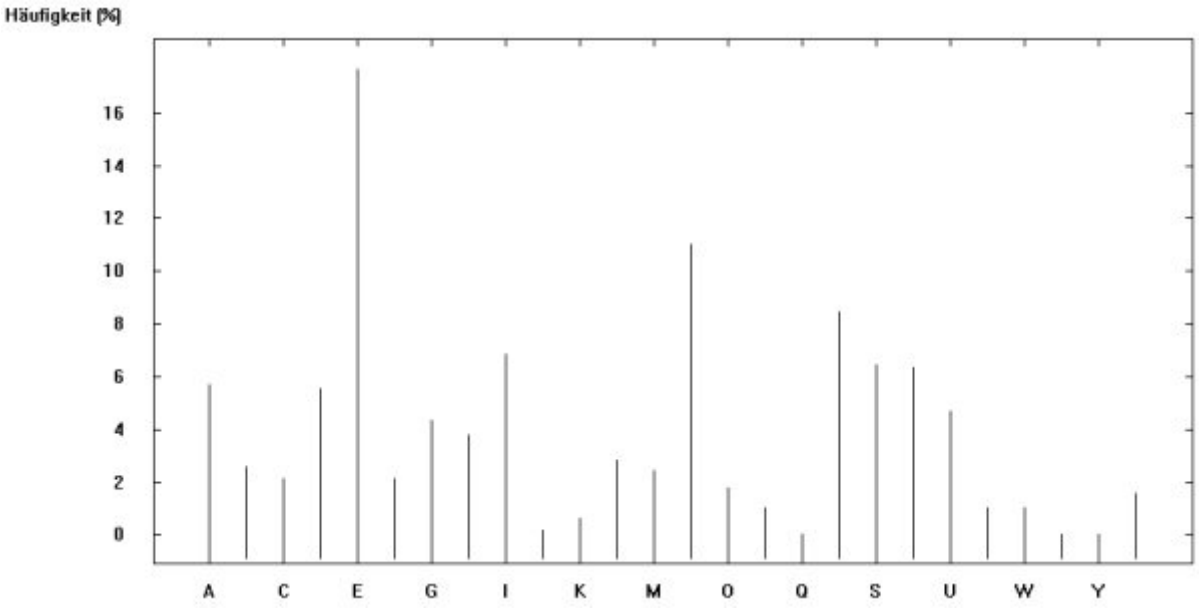
Der Substitution-Chiffre ist schon etwas komplexer da hierbei das Chiffre-Alphabet nicht mehr einfach verschoben wird, sondern völlig willkürlich neu angeordnet wird. Dadurch entsehen 26! Möglichkeiten (das sind 403‘291‘461‘126‘605‘635‘584‘000‘000). Diese Vielzahl der Möglichkeiten würden bei einem simplen Brutforce Algorithmus auch die Heutigen Hochleistungsrechner vor gewisse Probleme stellen. Die Schwachstelle dieses Chiffre liegt in der je nach Sprache ungleich verteilten Buchstaben.



Buchstabenverteilung europäischer Sprachen
http://de.wikipedia.org/wiki/Buchstabenh%C3%A4ufigkeit

Kryptoanalyse und Schwachstellen

Da bei der monoalphabetischen Chiffrierung jedes Zeichen auf genau ein anderes Zeichen abgebildet wird, kann diese Schwachstelle relativ einfach durch eine statistische Häufigkeitsanalyse der verwendeten Zeichen ausgenutzt werden.
Ist der Text lange genug, kann diese Häufigkeitsanalyse als verlässliche Referenz genutzt werden, um die verwendete Sprache abzuleiten.



Häufigkeitsanalyse eines deutschen Textes
http://www.cryptool-online.org/index.php?Itemid=117

Sobald die Sprache zuverlässig erkannt wurde, können weitere Techniken verwendet werden, um die Identität der einzelnen Zeichen zu ermitteln, hierzu einige Beispiele:

- Der Buchstabe ‚e' ist in den meisten europäischen Sprachen überdurchschnittlich dominant
- In der deutschen Sprache folgt mit hoher Wahrscheinlichkeit auf ein ‚c' entweder ein ‚h' (Kir**ch**e) oder ein ‚k' (Ba**ck**ofen) – daraus lassen sich also weitere Zeichen ableiten.
  Es können also sogenannte „Bigramme" (Gruppierung zweier aufeinanderfolgender Zeichen) gebildet werden, welche eine höhere Zuverlässigkeit bei der Erkennung erlauben. Dies lässt

sich beliebig ausdehnen (Trigramme = Gruppe dreier zusammenhängender Zeichen, etc.)
- je länger der Text und die verwendeten Wörter, umso eher kann mittels lexikalischer Suche (z.B. mittels Regulärer Ausdrücke) auf das Klartextwort geschlossen werden.

Allerdings bietet die Häufigkeitsanalyse nicht immer eine verlässliche Grundlage. Eines der bekanntesten Gegenbeispiele hierfür ist der Roman „La Disparition“ von Geroges Perec, in welchem der Autor strikte den Buchstaben ‚e‘ vermied.

Verbesserungsmöglichkeiten monoalphabetischer Verschlüsselungen

Eine naheliegende Verbesserung monoalphabetischer Verschlüsselungen wäre die Verwendung von Bigrammen: Anstatt jeweile nur ein Zeichen durch ein anderes zu ersetzen, können Bigramme auf andere (einzigartige) Bigramme abgebildet werden. Dies erschwert die statistische Häufigkeitsanalyse beträchtlich.

Polyalphabetische Chiffrierung

Im Gegensatz zur monoalphabetischen Chiffrierung – welche wie gezeigt

Vigenère-Chiffre

Das Vigenère-System ist das wohl bekannteste Beispiel der polyalphabetischen Chiffrierungen.
Die Schlüssel in diesem System sind Texte über dem lateinischen Alphabet.
Für die Ver- und Entschlüsselung wird das sogenannte „Vignenère-Quadrat“ und ein Codewort definiert.

Jedes einzelne Zeichen des Codeworts beschreibt hierbei die Zeile im Vignenère-Quadrat, welche für die Substitution verwendet werden soll. Nachdem jeder Buchstabe des Klartexts nacheinander den jeweiligen Zeilensubstitution durch das Codewort durchlaufen hat, ist der chiffrierte Text fertig.
Der chiffrierte Text kann durch Umkehrung des Codeworts auf die gleiche Art in den ursprünglichen Klartext umgewandelt werden.
Beispiel:

Klartext	`dasistdastorindemderschluesselist`
Schlüssel	*`keykeykeykeykeykeykeykeykeykey`*
Kryptotext	`NEQSWRNEQDSPSRBOQBOVQMLJEIQCIJSWR`

Dieses Verfahren ist um ein Vielfaches sicherer als monoalphabetische Chiffrierungen, da sich die Anzahl möglicher Schlüssel exponentiell erhöht.
Sie beträgt beim Vigenère-Code (auf dem lateinischen Alphabet mit 26 Zeichen):

Schlüssellänge (ohne Wiederholung)	Anzahl möglicher Schlüssel	
1	26	26
2	26*25	650
3	26*25*24	15.600
4	26*25*24*23	358.800
5	26*25*…*22	7.893.600
6	26*25*…*21	165.765.600
7	26*25*…*20	3.315.312.000
8	26*25*…*19	62.990.928.000
9	26*25*…*18	1.133.836.704.000
10	26*25*…*17	19.275.223.968.000

	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	C
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	C	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	C	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	C	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	C	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	C	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	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Vigenère-Quadrat
http://de.wikipedia.org/w/index.php?title=Datei:Vigenère_square.svg)

Kryptoanalyse und Schwachstellen

Trotz der verstärkten Sicherheit der Vigènere-Verschlüsselung, hat auch dieses Verfahren Schwachstellen. Diese lassen sich primär auf die Verwendung zu kurzer Schlüssel zurückführen. Wie im obigen Beispiel mit dem Schlüssel `key` gezeigt, entstehen im Kryptotext sich wiederholende Muster.

Der erste Schritt besteht also nun darin, die Schlüssellänge zu ermitteln. Hierzu gibt es mehrere Ansätze, welche für sich alleine nur beschränkte Aussagen erlauben, kombiniert aber die Schlüssellänge relativ sicher abschätzen lassen.

- Der **Kasisky-Test** liefert die Kandidaten für die Schlüssellänge, allerdings auch Vielfache davon (Ertel)
- Der **Friedman-Test** liefert eine Abschätzung über die ungefähren Wert für die Grössenordnung der Schlüssellänge (Ertel)

Implementierung

JAVA CODE HIER

Danksagung (erst zum Schluss)

Literaturverzeichnis

- 🕮 *KRYPTOLOGIE – ALGEBRAISCHE METHODEN UND ALGORITHMEN*
  (Christian Karpfinger ¦ Hubert Kiechle)
- 🕮 *DEUTSCHE WIKIPEDIASEITE (BUCHSTABENHÄUFIGKEIT)*
  http://de.wikipedia.org/wiki/Buchstabenh%C3%A4ufigkeit
- 🕮 *ANGEWANDTE KRYPTOGRAPHIE*
  (Wolfgang Ertel, Carl Hanser Verlag, 3. aktualisierte Auflage, 978-3-4464-1195-1)
- 🕮 *THE CODE BOOK – THE SECRET HISTORY OF CODES AND CODE-BREAKING*
  (Simon Singh)
- 🕮 *EINFÜHRUNG IN DIE KRYPTOLOGIE*
  (Karin Freiermuth ¦ Juraj Hromkovič ¦ Lucia Keller ¦ Björn Steffen)