10 Kryptographie

(*cryptography*) auch *Kryptologie* (griech.) = Lehre von den Geheimschriften

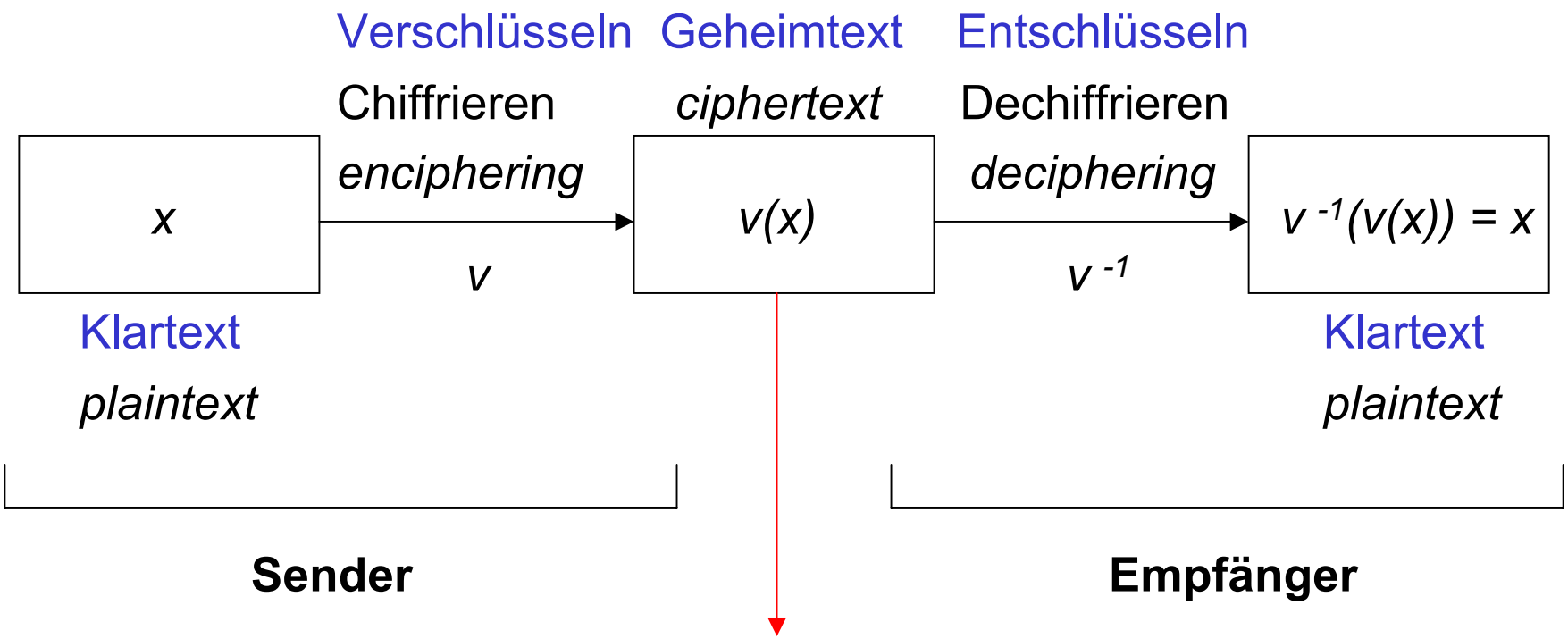
Zweck: *ursprünglich:* vertrauliche Nachrichtenübertragung/speicherung

rechnerbezogen: Vertraulichkeit, Authentizität, Verbindlichkeit

Stellenwert in der Informatik:

- *bei nicht vernetzten Systemen hilfreich* als zusätzliche Sicherung (zusätzlich zum Zugriffsschutz für Paßwort- und andere Dateien)
- *bei vernetzten Systemen unverzichtbar* wegen der Möglichkeit, in den Nachrichtenverkehr einzugreifen

10.1 Grundbegriffe



Anforderungen an v :

- $v : X \to Y$ ist injektiv

- $v(x) = E(K,x)$ $\quad v^{-1}(y) = D(K,y)$ mit Schlüssel K oder

  $v(x) = E(K_E,x)$ $\quad v^{-1}(y) = D(K_D,y)$ mit Schlüsselpaar K_E , K_D

  Zweck: v kann *leicht gewechselt* werden, indem nicht E und D, sondern nur K bzw. K_E , K_D gewechselt werden

  (E,D) heißt Verschlüsselungsverfahren *(cryptosystem, cipher)* –
  symmetrisches Verfahren mit K,
  asymmetrisches Verfahren mit K_E , K_D

  Prinzip: Sicherheit gegen Angriffe wird durch Geheimhaltung des Schlüssels – nicht des Verfahrens – erreicht !

Code knacken durch **Kryptoanalyse**:

Ziel: **Schlüssel** und **Klartext** herausfinden

Ansätze:

Entschlüsselungsangriff – wenn nur Geheimtext vorliegt

Klartextangriff – wenn zusätzlich Teile des Klartextes vorliegen (z.B. „login:“) oder wenn sogar E und K_E bekannt

Notwendige Voraussetzung:

Sprache der Nachricht muß bekannt sein !

Notwendige Voraussetzung für sichere Verschlüsselung:

Durchprobieren der Schlüssel muß aussichtslos sein

Beispiel: Klartextangriff mit Spezialrechner bei bekanntem symmetrischen Verfahren, 10^{10} Schlüssel pro Sekunde

Schlüsselgröße	*benötigte Zeit*	*Qualität*
40 Bits	100 Sekunden	schlecht
56 Bits	10 Tage	schwach
64 Bits	30 Jahre	mäßig
128 Bits	10^{20} Jahre	gut
256 Bits	10^{60} Jahre	sehr gut

10.2 Transpositionsverschlüsselung

Seien *A* bzw. *B* die **Alphabete** für Klartext bzw. Geheimtext.

Klassifikation symmetrischer Verschlüsselungsverfahren:

- **Transpositionsverfahren** $v : A^* \rightarrow A^*$

  Nachrichtenteile werden *umgestellt (permutiert)*

- **Substitutionsverfahren** $v : A^* \rightarrow B^*$

  Nachrichtenteile werden *ersetzt*

- **kombinierte Transposition/Substitution**

Transpositionsverschlüsselung allgemein:

$v(x_1\ x_2\ \ldots\ x_N) = x_{p(1)}\ x_{p(2)}\ \ldots\ x_{p(N)}$ (x_i = Buchstaben(-gruppen))

mit p = Permutation der Indizes $1,2,\ldots,N$

Beispiel 1: Permutation mit fester Periode d = 4

D	E	R	S	C	H	A	T	Z	I	S	T
R	E	S	D	A	H	T	C	S	I	T	Z

Schlüssel: 3 2 4 1

oder z.B. „Schlüsselwort“ ROSA *

* soll heißen „Jedes Buchstabe symbolisiert die Positionsnummer des Buchstabens bei Anordnung der Buchstaben in alphabetischer Reihenfolge“

oder: „Dechiffrieren wie alphabetisches Anordnen von ROSA“

Beispiel 2: Rechteck-Raster mit Kantenlänge 4

```
D E R S
C H A T
Z I S T
```

Schlüssel: 1 2 3 4

D Z C E H I R A S S T T

Beispiel 3: Zickzack-Raster

```
D   C   Z
 E S H T I T
  R   A   S
```

Schlüssel: 2 3 1

E S H T I T R A S D C Z

Kryptoanalyse:

Hier wie auch bei anderen Verschlüsselungsverfahren knackt der Angreifer den Code
mit Kenntnissen über die verwendete Sprache

① Verfahren herausfinden, sofern noch nicht bekannt:

- wenn *Buchstaben-Häufigkeiten* den Buchstaben-Häufigkeiten der Sprache entsprechen, liegt Transposition vor;

- wenn Häufigkeiten der *Digramme* (benachbarte Buchstaben) *nicht* den Digramm-Häufigkeiten der Sprache entsprechen, liegt Transposition einzelner Buchstaben vor;

- ... usw.

② Schlüssel und Klartext herausfinden, z.B. für Beispiel 1:

- *Präfixe* zunehmender Länge betrachten und
- deren **Anagramme** bilden,
- dabei die Häufigkeiten der jeweiligen *Digramme* betrachten,
- eventuell auch *Trigramme* und *Wörter* erkennen.

Beispiel 1 wird damit *sehr schnell* geknackt!

(Schlüsselgröße ist gering: für n Elemente gibt es n! Permutationen;
daher gibt es bei Beschränkung auf d=4 nur 4! = 24 mögliche Schlüssel !)

10.3 Substitutionsverschlüsselung

$$v(x_1\, x_2 \ldots x_N) = f_1(x_1)\, f_2(x_2) \ldots f_N(x_N)$$

mit $f_i(x_i)$ A oder $f_i(x_i)$ B,

alle f_i injektiv

10.3.1 Monoalphabetische Substitution

$$v(x_1\, x_2 \ldots x_N) = f(x_1)\, f(x_2) \ldots f(x_N)$$

mit einheitlichem f

Anzahl der verschiedenen Möglichkeiten für die Zuordnung der $f(x_i)$ zu den x_i

= Anzahl der Permutationen der x_i : $n!$ (mit $n = |A|$, z.B. $n = 26$)

→ $n!$ verschiedene Schlüssel, $n! \approx (n/e)^n \quad (2\pi n)$ (Stirlingsche Formel)

Schlüsselgröße ist $\log_2 26! \approx 88.4$

Demnach Qualität „mäßig bis gut“ ?

Schlüsselgröße ja, Verfahren nein ! (s.u.)

Beispiel 1: $f(x) = (x+k) \bmod n$ (bei Identifizierung von a,b,.. mit 0,1,..),

d.h. Einschränkung auf 26 mögliche Schlüssel k,

„verschobenes Alphabet“

D E R S C H A T Z I S T

Schlüssel k = 3

G H U V F K D W C L V W

„Cäsars Verschlüsselung“

Entschlüsselung mit $f^{-1}(y) = (y-k) \bmod n$,

denn $f^{-1}(f(x)) = f^{-1}((x+k) \bmod n)$

$= ((x+k) \bmod n - k) \bmod n$

$= ((x+k) \bmod n - k \bmod n) \bmod n$

$= ((x + k - k) \bmod n) \bmod n$

$= x \bmod n$

Beispiel 2: Schlüssel = Tabelle der f(a), f(b), . . , f(z) ➔ 26! Schlüssel

evtl. mit Schlüsselwort (➔ *kleinerer* Schlüsselraum!), z.B.

A B C D E F G H I J K L M N . . .

K O N S T A I P E L B C D F . . .

Schlüsselwort ist „Konstantinopel“

Entschlüsselung durch Umkehrung der Tabelle.

Beispiel 4: $f(x) = k\,x \mod n$ *„multipliziertes Alphabet“*

(Schlüsselraum nicht größer als bei Beispiel 1!)

Beispiel 5: $f(x) = (k_0 + k_1\,x) \mod n$ *affine Transformation*

Anzahl der möglichen Schlüssel (k_0, k_1) : 26*26

Kuriosität: für $f(x) = (k - x) \mod n$ gilt $f^{-1} = f$!

(Beweis: Übung)

Verallgemeinerung: Polynomielle Transformation

Achtung :

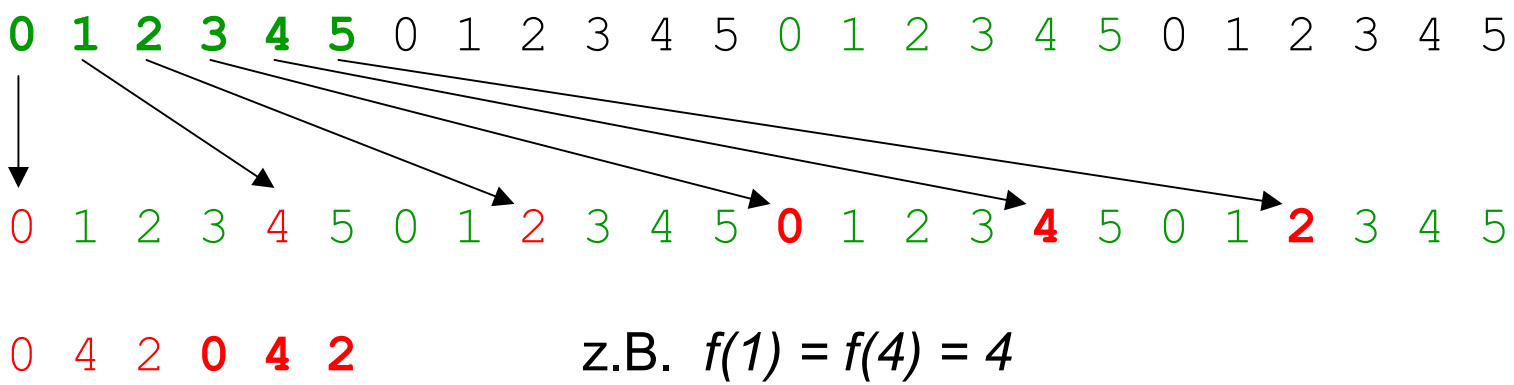
Wie kann die *Injektivität von f* garantiert werden?

Wie wird entschlüsselt?

10.3.1.1 Injektivität bei affiner Transformation

Beispiel für *nicht umkehrbare* Verschlüsselung:

$f(x) = 4x \mod 6$:



z.B. $f(1) = f(4) = 4$

? Wie kann ausgeschlossen werden, daß es x,y gibt derart, daß

zwar $x \neq y$ gilt, aber dennoch $k\,x \mod n = k\,y \mod n$

Satz 1: Wenn k und n *teilerfremd* sind, d.h. $ggT(k,n) = 1$, dann gilt für alle x,y mit $0 \le x < y < n$

$$k\,x \mod n \neq k\,y \mod n$$

Bemerkung: Das bedeutet, daß die $k\,i \mod n$, $i = 0,1,..,n-1$, eine Permutation der i sind !
Damit ist $f(x) = k\,x \mod n$ injektiv.

Beweis: (durch Widerspruch)

Wäre das nicht so, gäbe es x,y mit $k(y-x) \mod n = 0$, also $k(y-x) = a\,n$ für ein gewisses a .

Wegen $ggT(k,n) = 1$ muß n das $y-x$ teilen. Das geht aber nicht, weil $y-x < n$. □

Satz 2: Wenn k und n teilerfremd sind, dann hat

$$k\,x \mod n = a \qquad (0 \le a < n)$$

eine eindeutige Lösung mit $0 \le x < n$.

Beweis: folgt direkt aus Satz 1.

Satz 3: Wenn k und n teilerfremd sind,
ist die affine Transformation injektiv.

Beweis: wie Satz 1 .

Bemerkung: Wenn n eine *Primzahl* ist, sind k und n
garantiert teilerfremd ! (Leider ist 26 nicht prim.)

Satz 4: Wenn K_E und n teilerfremd sind, wird ein mit

$$f(x) = K_E\, x \mod n$$ verschlüsselter Text

mit

$$f^{-1}(y) = K_D\, y \mod n$$ entschlüsselt,

wobei

$$K_E\, K_D \mod n = 1$$ (d.h. K_D ist *Inverse* von K_E, *existiert* wegen Satz 2 !).

Beweis: Zu zeigen ist $f^{-1}(f(x)) = x$.

$$\begin{aligned} & K_D\,(K_E\, x \mod n) \mod n \\ = {} & (K_D \mod n)\,(K_E\, x \mod n) \mod n \\ = {} & K_D\, K_E\, x \mod n \\ = {} & (x \mod n)\underbrace{(K_D\, K_E \mod n)}_{=1} \mod n \\ = {} & x \end{aligned}$$

□

Satz 4 ist Spezialfall von

Satz 5: Wenn K_E und n teilerfremd sind, wird ein mit

$$f(x) = (k + K_E\, x) \bmod n$$ verschlüsselter Text

mit

$$f^{-1}(y) = K_D\,(y-k) \bmod n$$ entschlüsselt,

wobei

$$K_E\, K_D \bmod n = 1$$

Beweis: $f^{-1}(f(x)) = x$ zeigen wie bei Satz 4 .

10.3.1.2 Kryptoanalyse

durch Berücksichtigung von Mono/Di/Trigramm-Häufigkeiten

Genauere Kenntnis des Verfahrens erleichtert das Knacken des Codes, z.B.

Voraussetzung: Englischer Klartext sei mit *affiner Transformation* verschlüsselt

Englische Buchstaben nach abfallenden Häufigkeiten:

E T O A N I R S H . . .

Geheimtext: `GFKVLCLFKEKEEMCECREGKKLEMHKVLLFCSYFL`

Häufigkeiten:

K	L	E	F	C	.	.	.
6	6	6	4	4	.	.	.

→ *These:* `K` steht für E, `L` steht für T

Überprüfung der These:

$K = (k_0 + k_1 E) \bmod n$ d.h. $10 = (k_0 + 4\, k_1) \bmod 26$

$L = (k_0 + k_1 T) \bmod n$ d.h. $11 = (k_0 + 19\, k_1) \bmod 26$

Subtraktion der Gleichungen liefert

$1 = 15\, k_1 \bmod 26$, also $k_1 = 7$,

und damit $10 = (k_0 + 28) \bmod 26$, also $k_0 = 8$.

Inverse von 7 ist 15 – also zur Probe dechiffrieren mit

$$f^{-1}(y) = 15\,(y - 8) \bmod 26$$

und prüfen, ob sich vernünftiger Text ergibt.

Beispiel mit allgemeiner monoalphabetischer Substitution:

Geheime Anleitung zur Auffindung eines Schatzes
aus „Der Goldkäfer“ von E. A. Poe.

Sprache: Englisch

Englische Buchstaben nach abfallenden Häufigkeiten wie oben:

E T O A N I R S H . . .

Geheimtext:

5 3 ≠ ≠ 3 0 5)) 6 * ; 4 8 2 6) 4 ≠ .) 4 ≠) ; 8 0 6 *
T H E H H T E

; 4 8 8 / 6 0)) 8 5 ; 1 ≠ (; : ≠ * 8 8 3 (8 8) 5 *
T H E E E T T E E E E

; 4 6 (; 8 8 * 9 6 * ? ; 8) * ≠ (; 4 8 5) ; 5 * 2 :
T H T E E T E T H E T

* ≠ (; 4 9 5 6 * 2 (5 * - 4) 8 / 8 * ; 4 0 6 9 2 8 5) ;
T H H E E T H E T

) 6 8) 4 ≠ ≠ ; 1 (≠ 9 ; 4 8 0 8 1 ; 8 : 8 ≠ 1 ; 4 8
E H T T H E E T E E T H E

8 5 ; 4) 4 8 5 5 2 8 8 0 6 * 8 1 (≠ 9 ; 4 8 ; (8 8 ; 4
E T H H E E E E T H E T E E T H

(≠ ? 3 4 ; 4 8) 4 ≠ ; 1 6 1 ; : 1 8 8 ; ≠ ? ;
H T H E H T T E E T T

? E T O A N I R S H . . . 8 ; 4 ≠) * 5 6 (. . .

Typische Vorgehensweise:

1. These: `8` ist E.
   Wird gestützt durch einige EE.

2. These: `;` ist T.
   Wird gestützt durch Trigramm THE. Also

3. These: `4` ist H

4. These: `≠` ist A? Scheidet aus wegen `≠≠` (am Anfang).

5. These: `≠` ist O.

   `)` kann weder A noch N sein – wegen `))` und `4)4` – auch nicht I oder R.

6. These: `)` ist S.

..... usw.

10.3.2 Sicherheit von Verschlüsselungsverfahren

= Resistenz gegenüber Angreifern

Monogramme, Digramme, Trigramme, ... in Sprachen haben *unterschiedliche Häufigkeiten*, die zudem *sprachabhängig* sind !

Wenn man in einem deutschen Text das Fragment „Hafe“ findet, wird „n“ oder auch „r“ mit sehr viel größerer Wahrscheinlichkeit folgen als etwa „e“ oder „z“.

Die Buchstabenfolge „caxbfod“ gibt es garantiert nicht.

„die“ ist häufiger als „bla“.

usw. usf.

Angreifer können dies ausnutzen.

Angreifer können *nicht* eine verschlüsselte *Zufallszahl* herausfinden!

10.3.2.1 Redundanz natürlicher Sprache

Def.: **Absolute Rate** einer Sprache: $R = log_2\, n$

= Bitanzahl für die Codierung eines Zeichens

z.B. Deutsch/Englisch/... : $R \approx 4{,}7$

Aber nicht alle Buchstaben müssen mit der gleichen Bitanzahl codiert werden – für die häufigeren sollte man weniger Bits verwenden.

Beispiel Morsealphabet:

E	T	A	. . . .
•	–	• –	

Erwartete Bitanzahl eines Morsezeichens X in englischem Text ist

$$\sum p_i\, b_i \approx 2{,}5 < R \quad !$$

p_i: Monogramm-Häufigkeiten

Informationsgehalt eines Buchstabens X – gemäß Häufigkeiten:

$$H(X) \approx 1{,}5 < 2{,}5 < R \quad !$$

... und wenn noch der Kontext mitberücksichtigt wird (Digramme, Trigramme, ...):

Mittlere Entropie eines Buchstabens in Text X_N der Länge N:

$$r_N = H(X_N) / N$$

$$1{,}5 \approx r_1 > r_2 > \ldots\ldots$$

Def.: **Rate** einer Sprache: r

= mittlere Entropie eines Buchstabens „*in langem Text*“

Redundanz einer Sprache: $D = R - r$

Z.B. Englisch: $1{,}0 < r < 1{,}5$

$D > 3{,}2$

Merke:

D Rdndnz ntrlchr Sprchn st zmlch grß !

Sinnvolle versus sinnlose Buchstabenfolgen:

	mögliche Buchstaben	mögliche N-Folgen
Buchstabenfolge	$2^R = 26$	2^{RN}
sinnvoller *Text*	$2^r < 3$	2^{rN}

Anteil der N-*Texte* an allen N-*Folgen*:

$$(2^r / 2^R)^N = 1 / 2^{DN} < 1 / 8^N \quad !$$

10.3.2.2 Perfekte Verschlüsselung *(perfect cipher)*

ist dann gegeben, wenn gilt:

Hinter jedem Geheimtext kann *jeder gültige* Klartext stecken.

Notwendige Bedingung dafür ist,

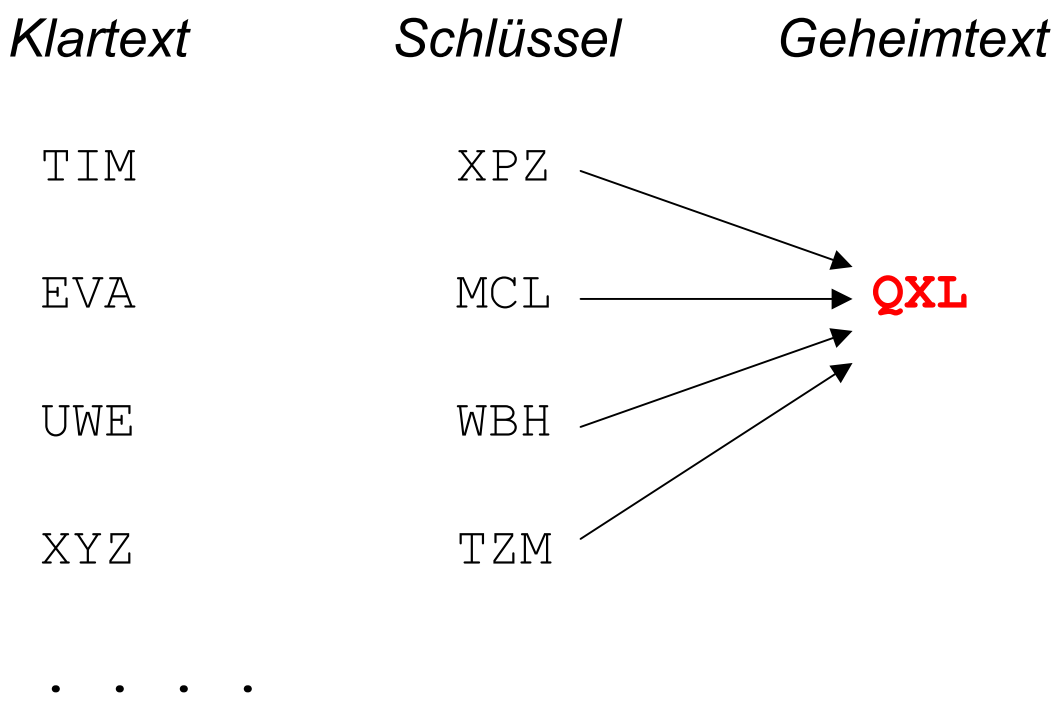
daß es mindestens so viele Schlüssel wie gültige Klartexte gibt
(und alle mit gleicher Wahrscheinlichkeit gewählt werden!).

Hinreichende Bedingung ist die Verwendung eines

Abreißblocks *(one-time pad)* von Zufallswerten,
d.i. eine unendliche Folge, mit deren Werten die *einzelnen*
Klartextzeichen *fortlaufend* verschlüsselt werden.
(„Polyalphabetische Substitution“, ➔10.3.3)

Beispiel: Abreißblock von Buchstaben (bzw. Zahlen von 0 bis 25)

Verschlüsselung: Verschobenes Alphabet



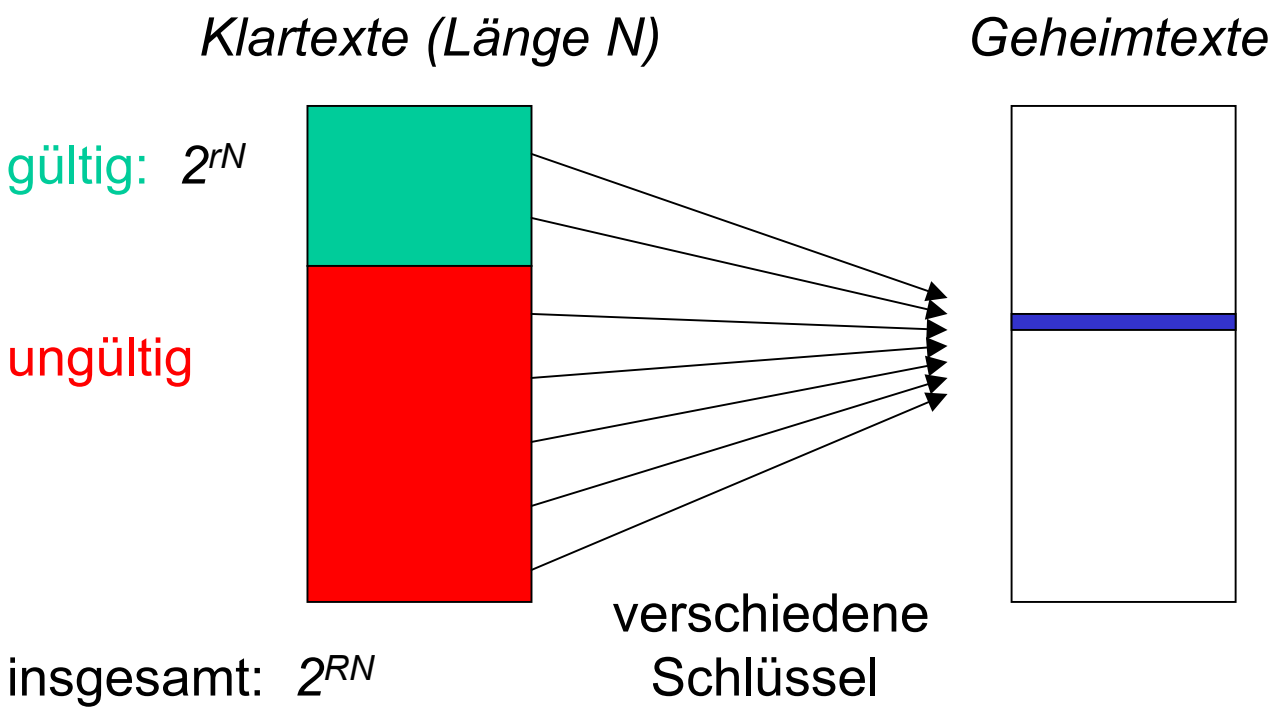
Durchsuchung des Schlüsselraums macht hier keinen Sinn!

10.3.2.3 Sichere Verschlüsselung *(unconditionally secure cipher)*

liegt dann vor, wenn gilt:

Hinter jedem Geheimtext können *viele gültige* Klartexte stecken.

Folgendes muß vermieden werden:



Anzahl der gültigen Klartexte hinter gegebenem Geheimtext: k

Anzahl der möglichen Schlüssel: $s = 2^{H(K)}$

Anzahl der gültigen Texte: $g = 2^{rN}$

Anzahl der Texte insgesamt: $t = 2^{RN}$

Es gilt $k = (g/t)\, s = 2^{rN-RN}\, 2^{H(K)} = 2^{H(K)-DN}$

– und somit sollte $H(K) - D\,N$ nicht in die Nähe von 0 kommen!

Das N, das diese Differenz zu 0 macht, heißt **Unizitätsdistanz** $U = H(K)/D$
und ist ein Maß dafür, ab welcher Textlänge das Verfahren als unsicher gelten muß.

Beachte: Bei $D=0$ ist jedes Verfahren sicher!

Beispiele: $D \approx 3{,}2$

① Verschobenes Alphabet: $H(K) = \log 26 \approx 4{,}7$

$U \approx 4{,}7 / 3{,}2 \approx 1{,}5$ (naja ...)

② Allgemeine monoalphabetische Substitution:

$H(K) = \log 26! \approx 88{,}4$ (10.3.1)

$U \approx 88{,}4 / 3{,}2 \approx 27{,}6$

③ Transposition mit Periode 26: desgl.

Merke: Neben der Redundanz der Sprache ist die Größe des Schlüsselraums entscheidend, nicht das Verfahren!

10.3.2.4 Praktisch sichere Verschlüsselung
(computationally secure cipher)

liegt dann vor, wenn gilt:

Die Kryptoanalyse erfordert exponentiellen Aufwand

in Abhängigkeit von der Schlüsselgröße.

Dies ist bei den bisher betrachteten symmetrischen Verfahren trivialerweise der Fall, muß aber bei anderen Verfahren (→10.5) gesondert gesichert werden.

Merke: Hier geht es um das *Verfahren*, die Schlüsselraum muß dabei natürlich hinreichend groß gewählt werden.

10.3.3 Polyalphabetische Substitution

! Verräterische Buchstaben-Häufigkeiten verbergen ?

Schlüssel von Zeichen zu Zeichen wechseln:

$$y_i = f_i(x_i) \quad \text{mit} \quad f_{i+d} = f_i \,, \quad \text{d.h. Periode } d$$

① Beispiel **Vigenère**-Verschlüsselung: $f_i(x) = (x+k_i) \bmod n$,

z.B. mit Schlüsselwort `CAFE` (also mit Periode 4):

Klartext: `DERS CHAT ZIST`

Schlüssel: `CAFE CAFE CAFE`

Geheimtext: `FEWW EHFX BIXX`

② Beispiel **Beaufort**-Verschlüsselung: $f_i(x) = (k_i - x) \bmod n$

Entschlüsselung genauso! (10.3.1)

③ Beispiel **Alberti**-Verschlüsselung: $f_i(x) = \pi((x+i) \bmod n)$

mit fester Permutation π,
unter Verwendung der *Alberti-Scheibe*, auch für Entschlüsselung

In beiden Fällen ist die Periode $d = n$.

Kryptoanalyse: Zunächst die Periode d herausfinden!

10.3.3.1 Kleine Perioden

Sprache hat Monogramm-Häufigkeiten .....

Geheimsprache habe Monogramm-Häufigkeiten p_i ($\sum p_i = 1$)

Beobachtung: je geringer die Varianz der p_i, desto größer ist d.

Varianz $v = 1/(n-1) \sum (p_i - 1/n)^2$

Rauhheitsmaß $\rho = \sum (p_i - 1/n)^2$ ($= (n-1)v$)

$= \sum p_i^2 - 1/n$

= Wahrscheinlichkeit, daß 2 zufällig ausgewählte Zeichen in Geheimtext *gleich sind*

In vorgelegtem Geheimtext der Länge N komme das Zeichen i n_i-mal vor.

Die Wahrscheinlichkeit, daß in diesem Text 2 zufällig ausgewählte Zeichen *gleich i sind*, ist

$$\frac{½\, n_i\, (n_i - 1)}{½\, N\, (N - 1)}$$

Die Wahrscheinlichkeit, daß in diesem Text 2 zufällig ausgewählte Zeichen *gleich sind*, ist

$$\kappa = \left(\sum n_i\, (n_i - 1) \right) / \left(N\, (N - 1) \right)$$ **Koinzidenzindex**

Nach Definition wird $\kappa \approx \rho + 0{,}038$ erwartet.

ρ liegt zwischen *0* (bei Gleichverteilung $p_i = 1/n$, $d = \infty$)

und *0,030* (bei monoalphab. Substitution, $d = 1$,
weil p_i wie bei Klartext,also
ρ = 0,068 – 0,038)

Also variiert κ zwischen *0,068* für $d = 1$ und *0,038* für $d = \infty$

Graph von κ als Funktion von d experimentell ermittelbar

→ Ermittlung der Periode *d*:

1. ermittle die absoluten Häufigkeiten n_i der Geheimzeichen;
2. berechne damit κ ;
3. *vermute* daraus *d* .

Testen der These „Periode ist d“:

Geheimtext sei $c_1\, c_2 \ldots$

Dann muß sich für jeden der „Texte“ $c_j\, c_{j+d}\, c_{j+2d} \ldots$, $j=1,2,\ldots$,
ergeben: $\kappa = 0{,}068$.

(So kann man *d* auch durch Probieren bestimmen – mit Rechner!)

Andere Methode (nach **Kasiski**):

Beobachtung: Wiederholung eines Trigramms (n-Gramms) im Klartext spiegelt sich in der Regel *nicht* im Geheimtext wieder

(idealerweise sollte wegen der Glättung der Häufigkeiten jedes Trigramm gleich häufig sein – mit $p_i = 1 / 26^3 \approx 0{,}00006$)

– es sei denn, eine Wiederholung im Klartext sei mit einer Wiederholung des Schlüssels zusammengefallen:

Klartext	T O B E O R N O T T O B E . . .
Schlüssel	H A M H A M H A M H A M . . . .
Geheimtext	A O N L O D U O F A O N L . . .

➔ *These: Periode ist 9 oder 3 oder 1 (Teiler von 9)*

10.3.2.2 Große Perioden

durch Verallgemeinerung der *Alberti-Scheibe (d=26):*

mehrere hintereinandergeschaltete **Rotoren**

❶ **Enigma**-Maschine von Hebern/Koch/*Scherbius*/Korn (1920-1940),
von Deutschland im 2. Weltkrieg eingesetzt,
von England (*Turing* u.a.) geknackt

4 Rotoren – wie Zählwerk arbeitend – mit jeweils

$$F_k(x) = (\pi_j((x - k) \bmod n) + k) \bmod n, \quad j=1,2,3,4$$

→ $d = 26^4 \approx 400\,000$

❷ **Hagelin**-Maschine von *Hagelin* (USA 1930)

- arbeitet mit 6 Stiftscheiben mit *unterschiedlich* vielen Stiften, deren aktuelle Stellung einen Schlüssel aus $0,1,2,...,2^6-1$ bestimmt – damit Beaufort-Verschlüsselung;

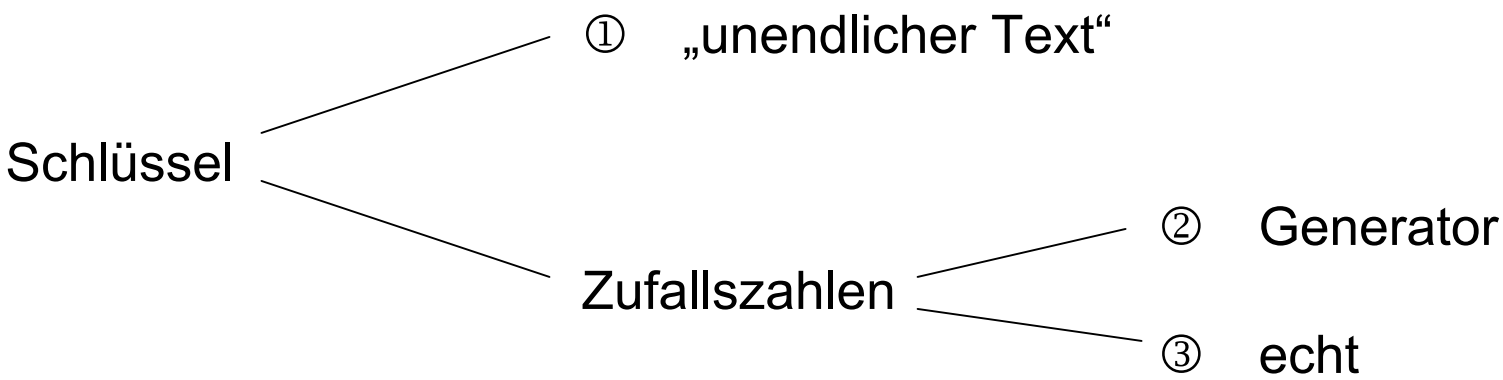
- für jedes Zeichen werden *alle* Scheiben um eine Position weitergedreht.

Stiftanzahlen müssen *teilerfremd* sein, damit maximale Periode erzielt wird, z.B.

$s_k = 17, 19, 21, 23, 25, 26$

→ $d = \prod s_k \approx 100\,000\,000$

10.3.2.3 Nichtperiodische Substitution



① *„unendlicher Text“* (gemeint ist gültiger Text, z.B. dickes Buch)

ist *nicht sicher*, weil der Schlüssel Text-Eigenschaften hat (!)

Angreifer ermittelt gleichzeitig Klartext und Schlüsseltext wie folgt (*Friedman* 1918):

Paarungen (Klartextbuchstabe, Schlüsselbuchstabe)
häufiger Buchstaben sind häufig:

1. Die häufigsten Buchstaben ETOANIRSH machen 70% aller Buchstaben in Texten aus.

2. Häufigkeit von Paarungen häufiger Buchstaben: $0{,}7^2 = 0{,}49$, d.h. ungefähr 50% der Geheimzeichen entstehen aus solchen Paaren.

Beispiel:

Klartext	T H E T R E A S U R E I S . . .
Schlüsseltext	T H E S E C O N D C I P H E R . .
Geheimtext	M O I L V G O F . . . . .

These: M ist ein solches Geheimzeichen
– wie könnte es entstanden sein?

Mögliche Paarungen:

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

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

Weitere Thesen: O und I ebenfalls . . .

Somit wären für M O I folgende Paarungen möglich:

M	O	I
E I T	A O H	A I E R
I E T	O A H	I A E R

Und somit könnte M O I entstanden sein aus

EAA EAI . . . THE . . . THR
IOI IOA . . . THE . . . THR

➔ *These:* dies ist richtig ...

② *Zufallszahlengenerator:*

Probleme:
- periodisch (zwar große Periode ...);
- eventuell bekannte Struktur, die ausgenutzt werden kann.

③ *Echte Zufallszahlen:*

Die Kommunikationspartner brauchen eine gemeinsames „Zufallsbuch“ („unendlich“ groß)

- das ist der *Abreißblock (one-time pad)* aus 10.3.2.2

Vernam-Verschlüsselung (*Vernam*, USA 1917):

- Klartext dual codiert
- verschlüsselt mit Bitfolge
- mittels XOR (Chiffrieren = Dechiffrieren!)