Symmetrische Kryptologie und ihre Veranschaulichung

Koch, A.
DOI: 10.18420/ibis-02-02-07

Zusammenfassung

Die Kryptologie, die Wissenschaft des Geheimen, gliedert sich in Kryptographie, die Wissenschaft der Geheimschriften und in Kryptoanalyse, die sich mit der Untersuchung der Sicherheit von Geheimschriften beschäftigt. Die Lehrpläne der Bundesländer enthalten eine große Vielfalt an Themen der Kryptologie. Beispielsweise sehen die Lehrpläne baden-württembergischer Gymnasien monoalphabetische Substitutionsverschlüsselungen und ihre Kryptoanalyse in Klassenstufe 7, polyalphabetische Substitutionsverschlüsselungen und Transpositionsverfahren in Klassenstufe 8 sowie symmetrische und asymmetrische Verfahren in der Sekundarstufe II vor.

Mit Verschlüsselungsverfahren, häufig kurz als Verschlüsselungen oder noch knapper als Verfahren bezeichnet, sind formal injektive Codierungen gemeint, deren Umkehrbarkeit auf Kenntnis eines Geheimnisses beruht. Dieses Geheimnis kann auch das Verfahren selbst sein. Die Umkehrung entspricht dem Entschlüsselungsverfahren. Parameter, die zur Durchführung der Ver- und Entschlüsselung benötigt werden, werden als Schlüssel bezeichnet. Das nach dem Niederländer Auguste Kerckhoffs (1835-1903) benannte Kerckhoffs'sche Prinzip besagt, dass die Sicherheit eines Verfahrens nicht auf Geheimhaltung des Verfahrens selbst beruhen darf. Dies impliziert als notwendige Bedingung, dass sichere Verfahren Schlüssel bedürfen.

In diesem Artikel werden Möglichkeiten für einen anschaulichen Unterricht zur symmetrischen Kryptologie aufgezeigt. Bei symmetrischen Verfahren sind die Schlüssel zur Entschlüsselung und Verschlüsselung gleich.

Transpositionsverschlüsselungen

Transpositionsverschlüsselungen sind Verfahren, bei denen die Positionen der Zeichen des Klartexts vertauscht werden, um den Geheimtext zu bilden. Die Spartaner verwendeten mit der Skytale (griechisch skytálē, für „Stab"), einem runden Holzstab, bereits vor über 2500 Jahren ein derartiges Verfahren, um ihre Nachrichten zu verschlüsseln. Dazu wickelten sie einen Pergamentstreifen um ihre Skytale, beschrieben ihn längs und drehten am Ende angekommen die Skytale weiter. Abbildung 1 zeigt die moderne Variante einer Skytale aus Gardinenstange und zusammengeklebten DIN A4-Blattstreifen.

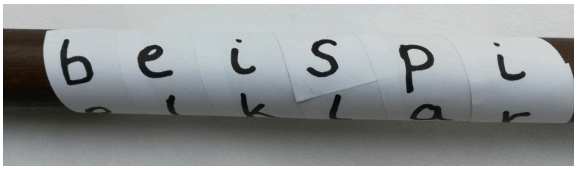

Abbildung 1: Skytale (Andreas Koch / CC BY-SA 4.0)

Der Klartext `beispielklartext` aus Abbildung 1 wird zunächst auf dem Papierstreifen notiert, der dabei sukzessive gedreht wird. Anschließend wird der Streifen abgerollt und die Leerzeichen werden entfernt. So ergibt sich der Geheimtext `BETELEIKXSLTPAIR` (Geheimtexte werden in diesem Artikel zur besseren Abgrenzung vom Klartext groß geschrieben). Durch das Entfernen der Leerzeichen wird die Kryptoanalyse erschwert. Der Schlüssel entspricht der Skytale selbst bzw. der Anzahl der Windungen, also in diesem Beispiel der Zahl 6. Er wird demzufolge durch den Umfang der Skytale festgelegt. Bei fünf Windungen als Schlüssel würde sich der Geheimtext `BIATEERILTS-KEPLX` ergeben.

Als Veranschaulichung, insbesondere zur Vorbereitung auf die Implementierung des Verfahrens, bietet sich ein Matrix-ähnliches Schema wie in Abbildung 2 an.

```
#s(palten)=schluessel=6
#z(eilen)=3

z/s 0   1   2   3   4   5
0 | b | e | i | s | p | i |
1 | e | l | k | l | a | r |
2 | t | e | x | t |   |   |

Durchlaufreihenfolge (von oben
links nach unten rechts):

0+0*6=0,  0+1*6=6,   0+2*6=12
1+0*6=1,  1+1*6=7,   1+2*6=13
2+0*6=2,  2+1*6=8,   2+2*6=14
3+0*6=3,  3+1*6=9,   3+2*6=15
4+0*6=4,  4+1*6=10,  4+2*6=16
5+0*6=5,  5+1*6=11,  5+2*6=17
```

Abbildung 2: Veranschaulichung der Verschlüsselung (Andreas Koch / CC BY-SA 4.0)

```
01 public static String Skytale(String text, int schluessel)
02 {
03   String geheimtext = "";
04   int z = (text.length() / schluessel) + 1; // Zeilenanzahl
05
06   for (int i=0; i<schluessel; i++) {
07     for (int j=0; j<z; j++) {
08       if (i+j*schluessel<text.length()) { // Platz gefüllt?
09         geheimtext += text.charAt( (i+j*schluessel)  );
10       }
11     }
12   }
13   return geheimtext;
14 }
```

Abbildung 3: Java-Implementierung der Skytale-Verschlüsselung (Andreas Koch / CC BY-SA 4.0)

Der Klartext kann demzufolge zeilenweise in ein zweidimensionales Array geschrieben werden, das anschließend spaltenweise ausgelesen wird, um den Geheimtext zu erhalten. Die tatsächliche Verwendung des Arrays ist in einer Implementierung der Verschlüsselung allerdings nicht notwendig, da die Indizes auch berechnet werden können (siehe die Rechnungen zur Durchlaufreihenfolge in Abbildung 2). Zu beachten ist, dass vorab überprüft werden muss, ob der Eintrag tatsächlich existiert, wie die Beispiele der Indizes 16 und 17 aus Abbildung 2 zeigen.

Die Implementierung der Entschlüsselung erfolgt in ähnlicher Weise. Die Skytale erfüllt das Kerckhoffs'sche Prinzip nicht, da die Anzahl der Schlüssel im Allgemeinen gering ist. Eine obere Schranke für die Schlüssellänge ist die Klartextlänge. Mit überschaubarem Aufwand können per Bruteforce verschiedene Windungszahlen in einem Schema wie in Abbildung 2 durchprobiert werden, bis sich ein sinnvoller Text ergibt.

Substitutionsverschlüsselungen

Bei Substitutionsverschlüsselungen wird jeweils ein Zeichen des Klartexts durch genau ein Zeichen (monoalphabetische Verschlüsselungen) wie bei der Cäsar-Verschlüsselung oder mehrere Zeichen (polyalphabetische Verschlüsselungen) des Geheimtextalphabets wie bei der Vigenère-Verschlüsselung ersetzt.

Cäsar-Verschlüsselung

„... wenn etwas Geheimes zu überbringen war, schrieb er in Zeichen, das heißt, er ordnete die Buchstaben so, dass kein Wort gelesen werden konnte: Um diese zu lesen, tauscht man den vierten Buchstaben, also D für A aus und ebenso mit den restlichen.“ (Aus „Sueton: De Vita Caesarum: Divus Julius LVI“.)

Die Cäsar-Verschlüsselung ist demzufolge eine monoalphabetische Substitutionsverschlüsselung, die das Kerckhoffs'sche Prinzip nicht erfüllt, da keine Schlüssel zur Ver- und Entschlüsselung benötigt werden.

Mithilfe einer Codetabelle wie in Abbildung 4 kann jede monoalphabetische Substitutionsverschlüsselung mit endlichem Alphabet definiert und dargestellt werden.

Beispielsweise besitzt der Klartext `geheim` den Geheimtext `JHKHLP`. Die Verschiebezahl 3, um die ein Klartextbuchstabe im Alphabet verschoben wird, kann variiert werden, wodurch die Cäsar-Verschlüsselung zu einem Verfahren mit Schlüssel wird. Unter den 26 Schlüsseln befinden sich abzüglich der Verschiebezahl 0 genau 25 sinnvolle Schlüssel. Mithilfe der sogenannten Cäsar-Scheibe aus Abbildung 5 können die 26 Verschlüsselungen veranschaulicht und Ver- und Entschlüsselung von Texten schnell realisiert werden.

Für eine Implementierung des Verschlüsse-

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

Abbildung 4: Codetabelle zum Cäsar-Verfahren (Andreas Koch / CC BY-SA 4.0)

lungsverfahrens kann der Schlüssel entweder als Zahl oder als Buchstabe in `Char`- oder `String`-Codierung eingegeben werden. Die Implementierung aus Abbildung 6 zeigt letztere Variante für die Verschlüsselung von Kleinbuchstaben, deren ASCII-Zahlenbereich von 97 zur Codierung des Buchstabens „a“ bis 122 für „z“ reicht. In Zeile 9 muss daher vom ASCII-Wert des Schlüsselbuchstabens der Wert 97 abgezogen werden, um die korrekte Verschiebezahl zu berechnen. In den Zeilen 10 und 11 wird überprüft, ob der ASCII-Bereich der Kleinbuchstaben überschritten wurde. Falls ja, muss der ASCII-Wert korrigiert werden, was die Cäsar-Scheibe durch ihren zyklischen Aufbau „automatisch“ macht.

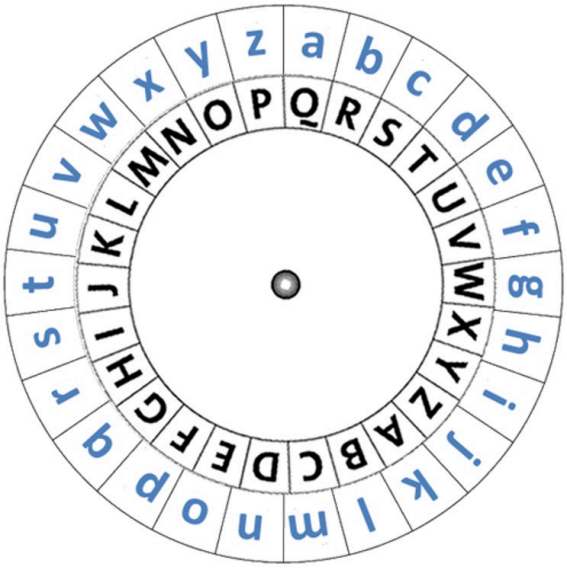

Abbildung 5: Cäsar-Scheibe (Andreas Koch / CC BY-SA 4.0)

Die Implementierung der Entschlüsselung ergibt sich durch Anpassung der Zeilen 9 bis 11.

Geheimtexte einer monoalphabetischen Substitutionsverschlüsselung können mithilfe einer Häufigkeitsanalyse ohne Kenntnis des Verfahrens entschlüsselt werden, unter der Voraussetzung, dass der Klartext zumindest näherungsweise die charakteristische statistische Häufigkeit des zugrundeliegenden Sprachalphabets aufweist. Die Häufigkeiten der Geheimtextbuchstaben werden lediglich permutiert und erlauben so die Rekonstruktion der Zuordnung zum passenden Klartextbuchstaben, sofern o.g. Voraussetzung erfüllt ist. Ein Vergleich der Abbildungen 7 und 8 lässt den Schluss zu, dass der exemplarische Geheimtext Cäsar-verschlüsselt ist, da sich die Häufigkeiten lediglich um eine Konstante unterscheiden, die dem Schlüssel, der Verschiebezahl 3, entspricht.

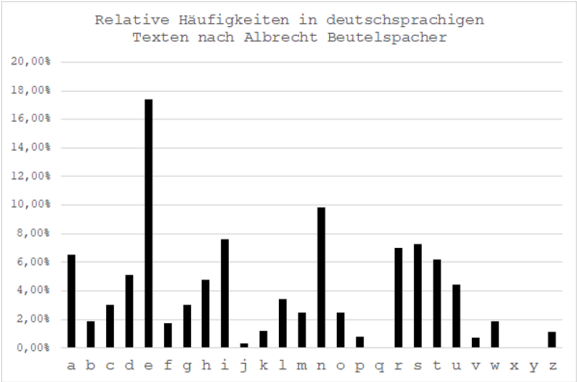

Abbildung 7: Relative Häufigkeiten in deutsprachigen Texten nach Albrecht Beutelspacher. Grafische Aufbereitung vom Autor (Andreas Koch / CC BY-SA 4.0)

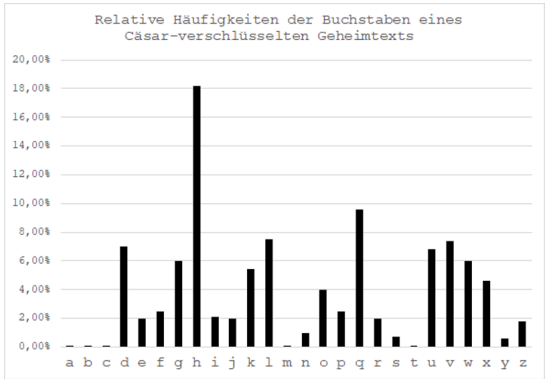

Abbildung 8: Relative Häufigkeiten der Buchstaben eines Cäsar-verschlüsselten Geheimtexts

Das Kerckhoffs'sche Prinzip ist demzufolge auch bei der Cäsar-Verschlüsselung mit Schlüssel nicht erfüllt.

Vigenère-Verschlüsselung

Die Vigenère-Verschlüsselung stammt vom französischen Kryptographen Blaise de Vigenère aus dem Jahr 1586. Sie setzt mehrere Cäsar-Verschlüsselungen positionsabhängig ein, womit sie ein Beispiel für eine polyalphabetische

```
01 public static String Caesar(String text, String schluessel)
02 {
03   String geheimtext = "";
04     for (int i=0; i<text.length(); i++) {
05       char c = text.charAt(i);
06       int  z = (int) c;
07       int  s = (int) schluessel.charAt( 0 );
08       if ((97<=z) && (z<=122)) {
09         z += s-97;
10         if (z>122) {
11           z -= 26;
12         }
13       }
14       geheimtext += (char) z;
15     }
16     return geheimtext;
17 }
```

Abbildung 6: Java-Implementierung der Cäsar-Verschlüsselung (Andreas Koch / CC BY-SA 4.0)

Substitutionsverschlüsselung ist. Anstelle eines einzelnen Buchstabens werden Wörter als Schlüssel verwendet. Damit eignet sich eine Häufigkeitsanalyse, die auf einen Vigenère-verschlüsselten Geheimtext angewendet wird, im Allgemeinen nicht zu dessen Kryptoanalyse. Abbildung 9 zeigt exemplarisch, wie die relativen Häufigkeiten der Buchstaben um den Wert einer Gleichverteilung auf 26 Buchstaben streuen. Die Vigenère-Verschlüsselung glättet sozusagen die charakteristischen statistischen Häufigkeiten des Sprachalphabets und räumt damit die zentrale Schwachstelle monoalphabetischer Substitutionsverschlüsselungen aus.

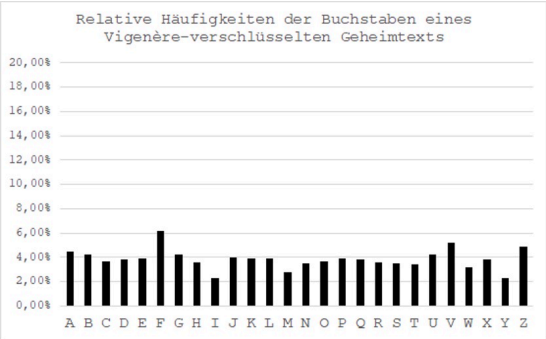

Abbildung 9: Relative Häufigkeiten der Buchstaben eines Vigenère-verschlüsselten Geheimtexts (Andreas Koch / CC BY-SA 4.0)

Ver- und Entschlüsselung lassen sich wie in den Abbildungen 10 und 11 veranschaulichen.

Verschlüsselung mit Schlüssel *ulm*

Klartext	b	a	n	a	n	e	n
+	u	l	m	u	l	m	u
Geheimtext	V	L	Z	U	Y	Q	H

Abbildung 10: Vigenère-Verschlüsselung (Andreas Koch / CC BY-SA 4.0)

Verschlüsselung mit Schlüssel *ulm*

Klartext	b	a	n	a	n	e	n
+	u	l	m	u	l	m	u
Geheimtext	V	L	Z	U	Y	Q	H

Cäsar-Verschlüsselung mit Schlüssel **u**

Abbildung 11: Vigenère-Entschlüsselung (Andreas Koch / CC BY-SA 4.0)

Ein methodisches Hilfsmittel für Schülerinnen und Schüler stellt das sogenannten Vigenère-Quadrat aus Abbildung 12 dar, das alle 26 Cäsar-Verschlüsselungen tabellarisch auflistet und so gegenüber der Cäsar-Scheibe, die vor jedem Ver- und Entschlüsselungsvorgang korrekt einzustellen ist, ein zügigeres Anwenden des Verfahrens ermöglicht.

Die Implementierung der Vigenère-Verschlüsselung lässt sich auf Basis der Cäsar-Verschlüsselung durch wenige Anpassungen vornehmen, wenn das Einlesen des Schlüsselbuchstabens wie in Abbildung 6 bereits als `String` erfolgt. Der Schlüssel muss zyklisch gelesen werden, da seine Länge n die des Texts unterschreiten kann, was der Regelfall ist. Dies lässt sich entweder mithilfe der Modulo-Arithmetik lösen, indem die Schlüsselposition durch `j mod n` für Textpositionen `j` berechnet wird, oder wie in Abbildung 13 durch Verwenden einer weiteren Variable für den Schlüsselindex.

Schlüsselbuchstabe	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
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

Abbildung 12: Vigenère-Quadrat (Andreas Koch / CC BY-SA 4.0)

Ein Verfahren zur Kryptoanalyse der Vigenère-Verschlüsselung wurde erst ca. dreihundert Jahre nach ihrer Einführung publik. Der preußische General Friedhelm Kasiski veröffentliche im Jahr 1863 ein Verfahren zur Bestimmung der Schlüssellänge n. Wenn diese bekannt ist, kann ein Vigenère-verschlüsselter Geheimtext in n Cäsar-verschlüsselte Geheimtexte durch Auswahl der jeweils n-ten Buchstaben zerlegt werden. Nun liefern n Häufigkeitsanalysen das Schlüsselwort.

Die Bestimmung der Schlüssellänge macht sich zu Nutze, dass ein Schlüsselwort im Allgemeinen erheblich kürzer als der Klartext ist. Dadurch ergeben sich im Geheimtext gleiche Textteile, wenn der Schlüssel bei seiner zyklischen Verwendung auf den gleichen Klartext trifft. Umgekehrt gilt: Mit einer Wahrscheinlichkeit von $1 - 26^{-2}$ sind gleiche Bigramme im Geheimtext mit dem gleichen Schlüsselteil verschlüsselt, gleiche Trigramme mit einer Wahrscheinlichkeit von $1 - 26^{-3}$ usw. Die Abstände gleicher Textteile im Geheimtext liefern demzufolge mit entsprechender Wahrscheinlichkeit Kandidaten für Vielfache der Schlüssellänge. Durch sukzessive Auswahl gleicher Primfaktoren dieser Ab-

```
01 public static String Vigenere(String text, String schluessel)
02 {
03   String geheimtext = "";
04   int j=0;
05   for (int i=0; i<text.length(); i++) {
06     char c = text.charAt(i);
07     int  z = (int) c;
08     int  s = (int) schluessel.charAt( j );
09     if ((97<=z) && (z<=122)) {
10       z += s-97;
11       if (z>122) {
12         z -= 26;
13       }
14     }
15     geheimtext += (char) z;
16     j++;
17     if (j>schluessel.length()) {
18       j = 0;
19     }
20   }
21       return geheimtext;
22 }
```

Abbildung 13: Java-Implementierung der Vigenère-Verschlüsselung (Andreas Koch / CC BY-SA 4.0)

stände lässt sich so meist die Schlüssellänge berechnen wie Abbildung 14 beispielhaft zeigt.

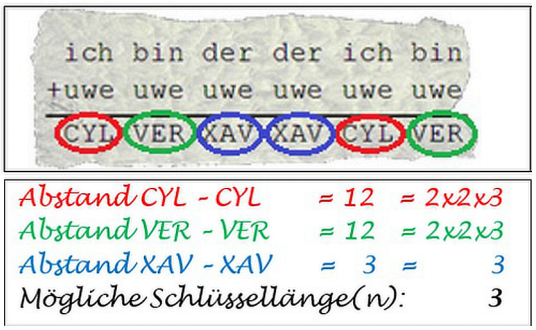

Abbildung 14: Bestimmung der Schlüssellänge mittels Kasiski-Verfahren (Andreas Koch / CC BY-SA 4.0)

Die Sicherheit des Vigenère-Verfahrens hängt somit von der Wahl des Schlüssels ab. Wenn er mindestens so lang wie der Klartext ist und die einzelnen Schlüsselbuchstaben für jeden Verschlüsselungsvorgang zufällig gewählt werden, spricht man von einem absolut sicheren Verfahren, einem One-Time-Pad (Einmal-Abriss-Block). In diesem Fall ist das Kerckhoffs'sche Prinzip erfüllt.

Visuelle Verschlüsselung

Das Verfahren der Visuellen Verschlüsselung von Moni Naor und Adi Shamir aus dem Jahr 1993 ist ein One-Time-Pad und erfüllt damit das Kerckhoffs'sche Prinzip. Durch Übereinanderlegen zweier Folien, von denen eine dem Geheimtext und die andere dem Schlüssel entspricht, entsteht das Originalbild. Für sich genommen enthalten die Folien laut Naor und Shamir nur „random noise" (zufälliges Rauschen), d.h. sie geben keinerlei Information über das Originalbild preis.

Am Beispiel von schwarz-weißen Pixelbildern lässt sich das Verfahren wie in Abbildung 16 veranschaulichen. Ein Pixel im Originalbild (schwarz oder weiß) wird jeweils zu vier Pixeln (grau) auf jeder Folie verschlüsselt. Die Wahl des Diagonalmusters auf der ersten Folie (Schlüssel) erfolgt zufällig, die Wahl des Diagonalmusters auf der zweiten Folie dazu gespiegelt.

Abbildung 15: Visuelle Verschlüsselung (Andreas Koch / CC BY-SA 4.0)

Die Entschlüsselung erfolgt analog. Dabei ist zu beachten, dass weiße Pixel im Originalbild vier Pixeln mit Diagonalmuster entsprechen, wodurch der Eindruck eines grauen Pixels entsteht.

Das Verfahren lässt sich auf digitale schwarz-weiße Pixelbilder übertragen, indem eine passende Zuordnung der Pixelkonstellationen von schwarz, weiß und grau zu den Bits 0 und 1 erfolgt. Eine mögliche Variante zeigt Abbil-

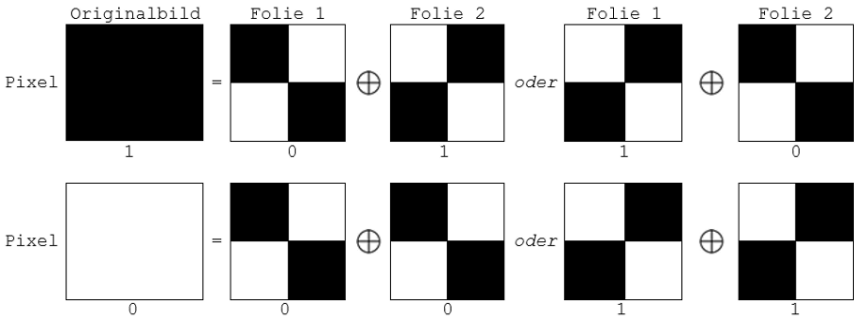

Abbildung 16: Verschlüsselung von schwarz-weißen Pixelbildern (Andreas Koch / CC BY-SA 4.0)

dung 16: Zuerst werden dem schwarzen Pixel (im Beispiel 1) und dem weißen Pixel (0) die Bitwerte 0 bzw. 1 zugeordnet. Da die beiden Folien lediglich die binäre Information abbilden, ob die Diagonalmuster gleich sind und damit einem weißen Pixel im Originalbild entsprechen oder verschieden sind und damit einem schwarzen Pixel im Originalbild entsprechen, kann ihnen ebenfalls der Bitwert 0 oder 1 zugeordnet werden. Für die erste Folie wird er zufällig gewählt und für die zweite mithilfe der binären XOR-Funktion berechnet, um die Gleichheit bzw. Verschiedenheit der Diagonalmuster korrekt abzubilden. Der Bitwert des Pixels der Folie 2 berechnet sich für ein schwarzes Pixel im Originalbild und des durch 0 zufällig gewählten Pixelbitwerts auf Folie 1 zu `1 XOR 0 = 1`. Wäre der Pixelbitwert auf Folie 1 stattdessen 1, so ist der Pixelbitwert auf Folie 2 durch `1 XOR 1 = 0` gegeben.

Abbildung 17 zeigt den Screenshot eines Programms zur Visuellen Verschlüsselung von schwarz-weißen Pixelbildern, das unter www.andreas-koch.de heruntergeladen werden kann.

Eine Implementierung ohne grafische Ausgabe zeigt Abbildung 18. Das Originalbild wird als `String` eingelesen, der nur mit 0 oder 1 zu belegen ist. So erleichtert sich die Verarbeitung, da beim Konkatenieren der Ausgabe-Strings

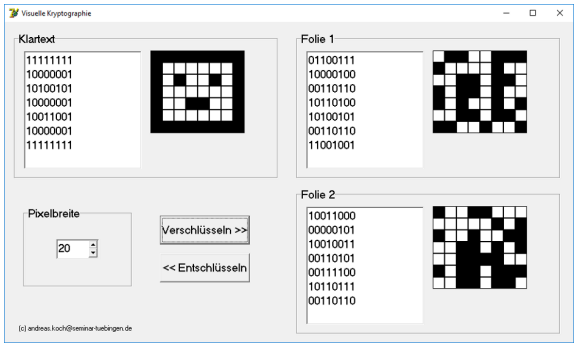

Abbildung 17: Screenshot eines Programms zur Visuellen Verschlüsselung (Andreas Koch / CC BY-SA 4.0)

keine Typecasts notwendig sind. Dadurch muss allerdings in Zeile 9 der ASCII-Wert der `char`-Variablen zuerst in einen `int`-Wert umgerechnet werden (ASCII-Code 48 bei 0 und 49 bei 1). Alternativ wäre ein Typecast mittels Methodenaufruf `Character.getNumericValue(c)` denkbar.

Eine Methode zur Entschlüsselung benötigt zwei Parameter, nämlich die Strings der Bitwerte der beiden Folien. Die Implementierung ist verglichen mit der Entschlüsselung leichter. Die Bitwerte der Strings müssen sukzessive XOR-verknüpft ausgegeben werden. Eine Zufallszahl muss nicht bestimmt werden.

Die Herleitung der Implementierung zeigt auch, dass das Verfahren verallgemeinert werden

```
01  public static void Visuell(String bits)
02  {
03    String folie1 = "";
04    String folie2 = "";
05    for (int i=0; i<bits.length(); i++) {
06      char c = bits.charAt(i);
07      int z = (int) (Math.random()*2); // 0 oder 1 würfeln
08        folie1 += z;
09        if (z != ((int) c - 48)) { // XOR
10          folie2 += "1";
11        } else {
12           folie2 += "0";
13        }
14    }
15    System.out.println( "Folie 1: " + folie1 );
16    System.out.println( "Folie 2: " + folie2 );
17  }
```

Abbildung 18: Java-Implementierung der Visuellen Verschlüsselung (Andreas Koch / CC BY-SA 4.0)

kann: Die Art des Musters der Folien muss sich nicht auf Diagonalen beschränken. Die Anzahl der Folien kann auf n Stück ausgeweitet werden, indem die Pixelzahl pro Originalbildpixel auf n*n Stück erhöht wird.

Unterrichtsgang

Die vorgestellten Verfahren eignen sich zur Ausgestaltung eines acht Stunden umfassenden Unterrichtsgangs zum Thema „Symmetrische Kryptologie“ unter Veranschlagung von jeweils einer Doppelstunde pro folgendem Verfahren: Transposition am Beispiel der Skytale, monoalphabetische Substitution am Beispiel der Cäsar-Verschlüsselung, polyalphabetische Substitution am Beispiel der Vigenère-Verschlüsselung und One-Time-Pad am Beispiel der Visuellen Verschlüsselung.

Die vorgestellten Veranschaulichungen ermöglichen auch einen Unterricht in der Sekundarstufe I. Hier sollte der Schwerpunkt auf der händischen Durchführung und Beurteilung der Sicherheit der Verfahren liegen. In der Sekundarstufe II sollte er auf die Implementierung der Verfahren gelegt werden. Nach einer gemeinsamen Erarbeitung der Implementierung der Verschlüsselung bietet sich zur Lernzielkontrolle eine Übungsphase an, in der die Schülerinnen und Schüler die Entschlüsselung selbstständig implementieren. Zu beachten ist, dass bei der Skytale der Schwierigkeitsgrad einer Implementierung der Entschlüsselung aufgrund der notwendigen Anpassung der Indizes gegenüber der Verschlüsselung höher als bei den anderen Verfahren ist.

Java-Implementierungen aller Verfahren für BlueJ inklusive einer Vorlage für Schülerinnen und Schüler können angefragt werden.

Die Diskussion des logistischen Problems des Schlüsselaustauschs bei symmetrischen Verfahren ist ein sinnvoller Ausgangspunkt für die Einführung asymmetrischer Verfahren.

Quellen

Alle Webseiten/Links wurden zuletzt geprüft am 02.01.2024.

Beutelspacher, A. (2005): Kryptologie. 7. Auflage. Vieweg Verlagsgesellschaft, Wiesbaden, Seite 10.

de Vigenère, B. (1586): Traicté des Chiffres ou Secrètes Manières d'Escrire. Paris.

Kasiski, F. (1863): Geheimschriften und die Dechiffrierkunst – Mit besonderer Berücksichtigung der deutschen und der französischen Sprache. Mittler und Sohn, Berlin.

Ministerium für Kultus, Jugend und Sport Baden-Württemberg (2016): Bildungsplan Aufbaukurs Informatik 7. https://www.bildungsplaene-bw.de/,Lde/LS/BP2016BW/ALLG/SEK1/INF7.

Naor, M.; Shamir, A. (1994): Visual Cryptography, EUROCRYPT, S. 1-12.

Lizenz

Kontakt

Andreas Koch
E-Mail: kocha@posteo.de