Grundlagen

Welche Ziele verfolgt man beim Codieren von Informationen?
Es gibt zwei Grundlinien, weshalb man Informationen codiert:

1. Die Art der Speicherung oder Übermittlung soll nach irgendeinem Kriterium optimiert oder auch einfach nur anders dargestellt werden.
2. Der Informationsgehalt in Daten soll bei Speicherung oder Übermittlung vor unberechtigtem Zugriff geschützt werden.

Beispiele zu 1. sind:

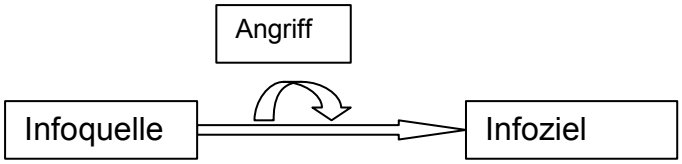
- der Strichcode, der bei der Kennzeichnung von Waren verwendet wird: Die Daten sollen in eine (leichter) maschinenlesbare Form gebracht werden
- die Codierung von Bilddateninformationen, mit dem Ziel, den Umfang der Daten zu verringern (Bilddatenkompression)

Beispiele zu 2. sind:

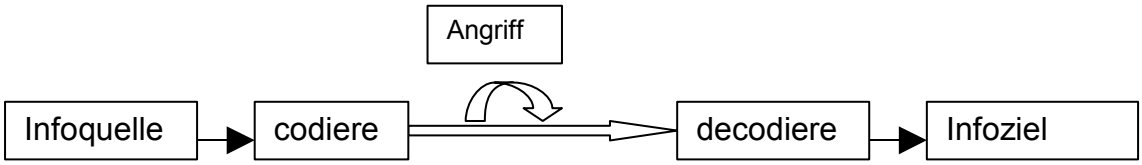
- die Verschlüsselung von Daten auf der Festplatte mit dem Ziel, dass sie jemand fremdes zwar auslesen, aber nicht verstehen kann
- die Verschlüsselung von Daten vor dem Versenden durch das Internet mit dem Ziel, dass jemand anders sie bei der Übermittlung zwar lesen aber nicht verstehen oder so verändern kann, dass der rechtmäßige Empfänger das nicht bemerkt.

Wir werden uns zunächst (?) nur mit der zweiten Variante beschäftigen.

Problem



Lesbarkeit wird durch Verschlüsselung verhindert :



Wie geht man vor ?

Verschlüsselung :
Die Informationen werden so „durcheinander gewürfelt“, dass sie von einem anderen zwar gelesen, aber nicht verstanden werden können.

Nun nützt es nichts, wenn bei diesem Durcheinanderwürfeln der ursprüngliche Informationsgehalt endgültig verloren geht, so dass er nicht mehr wieder hergestellt werden kann. Das bedeutet aber, dass die Informationen prinzipiell wiederherstellbar bleiben müssen. Auf codierte Daten bleiben daher zwei Zugriffe möglich :

dechiffrieren ...	
gewollt, autorisiert	nicht gewollt, unautorisiert
Das Wiederherstellen der ursprünglichen Informationen durch den gewünschten Nutzer mit dem ihm bekannten Schlüssel und Verfahren.	Das Wiederherstellen der ursprünglichen Informationen durch einen ungewollten Nutzer mit dem irgendwie beschafften Schlüssel und Verfahren.

Historische Vorbilder

Zu den einfachen Beispielen von Verschlüsselungen, deren Anwendungen historisch belegt sind, gehören ***skytale*** und ***Caesar*** – Verschlüsselung, wobei bei letzterem die Zuordnung zu Caesar nicht sicher belegt ist.
Beim ***skytale*** , dem griechischen Historiker Plutarch nach von den Spartanern eingesetzt, wird ein zu beschreibender Streifen um einen Stab gelegt und dann in Stabrichtung zeilenweise nacheinander beschriftet. Die Entschlüsselung erfolgt dann, indem man den Streifen um einen Stab mit derselben Dicke wickelt.

Der Verschlüsselungsalgorithmus besteht in einer Positionsverschiebung der Buchstaben des Textes. Der verschlüsselte Text enthält exakt dieselben Buchstaben wie der Originaltext, sie befinden sich nur an anderen Positionen. Die Entschlüsselung erfolgt durch die Rückverschiebung. Ein Verschlüsselungsalgorithmus, bei dem die ***Positionen der Inhalte*** verschoben werden, heißt ***Transpositionsalgorithmus***, nach dem lateinischen Wort transponere = hinüberbringen, verschieben.

Beim Caesar – Verschlüsselungsalgorithmus wird jeder Buchstabe des Textes durch einen anderen Buchstaben des Alphabetes ersetzt. Ein solcher Verschlüsselungsalgorithmus heißt nach dem lateinischen Wort für ersetzen = substituere ***Substitutionsalgorithmus***. Der Caesar – Verschlüsselungsalgorithmus arbeitet allerdings mit einem Spezialfall von Codierungstabellen, einem ***Verschiebechiffre***, weil die Ersetzung für alle Buchstaben nach demselben Verfahren erfolgt, nämlich um eine bestimmte Zahl von Positionen der Buchstaben im Alphabet.

Eine komplette Übersetzungstabelle für eine Caesar – Verschlüsselung könnte folgendermaßen aussehen:

Übersetzungstabelle

KEY	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

Ist das Wort HUNDERT und der Schluesselbuchstabe d , dann würde es also durch kxqghuw ersetzt werden.

Beide Verschlüsselungsverfahren, sowohl ***skytale*** als auch ***Caesar*** leiden unter zwei schweren Mängeln:

1. Der Algorithmus ist nicht besonders kompliziert.
2. Es gibt nur eine geringe Zahl von Schlüsseln.

Beide lassen sich also sehr leicht durch Probieren knacken, wenn man das Verfahren, das verwendet wurde, erkannt hat.

Beim Caesar – Verschlüsselungsverfahren reicht die Kenntnis eines einzelnen Buchstaben. Leichter geht es natürlich mit Durchprobieren aller 25 möglichen Schlüssel: Ist der Text dann lesbar, hat man den Schlüssel gefunden.

Man kann das Finden des Schlüssels sogar automatisieren, wenn man eine statistische Analyse der Buchstabenhäufigkeit durchführt. Dazu untersucht man die Häufigkeit, mit der die einzelnen Buchstaben im codierten Text auftreten und vergleicht sie mit typischen Buchstabenhäufigkeiten. Dabei kann man dann in der Regel den Buchstaben e zuordnen und somit den Text automatisch entschlüsseln.

Die folgende Tabelle der Buchstabenhäufigkeiten in deutschen Texten entstammt Beutelspacher, Kryptologie.

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

Häufigkeitsanalyse

Der wichtigste Zugang zum Knacken einer Chiffrierung sind statistische Analysen, bei der Caesarverschlüsselung eine Statistik der Häufigkeit der auftretenden Buchstaben des Codetextes. Der Vergleich mit der (vorher angegebenen) Tabelle zeigt dann eine charakteristische Verteilung, die bei inhaltlich nicht sehr speziellen Texten, die lang genug sind, zu einer fast hundertprozentigen Identifikation führen.

Dabei ist wegen der Redundanz von Texten noch nicht einmal eine hundertprozentige Identifikation notwendig. Wenige Fehler lassen sich sehr schnell per Hand bearbeiten.

Kann man das Caesar – Verfahren verbessern ?

Ein Verfahren, das etwas besser ist, verschiebt nicht alle Buchstaben nach demselben Muster, sondern schüttelt alle oder möglichst viele Buchstaben durcheinander.

Man nehme sich ein Schlüsselwort , z.B.: G E H E I M S C H R I F T , entferne alle wiederholten Buchstaben , so dass in unserem Beispiel G E H I M S C R F T entstünde und setze es unter eine Alphabetzeile mit einem gewählten Startbuchstaben, hier ist es das J . So entsteht z.B. folgender Schlüssel:

KEY 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

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

Auch einen solchen Schlüssel kann man aber mit akzeptablem Aufwand über Häufigkeitsanalysen und etwas Handarbeit knacken.

Vigenère – Ciffre

Eine Möglichkeit, die Vorteile der leichten Berechenbarkeit bei der Caesar – Verschlüsselung zu nutzen und trotzdem die Sicherheit zu verbessern, ist eine Erfindung u.a. des französischen Diplomaten Vigenère (16. Jahrhundert). Man nehme ein Schlüsselwort, ordne jedem Buchstaben des zu verschlüsselnden Textes nacheinander einen Buchstaben des Schlüsselwortes zu und verschlüssele diese jeweils mit Caesar. Der Text wird üblicherweise sehr viel länger sein als das Schlüsselwort, daher setzt man es so oft hintereinander, bis es lang genug ist.
Der wesentliche Vorteil ist, dass nun nicht mehr ein einzelner Buchstabe reicht, um den Text zu entschlüsseln. Im Prinzip braucht man das ganze Schlüsselwort.
Leider (oder auch nicht) sind aber auch in diesem Fall ungewünschte Zugriffe möglich. Kennt man die Länge des Schlüsselwortes, so kann man die statistische Analyse jeweils auf die Teilmenge der Buchstaben des Textes anwenden, die in Abständen der Schlüssellänge im Text auftauchen. Für diese verfährt man nun wie vorher für das ganze Wort und man ist fertig.
Wie kommt man aber an die Schlüsselwortlänge ? Eine Möglichkeit dafür bietet der ...

Kasiski – Test

Man suche im codierten Text nach (möglichst häufig) wiederkehrenden gleichen Buchstabengruppen. Zu mehreren von ihnen bestimme man den ggT ihrer Abstände im Text. Mit guter Wahrscheinlichkeit hat man dann die Schlüssellänge oder ein kleines Vielfaches von ihr getroffen. Dazu ein Beispiel aus meinem WIND.TXT :

Der erstaunliche Aspekt ist, dass ich bei der Suche nach solchen Buchstabengruppen auf einen Fehler in meiner ursprünglichen Darstellung gestoßen bin. Ich wollte 50 Zeichen pro Zeile haben, hatte aber nicht bemerkt, dass es in Wahrheit nur 49 waren. Da ich die Schlüssellänge meines Textes kannte, war mir klar, dass die ungewöhnlich lange mehrfach auftretende Buchstabengruppe (≡Zufall unwahrscheinlich !) nicht 49 Zeichen weiter sich wiederholen kann. Nach Korrektur der Zeilenlänge stimmt es: Es sind in Wahrheit 48 Zeichen !

Die Abstände für einige Beispiele (Es gibt noch viele andere) :

WAV	8 ; 96 ; 120
WA	8 ; 96 ; 120 ; 156 ; *126 ; 76* ; 56 ; 16 ; 88 ; 80 ;
VQZQNQAZ	48 ; 184 !!!!
AQA	92 ; 164 ; 276 ; 72

Es zeigt sich, dass es sinnvoll sein wird, mit einer Schlüssellänge von z.B. 8 zu arbeiten. Geht man danach wie beschrieben vor, erhält man problemlos den (allerdings verdoppelten) Schlüssel.

Allgemeine Anmerkungen zu den einfachen Verfahren

Grundsätzlich erscheint es unbefriedigend, in den Codetexten nur mit reinen Großbuchstaben zu arbeiten. Dieser Mangel ist jedoch nicht wesentlich. Entscheidend beim Caesar – und Vigenère – Verfahren ist die Möglichkeit, durch eine einfache Berechnung zur Übersetzung zu gelangen. Hierfür bieten sich natürlich beliebige Erweiterungen des Ausschnittes aus dem Zeichensatz an.
Dabei ist die Unterscheidung von Klein – und Großbuchstaben tatsächlich ein relativ ungünstiges Problem. Sie treten nämlich immer an markanten Stellen im Text auf. Würde man zusätzlich die Leerzeichen berücksichtigen, dann fände man Großbuchstaben praktisch immer nach den Leerzeichen. Alle solche Besonderheiten in den zu codierenden Texten bieten eine hervorragende Möglichkeit für kryptoanalytische Angriffe.
Würde man im Code die Unterscheidung zwischen Klein – und Großbuchstaben weglassen, dann könnte man den Großbuchstaben (ASCII 65 – 90) z.B. einfach die benachbarten Zeichen @ (ASCII 64) [, \ und] (ASCII 91 – 93) hinzufügen[1], um mit ihnen die anderen Zeichen auszudrücken. Beispielsweise:
@D\ER[LE]@M\OND[LE]IST[LE]AUFGGANGEN[PU]@D\IEGOLDNEN[LE]STERNLEIN[LE]PRANGEN ...
Schon in diesem kurzen Abschnitt aber ist zu erkennen, dass man bei einem solchen Verfahren natürlich sehr schnell auf sich wiederholende Buchstabengruppen trifft, die hervorragender Ansatzpunkt für eine Kryptoanalyse sind.
Außerdem kommt hinzu, dass nun bestimmte Zeichengruppen mit hoher Sicherheit an bestimmten Positionen stehen und man gezielt zunächst nach den Übersetzungen dieser Gruppen suchen muss[2].
Ist das Ziel, vorrangig nur eine überhaupt verständliche Information wiederherzustellen und nicht eine exakte Wiederherstellung des ursprünglichen Textzustandes, dann reicht eine Beschränkung auf allein die Großbuchstaben völlig aus.

1 Interessant wäre es, zusätzlich die nächsten beiden Zeichen **^** und **_** (ASCII 94 und 95) zusätzlich mit einzubauen, da man dann einen bei 64 beginnenden 32 Zeichen langen Block von Zeichen verwenden kann.
Die Binärdarstellungen sind 64: **0001 0000** bis 95: **0001 1111**, sodass ggf. einfache Rechenoperation auf Binärbasis genutzt werden können.
Das oben beschriebene Problem wäre damit nicht zu lösen.

2 Der blinde Perfektionismus und Schematismus, zu dem Militärs neigen, hat zusammen mit Unkenntnis der kryptoanalytischen Angriffspunkte oft zu schwerwiegenden Fehlern geführt. Am Beginn von Funksprüchen wurden häufig besondere Kenngruppen gesendet. Da der Gegner diese in der Regel nach einer gewissen Zeit kannte, war es sehr einfach, einen Schlüssel zu knacken, wenn diese Kenngruppen mit demselben Schlüssel mit verschlüsselt wurden.

Tabelle eines codierten Textes :

12345678901234567890123456789012345678901234567890
↓↓

000+ UNUPMESBIUODMVQAZFSWMUWAVNFDCAWAVNEAZIRJPVHNSBQJBR

050+ HNVVFDBFHNSRQJMAVEKUHNPNEAVJLNIHFDMVQAVZDNBRULNNKH

100+ QUUSMEGABVKJAPKKVARYPFHDMADXMEHEORQPTVFDPRLOAGGAZT

150+ DNVVFDBZDNBRULNNKHABQZMEQPWGHICAGEAGGWAMHEKUHJLRVO

200+ BNPIMFKAZELJLVDJMEVEVQVEMRLJAVRQF***VQZQNQAZ***BGAZRLJIC

250+ DYPRKAQAHNENUOQPKJQPKPAVFDMEZEM***VQZQNQAZ***NQVCEHZMAVE

300+ VQDXMEHNJRPQMUWAAVFDLNVCMFSNIRFDQAJWVTCQPNOPMAPAQA

350+ HHQREAVSUACAGAQPKDMVVOMALYPGKAZELJLVDJMEXJLNXYPALY

400+ PGJNWFVAZODAZBGAZFRWMUQHQPKEKUKAQFVASBQNIQXJLVFDJV

450+ QQMOHNPNXLBXHEV***VQZQNQAZ***VFDJVQBQYPOKUDQACLATRUQVQZE

500+ ZQUAPRQDQRUAQAHJQAGEIAHNNVOIPRLJMEJQKXWARNVIQADJCA

550+ GFIFPEVUHEVRUZIAQCCPNPMAVEMOHELRNKVEDZIAXJLQDJVOUQ

600+ MYOPMAVEMNOHMYRODBUHIPKAVFDCBZDHSBQNIQKEMYWOQPKZMA

650+ EWCPKQVQMWXFWAVNFDTHIPANJPUNOSCFVPMGLDZQDOVVFDBIRN

700+ TNXPMEOWKUHJTVHBMALDUGUWMAHJLVHSIAJAVUHNCAWAZHQZDR

750+ UOKUPEMEWAVQLAONQVMFFDUVQGMWDOUVQQVQKAQAHNAPKQMGWA

800+ TGHJLVHGWRSBMFLAENUAVQXNKUGAVYDYPNQBIYOOWNXOARUWBR

850+ PZIFVOQRNAQAHJBBQDMEDQAOUWKUWAVQLALEHEENUAVNPNIAGA

900+ LRVZWEIAANQCMXRIURQQVQDHTRGAVRQOQREAORJJMGHJAPKWCG

950+ HJAVHAVGJAQFWAZGDJMVQARHQCMSUWCGUWBNXBAVHVCHQZNEDC

↑↑
12345678901234567890123456789012345678901234567890

Friedmann - Test

Zunächst das Problem: Kennt man die Schlüsselwortlänge, dann ist das Entschlüsseln nur noch ein einfacher Schritt! Zu einer ziemlich guten Aussage über die Schlüsselwortlänge kommt man mit Hilfe des Friedmann – Tests. Auch wenn die Vigenère – Verschlüsselung nicht der Stand der Technik ist und damit natürlich auch der Friedmann – Test, so sind beide dennoch gut geeignet, die grundlegenden Gedanken statistischer Analyseverfahren zur Entschlüsselung von Texten deutlich zu machen. Beim Friedmanntest geht es darum, dass das Auftreten von Paaren von gleichen Buchstaben in Texten Aussagen über die Art des Textes machen kann.

Aufgabe:
Untersuchen Sie selbst einmal an Hand eines Textes, mit welcher Wahrscheinlichkeit Sie beim willkürlichen Auswählen von zwei Buchstaben zufällig denselben treffen!

Herleitung der Berechnung der Schlüssellänge

Zunächst nur das Prinzip, für einen Text aus einem rein willkürlichen Buchstabensalat ist die Überlegung einfach:

Jeder Buchstabe tritt dann mit der Wahrscheinlichkeit 1/26 auf, Paare – nicht nebeneinander, sondern im Text verstreut, willkürlich gewählt – genau mit diesem Wert, da zu irgendeinem Buchstaben ein beliebiger anderer mit der Wahrscheinlichkeit 1/26 genau der selbe ist. Diese Zahl dezimal ausgedrückt ist 1/26 = 0,0385.

Betrachten man nun dagegen einen „normalen“ deutschen Text, dann gibt es – wie wir schon gesehen haben – häufige Buchstaben wie das ***e*** und weniger häufige wie etwa das ***q***. Obwohl in einem Text der gleichen Länge wie bei dem mit dem Buchstabensalat natürlich immer noch dieselbe Gesamtzahl von Buchstaben enthalten ist, tritt der Buchstabe ***e*** in einem größeren Anteil auf, nämlich mit der Wahrscheinlichkeit n_e / n , dabei bedeuten n_e die Anzahl der ***e***'s im Text und ***n*** die Anzahl <u>aller</u> Buchstaben im Text.

Dazu gibt es nun im Text n_e-1 andere und mit der Wahrscheinlichkeit $\frac{n_e}{n} \cdot \frac{(n_e-1)}{n}$ ein Paar von ***e***'s. Eine entsprechende Formel gibt es für jeden der Buchstaben des Alphabetes. Im Gegensatz zum Buchstabensalat sind diese Werte aber nicht für alle Buchstaben gleich. So wird dieser Wert für ***y*** in einem deutschen Text sicher sehr viel kleiner sein. Addiert man alle diese Werte auf, sollte sich das dann nicht aufheben und insgesamt wieder $\frac{1}{26}$ herauskommen? Die Antwort lautet nein!

Es hebt sich nicht auf, weil die Häufigkeiten der einzelnen Buchstaben quadratisch eingehen und durch das Quadrieren verstärken sich die größeren Werte und wenn man diese Quadrate für alle Buchstaben zusammen zählt, bekommt man eine Zahl heraus, die größer als $26 \cdot (\frac{1}{26})^2 = \frac{1}{26}$ ist!

Für einen normalen deutschen Text erhält man einen Wert von etwa 0,0762, also etwa das doppelte des Wertes beim Buchstabensalat.

Aufgabe:
Berechnen Sie (*eine Näherung für*) diesen Wert mit Hilfe der auf Seite 4 angegebenen Wahrscheinlichkeiten!

Was hat das nun mit der Vigenère – Verschlüsselung zu tun?

Die oben beschriebene Summe

$$\kappa = \frac{\sum_{i=1}^{26} n_i \cdot (n_i - 1)}{n \cdot (n-1)}$$

bezeichnet man als Friedmannschen Koinzidenzindex, wofür Friedmann selbst das Symbol κ [1] verwendete.

Für längere Texte ist der Unterschied zwischen n_i und n_i-1 nicht groß, so dass man diese Werte auch durch die Wahrscheinlichkeiten p_i ersetzen kann. Man verwendet dann also

$$\kappa = \sum_{i=1}^{26} p_i^2$$

Wird nun nicht mit Caesar, sondern mit Vigenère verschlüsselt, dann nähert man sich durch die unterschiedlichen Verschiebungen der verschiedenen Buchstaben des Schlüsselwortes einem Buchstabensalat an. Je länger, desto mehr[2]! An dieser Stelle geht daher die Beziehung zur Schlüsselwortlänge ein!

Wir nehmen einmal an, wir würden die Schlüsselwortlänge ℓ kennen und hätten den Text buchstabenweise in eine Tabelle mit genau ℓ Spalten eingetragen. Dann finden wir in jeder Spalte genau die Codetextbuchstaben, die mit dem selben Schlüsselwortbuchstaben übersetzt wurden. Für jede dieser Spalten selbst muss der Koinzidenzindex dem oben angegebenen Wert für einen normalen deutschen Text von 0,0762 entsprechen. Für Buchstabenpaare aus verschiedenen Spalten wäre bei einem langen Schlüsselwort der Wert aber nahe 0,0385.

Die Anzahl von Buchstabenpaaren im gesamten Text ist:

$$\frac{n \cdot (\frac{n}{l} - 1)}{2} = \frac{n \cdot (n-l)}{2 \cdot l}$$ für dieselbe Spalte,

$$\frac{n \cdot (n - \frac{n}{l})}{2} = \frac{n^2 \cdot (l-1)}{2 \cdot l}$$ für eine andere Spalte und

$$A = \frac{n \cdot (n-l)}{2 \cdot l} \cdot 0{,}0762 + \frac{n^2 \cdot (l-1)}{2 \cdot l} \cdot 0{,}0385$$ ist für beide Fälle zusammen die erwartete Anzahl von Paaren aus gleichen Buchstaben ...

1 Der hier beschriebene Test heißt daher auch manchmal KAPPA – Test.
Wegen der Schwierigkeit, I (großes i) und l (kleines el) im Text unterscheiden zu können, bin ich dabei geblieben.

2 Das erstaunliche dabei ist, das bei einem Schlüsselwort, das nicht selbst ein völliger Buchstabensalat ist, selbst dann noch Abweichungen von dem Wert 1/26 auftreten, wenn das Schlüsselwort so lang wie der Text selbst ist !
Aus diesen Überlegungen lässt sich weiterhin eine allgemein gültige Anforderung an Verschlüsselungsverfahren ableiten: Bei einem sicheren Verfahren sollten die im Text (notwendigerweise) vorhandenen Informationen sich möglichst wenig in den verschlüsselten Daten statistisch nachweisen lassen!

...und damit die entsprechende Wahrscheinlichkeit geteilt durch n*(n-1)/2:

$$\frac{A}{n(n-1)/2} = \frac{(n-l)}{(n-1)\cdot l}\cdot 0{,}0762 + \frac{n\cdot(l-1)}{(n-1)\cdot l}\cdot 0{,}0385$$
$$= \frac{1}{(n-1)\cdot l}\cdot(0{,}0377\cdot n + l(0{,}0385\,n - 0{,}0762))$$

Dieser Wert nun sollte dem Friedmannschen Koinzidenzindex $\boldsymbol{\kappa}$ entsprechen und damit die Möglichkeit ergeben, eine Näherung für ℓ zu errechnen:

$$\boldsymbol{l}\cdot(n-1)\cdot\boldsymbol{K} \approx 0{,}0377\,n + \boldsymbol{l}\cdot(0{,}0385\,n + 0{,}0762)$$
$$\boldsymbol{l}\cdot(n-1)\cdot\boldsymbol{K} - \boldsymbol{l}\cdot(0{,}0385\,n + 0{,}0762) \approx 0{,}0377\,n$$
$$\boldsymbol{l}\cdot((n-1)\cdot\boldsymbol{K} - (0{,}0385\,n + 0{,}0762)) \approx 0{,}0377\,n$$
$$\boldsymbol{l} \approx \frac{0{,}0377\,n}{(n-1)\cdot\boldsymbol{K} - 0{,}0385\,n + 0{,}0762}$$

Alle in dieser Formel nun enthaltenen Teile sind einfach zu bestimmen, so dass eine einfache Möglichkeit besteht, eine Näherung für die Schlüsselwortlänge zu erhalten.