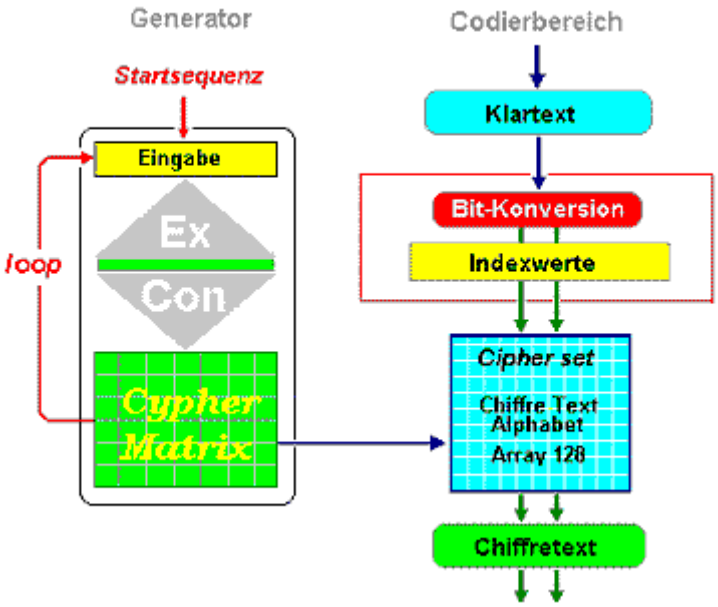
Der Datengenerator

(Ernst Erich Schnoor)

Im CypherMatrix Verfahren – erläutert im Artikel: "Grundlagen des CypherMatrix Verfahrens" – ist die Verschlüsselung einer Information relativ einfach [#1]:

> Ein **Generator** erzeugt das erforderliche **System-Alphabet** und
> im **Codierbereich** wird die Verschlüsselung durchgeführt.



Beide Bereiche werden kombiniert, können aber auch getrennt verwendet werden. Aufgabe des Generators ist die zur Verschlüsselung erforderlichen Steuerungsparameter bereitzustellen. Die eigentliche Verschlüsselung findet im Codierbereich statt.

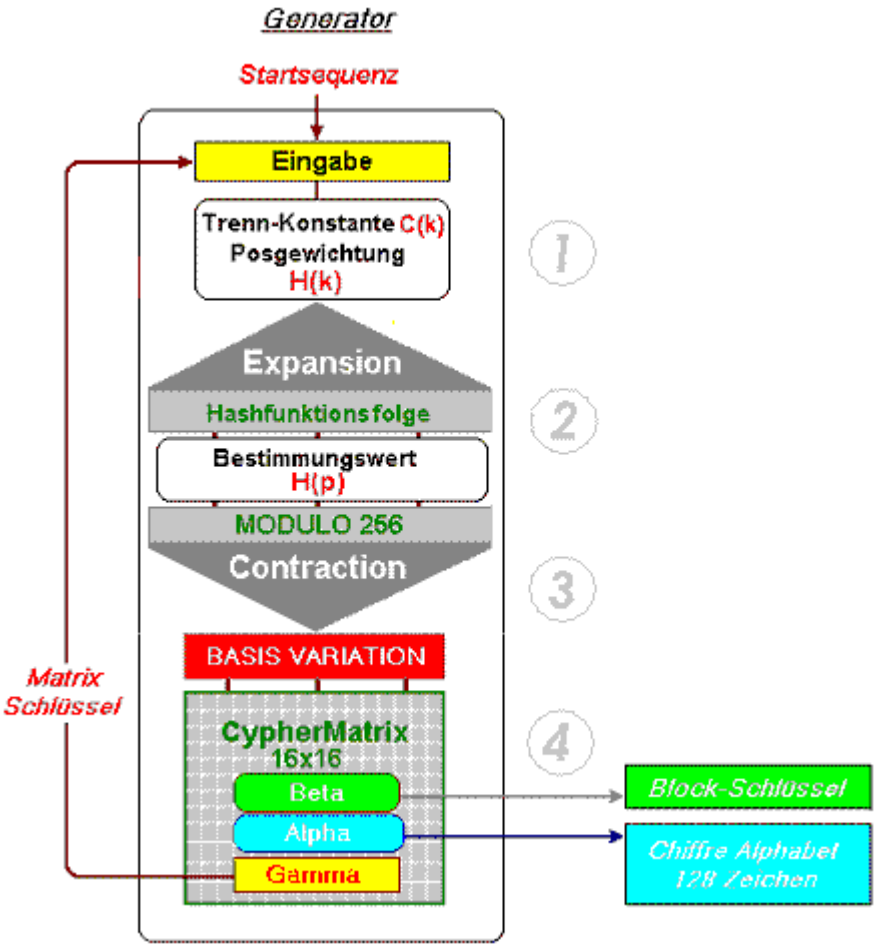
Der Generator

Eine beliebige **Startsequenz** (Passphrase) mit mindestens 36 Zeichen (optimal 42) steuert das gesamte Verfahren. Einige Beispiele:

Ein Fliegenpilz steigt in die Stratosphäre	[42 Bytes]
Die weiße Elster fließt in das schwarze Meer	[44 Bytes]
Leonardo erobert Florenz mit Schneekanonen	[42 Bytes]
7 kangaroos jumping along the Times Square	[43 Bytes]

Die Startsequenz sollte ungewöhnlich sein und dennoch leicht zu behalten, so dass sie nicht aufgeschrieben werden muss aber auch nicht geraten werden kann. Wegen ihrer Länge kann sie weder durch Iteration noch durch Wörterbuchangriffe analysiert werden. Ein Angreifer kann auch nicht mit Erfolg versuchen, Teile des Schlüssels getrennt oder nacheinander zu brechen, da die Startsequenz nur in einem Durchgang als Ganzes gefunden werden kann, wenn überhaupt.

Sowohl beim Sender als auch beim Empfänger erzeugt die Startsequenz einen identischen Ablauf des Verfahrens und inhaltsgleiche Steuerungsparameter.



In jedem Durchlauf liefert der Generator die zur Verschlüsselung erforderlichen Runden-Parameter:

1. Das Chiffre-Alphabet (System-Alphabet) für die aktuelle Runde,
2. einen Blockschlüssel für die XOR-Verknüpfung und
3. den Matrix-Schlüssel als Startsequenz für die nächste Runde.

Startsequenz

Für die Analyse der Daten wird folgende Startsequenz gewählt:

7 Nordlichter wandern über den Großen Belt [n=42]

37 20 4E 6F 72 64 6C 69 63 68 74 65 72 20 77 61 6E 64 65 72 6E
20 81 62 65 72 20 64 65 6E 20 47 72 6F E1 65 6E 20 42 65 6C 74

Es gilt eine eindeutige Bestimmungsbasis für die Analyse der Eingabe zu finden.

Die Eingabe **m** ist eine Folge bestimmter Zeichen **a**(**i**) mit der Länge **n**. Um die Folge zu analysieren, muss sie systematisiert (skaliert) werden. Dazu wird jedem Zeichen **a**(**i**) ein Index zugeordnet und alle **n** Zeichen werden in sachgerechter Weise miteinander verknüpft (Addition):

$$\mathbf{m} = a_1 + a_2 + a_3 + \ldots a_i + \ldots a_n$$

(Der einzelne Wert für "a_i" wird um (+1) erhöht da sonst ASCII-null (0) nicht berücksichtigt wird)

$$\mathbf{m} = \sum_{i=1}^{n} (a_i + 1)$$

$$\mathbf{m} = 4066$$

Um die einzelnen Zeichen **a**(**i**) innerhalb der Folge zu unterscheiden, müssen weitere Merkmale hinzukommen, insbesondere: Positionsgewichtung und Kollisionsfreiheit.

Positionsgewichtung

Mit Besinnung auf **Renè Descartes** (1596 - 1650) wissen wir, dass jeder Sachverhalt - soweit er in seinen Dimensionen skalierbar ist – eindeutig durch seine Koordinaten für **Gegenstand**, **Ort** und **Zeit** bestimmt wird (kartesisches Koordinatensystem).

Um eine eindeutige Bestimmung für jedes Zeichen zu erhalten, verknüpfen wir **a(i) + 1 Gegenstand, p(i) Ort** und die **Zeit t(i)** durch Multiplikation ihrer Dimensionswerte und addieren jedes Ergebnis zum Bestimmungswert **H(k)**. Die Zeit ist nicht relevant, daher: **t = 1**

$$\mathbf{H(k)} = \sum_{i=1}^{n} (a_i + 1) * p_i * t_i \qquad t_i = 1$$

$$\mathbf{H(k)} = 89247$$

Jedes Zeichen **a(i)+1** wird mit seiner Position **p(i)** multipliziert, d.h. **positionsgewichtet.**

Kollisionsfreiheit

Aber **Kollisionen** als Folge eines Austausches von Zeichen innerhalb der Information sind noch nicht ausgeschlossen. Um das zu erreichen, erhöhen wir die Werte der Zeichen in einen Bereich oberhalb der Länge **n** der Zeichenfolge. Der Ort **p(i)** wird mit der Trenn-konstanten **C(k)** erweitert. Die Ableitung der Trennkonstanten **C(k)** wird im Artikel "Bestimmungsfaktoren für Kollisionsfreiheit" dargelegt.

$$\mathbf{C(k)} = \mathbf{n * (n - 2) + code}$$

$$\mathbf{C(k)} = 1681$$

Mit Code – eine gewählte Zahl zwischen 1 und 99 - wird die Funktion individualisiert. Wir setzen: **code = 1**

Mit Einbindung der „Konstanten“ **C(k)** wird der Bestim-mungswert **H(k)** wie folgt errechnet (r = Runde):

$$H(k) = \sum_{i=1}^{n} (a_i + 1) * (p_i + C_k + r)$$

$$H(k) = 6928259$$

Der errechnete Bestimmungswert **H(k)** mit einer Spanne von etwa 10^7 = 10000000 Möglichkeiten vermeidet zwar Kollisionen, erscheint aber für eine eindeutige Abbildung noch zu niedrig zu sein.

Expansion

Zur Erweiterung der Bestimmungsbasis wird die **Hashfunktionsfolge** (**HF)** eingeführt, die die Eingangssequenz zu einer umfangreichen Folge in einem höherwertigen Zahlensystem expandiert. Das Zahlensystem der Expansion ist wählbar zwischen 64 bis 96. Hier wird die **Basis 77** festgelegt. Für jedes Zeichen der Eingabeseqzenz errechnet das Verfahren den dezimalen Wert (s_i), der dann zu (**di**) **-** Ziffern im Zahlensystem zur Basis 77 - umgewandelt wird. Gleichzeitig ermittelt das Verfahren die Summe aller Einzelergebnisse (s_i) als zusätzlichen Wert **H(p)** zur Bestimmung verschiedener Steuerungsparameter und fügt die Ergebnisse (d_i) seriell zur Hashfunktionsfolge (HF) zusammen.

$$s_i = (a_i + 1) * p_i * H_k + p_i + code + r$$

$$s_i \rightarrow d_i \quad \text{(base 77)}$$

$$HF = d_1 + d_2 + d_3 + \ldots + d_i + \ldots d_m$$

(m = Anzahl der Zahlen im System zur Basis 77)

$$H_p = \sum_{i=1}^{n} S_i$$

$$H_p = 618326331960$$

Das gewählte Zahlensystem zur Basis 77 umfasst die folgenden Ziffern:

0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz&#@àáâãäåæçèéêë
(definiert vom Autor, nicht standardisiert)

Hashfunktionsfolge

Bei der Generierung der Hashfunktionsfolge ergibt sich beispielsweise für die Teilsequenz "**wandern**" an den Positionen 15 bis 21 der Eingabesequenz folgende Berechnung:

char	pi		Hk	(ai+1)*pi*Hk	Si	Basis 77
(ai+1)	(ai+1)*pi			pi+code+r		
w 120	15	1800	6928259	12470866200	17	12470866217 4kwZge
a 98	16	1568	6928259	10863510112	18	10863510130 412pàB
n 111	17	1887	6928259	13073624733	19	13073624752 4#äwOY
d 101	18	1818	6928259	12595574862	20	12595574882 4oNmHi
e 102	19	1938	6928259	13426965942	21	13426965963 4èètuQ
r 115	20	2300	6928259	15934995700	22	15934995722 5ãNSOU
n 111	21	2331	6928259	16149771729	23	16149771752 5éV&êb
				Summe:		618326331960 2éXgrâT

Die Hashfunktionsfolge **HF** umfasst 249 Ziffern im Zahlensystem zur Basis 77:

B2à7áD0kaxkspUV1BMw&m1aP7691gXfvG1èTBAU2DA6#e2NTHxz2qçkVJ3MoKzk3AIE
ç53#n@kp1E4HQG4kwZge412pàB4#äwOY4oNmHi4èètuQ5ãNSOU5éV&êb1á6nJD7oMq
GK66LuD46eiOrá7oMqGN2Lknçe7ISFAG7hë40â8eNYé#2llàä05ä75a89sèFæ29vdbå8Klé
pàN9Usi&BAdY4mn3GBa576qëUhjAY9O9CBXysGuCibçW9

Die Variablen sind Ziffern (keine Zeichen) im Zahlensystem zur Basis 77. Es gibt keinen Weg zurück zur Startsequenz (erste **Einweg-Funktion**). Gleichzeitig errechnet das Programm die folgenden Bestimmungsfaktoren:

Bestimmungsfaktoren

Trenn-Konstante C(k):	1681
Positionsgewichteter Wert (H_k):	6928259
Bestimmungswert (H_p):	618326331960
Gesamtwert (H_p+H_k):	618333260219

Aus den Bestimmungsfaktoren werden folgende Steuerungsparameter abgeleitet:

Variante	(H_k MOD 11) +1	=	9	Beginn der Kontraktion
Alpha	((H_k + H_p) MOD 255) +1	=	150	Offset Chiffre-Alphabet
Beta	(H_k MOD 169) +1	=	105	Offset Block-Schlüssel
Gamma	((H_p + code) MOD 196) +1	=	94	Offset Matrix-Schlüssel
Delta	((H_k + H_p) MOD 155) +code	=	20	dynamische Bitfolgen
Theta	(H_k MOD 32) +1	=	4	Offset Rückrechnung
Omega	(H_k MOD 95) +1	=	5	Beginn Doppelzeichen
Kappa	(H_k MOD 16484) +1	=	4980	Länge Doppelzeichen

Die Steuerungsparameter dienen zur Lösung verschiedener kryptographischer Aufgaben.

Kontraktion

Um die Bestimmungsbasis auf dezimale Größen zurückzuführen wird eine **Kontraktion** eingeführt. Für die Ziffern der **Hashfunktionsfolge** wird das Zahlensystem zur **Basis 78** (Expansions-basis +1) unterstellt. Jeweils drei Ziffern der Hashfunktionsfolge werden seriell durch **MODULO 256** in dezimale Zahlen 0 bis 255 (ohne Wiederholung) zurückgerechnet. Der Parameter **Theta** wird abgezogen.

Die ersten sechs Rückrechnungen ab **Variante** = **9** zeigen sich wie folgt:

3 Ziffern Basis 78	dezimal	Modulo 256	- Theta	Element
axk	223672	184	4	180
xks	362598	102	4	98
ksp	284127	223	4	219
spU	332544	0	4	252
pUV	312655	79	4	75
UV1	184939	107	4	103

Basis Variation

Die Ergebnisse der Kontraktion (Rückrechnungen) werden im Array BASIS VARIATION gespeichert, einem Array von 16x16 Elementen. Eine rückwärts gerichtete Bestimmung vorhergehender Daten ist nicht möglich (zweite **Einweg-Funktion**).

BASIS-VARIATION (256 Elemente)
Verteilung der Elemente

```
180 098 219 252 075 103 017 048 086 190 120 021 046 209 049 072
053 083 088 173 091 247 206 129 211 027 193 148 146 202 157 132
002 183 238 106 079 163 127 153 055 192 128 164 234 063 185 133
136 047 036 154 016 184 040 102 050 080 169 165 249 171 104 051
052 191 100 008 253 084 254 200 162 074 003 056 188 089 213 092
143 081 225 030 223 131 121 149 214 189 110 095 118 073 022 108
045 147 085 218 066 205 174 159 042 057 220 134 196 123 087 231
115 175 224 226 137 150 090 135 068 170 044 194 195 093 038 125
054 058 215 059 009 240 028 060 197 179 216 172 061 198 199 064
076 024 069 107 152 094 208 241 096 010 151 139 138 201 020 101
082 111 227 099 203 062 207 217 004 005 140 011 204 221 210 222
248 006 212 007 018 158 161 186 029 255 124 105 167 109 176 025
070 019 141 112 237 228 116 065 155 250 156 229 160 168 067 113
230 177 122 071 187 114 026 251 097 232 023 126 077 178 166 012
233 078 039 181 182 235 236 117 119 130 239 242 142 144 243 145
244 245 246 000 041 001 013 014 015 031 032 033 034 035 037 043
```

Indexierung

Für die Berechnung der CypherMatrix werden die Elemente der BASIS VARIATION einer Permutation unterworfen. Dabei werden die Indexwerte der Matrix (16x16) in einem zweidimensionalen Array neu generiert (Randomfolgen 1 bis 16).
Auszug aus dem Quellcode:

```
SHARED IndexFolge(2,16), Delta, Omega

FOR B = 1 TO 2
   IF B=1 THEN X = Delta
   IF B=2 THEN X = Omega
   FOR C = 1 TO 16
      INCR X
      IF X > 256 THEN X = 1
      A = VARIATION(X) MOD 16            Daten aus der BASIS-VARIATION
      IndexFolge(B,C) = A
      IF C>1 THEN
         L = 0
         DO
           INCR L
           IF IndexFolge(B,L) = A THEN
              INCR A
              A = (A MOD 16)
              IndexFolge(B,C) = A
              L = 0
           END IF
         LOOP UNTIL L = C-1
      END IF
      IndexFolge(B,C)  =  IndexFolge(B,C) + 1    Array IndexFolge (2,16)
```

```
    NEXT C
 NEXT B

N = Alpha
FOR I = 1 TO 16
   FOR J = 1 TO 16
      X = IndexFolge(1,I)
      Y = IndexFolge(2,J)
      Matrix$(X,Y) = VARIATION$(N)          Generierung der CypherMatrix
      INCR N
      IF N > 256 THEN N = 1
    NEXT J
 NEXT I
```

Index-Folge 1: 12 8 15 2 4 13 3 5 6 11 14 7 9 10 16 1
Index-Folge 2: 8 2 1 7 15 9 6 16 3 4 10 11 5 12 14 13

CypherMatrix

Mit der CypherMatrix wird eine eindeutige Bestimmungsbasis für die Analyse der Eingabe (hier: Startsequenz) erreicht. Nach der Wahrscheinlichkeitsrechnung entsteht eine gleiche CypherMatrix erst in **256!** (Fakultät) = **8E**+**506** Fällen.

Endgültige CypherMatrix (Variante D)

1	3C 1C C6 C7 18 AC C5 F0 D8 40 4C 45 98 6B B3 3D	*16*
17	41 74 A8 43 B1 E5 9B E4 9C 71 E6 7A BB 47 FA A0	*32*
33	0E 0D 23 25 62 21 0F 01 20 2B B4 DB 4B FC 1F 22	*48*
49	FB 1A B2 A6 4E 7E 61 72 17 0C E9 27 B6 B5 E8 4D	*64*
65	30 11 D1 31 53 15 56 67 78 48 35 58 5B AD BE 2E	*80*
81	81 CE CA 9D B7 94 D3 F7 C1 84 02 EE 4F 6A 1B 92	*96*
97	C8 FE 59 D5 51 38 A2 54 03 5C 8F E1 DF 1E 4A BC	*112*
113	D9 CF DD D2 06 0B 04 3E 8C DE F8 D4 12 07 05 CC	*128*
129	95 79 49 16 93 5F D6 83 6E 6C 2D 55 42 DA BD 76	*144*
145	9F AE 7B 57 AF 86 2A CD DC E7 73 E0 89 E2 39 C4	*160*
161	99 7F 3F B9 2F A4 37 A3 80 85 88 24 10 9A C0 EA	*176*
177	F1 D0 C9 14 6F 8B 60 5E 97 65 52 E3 CB 63 0A 8A	*192*
193	75 EC 90 F3 F5 F2 77 EB EF 91 F4 F6 29 00 82 8E	*208*
209	66 28 AB 68 BF A5 32 B8 A9 33 34 64 FD 08 50 F9	*224*
225	BA A1 6D B0 13 69 1D 9E 7C 19 46 8D ED 70 FF A7	*240*
241	87 5A 5D 26 3A C2 44 96 2C 7D 36 D7 09 3B AA C3	*256*

Die zur Steuerung des Verfahrens erforderlichen Parameter werden der CypherMatrix in jeder Runde an definierten Positionen entnommen.

Steuerungsparameter

Ab Position Alpha = **150** werden 128 Zeichen als Chiffre-Alphabet des Bitsystems zur Basis 7 entnommen. Bestimmte Zeichen (hex: 00 bis 20, 22, 2C, B0, B1, B2, D5, DB, DC, DD, DE, DF und weitere) werden ausgeklammert, weil sie in einigen Situationen noch ihre ursprünglichen Aufgaben wahrnehmen (zB. **1A** =ASCII-26) und die ordnungsgemäße Durchführung des Programms stören.

Chiffre-Alphabet (hex)

```
3C .. C6 C7 .. AC C5 F0 D8 40 4C 45 98 6B B3 3D
41 74 A8 43 .. E5 9B E4 9C 71 E6 7A BB 47 FA A0
.. .. 23 25 62 21 .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. 86 2A CD .. E7 73 E0 89 E2 39 C4
99 7F 3F B9 2F A4 37 A3 80 85 88 24 .. 9A C0 EA
F1 D0 C9 .. 6F 8B 60 5E 97 65 52 E3 CB 63 .. 8A
75 EC 90 F3 F5 F2 77 EB EF 91 F4 F6 29 .. 82 8E
66 28 AB 68 BF A5 32 B8 A9 33 34 64 FD .. 50 F9
BA A1 6D .. .. 69 .. 9E 7C .. 46 8D ED 70 FF A7
87 5A 5D 26 3A C2 44 96 .. 7D 36 D7 .. 3B AA C3
```

Chiffre-Alphabet (ASCII)

```
  1    † * Í ç s à ‰ â 9 Ä ™ □ ? ¹ / ¤    16
 17    7 £ € … ˆ $ š À ê ñ Ð É o ‹ ` ^    32
 33    — e R ã Ë c Š u ì # ó õ ò w ë ï    48
 49    ' ô ö ) , Ž f ( « h ¿ ¥ 2 ¸ © 3    64
 65    4 d ý P ù ° ¡ m i ž | F # í p □    80
 81    § ‡ Z ] & : Â D – } 6 × ; ª Ã <    96
 97    Æ Ç ¬ Å ð Ø @ L E ˜ k ³ = A t ¨   112
113    C å › ä œ q æ z » G ú   # % b !   128
```

Block-Schlüssel

Der ab Position **105** der CypherMatrix entnommene Block-Schlüssel umfasst 42 Zeichen:

#\Åáß#J¼ÙÏÝÒ###>ŒÞøÔ###Ì•yl#“_Ö*ƒ*nl-UBÚ½vŸ®

```
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. 03 5C 8F E1 DF 1E 4A BC
D9 CF DD D2 06 0B 04 3E 8C DE F8 D4 12 07 05 CC
95 79 49 16 93 5F D6 83 6E 6C 2D 55 42 DA BD 76
9F AE .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
```

Matrix-Schlüssel

Als Startsequenz für die nächste Runde entnimmt das Verfahren ab Position **94** einen neuen Matrix-Schlüssel mit 42 Zeichen:

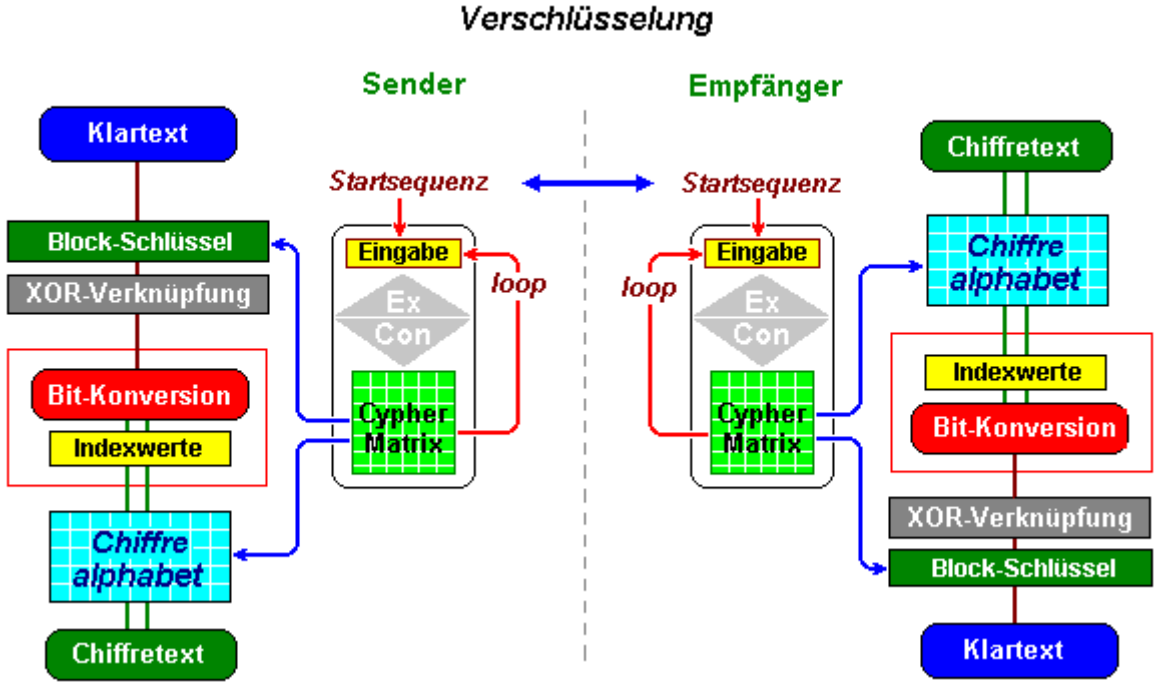
j#'ÈþYÕQ8¢T#\Åáß#J¼ÙÏÝÒ###>ŒÞøÔ###Ì•yl#"_Ö

Der Matrix-Schlüssel wird auf den Beginn der Funktion zurückgeführt (**loop**). Die jeweiligen Matrix-Schlüssel steuern den gesamten Ablauf des Verfahrens, inhaltsgleich, sowohl beim Sender als auch beim Empfänger.

```
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. 6A 1B 92
C8 FE 59 D5 51 38 A2 54 03 5C 8F E1 DF 1E 4A BC
D9 CF DD D2 06 0B 04 3E 8C DE F8 D4 12 07 05 CC
95 79 49 16 93 5F D6 .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
```

Der Codierbereich

Die Verschlüsselung – das Schreiben und Lesen von geheimen Informationen – findet ausschließlich im Codierbereich statt. Mit Eingabe der gleichen Startsequenz, sowohl beim Sender als auch beim Empfänger, werden im gesamten Verfahren ein identischer Verlauf und identische Steuerungsparameter erzeugt. Das folgende Schema zeigt die Zusammenhänge:



Die Verschlüsselung wird in folgenden Alternativen durchgeführt:

Basis-Coding: Bit-Konversion allein ohne weitere Operationen oder
Verbund-Coding: Bit-Konversion mit zusätzlichen Operationen, und zwar:

a) mit XOR-Verknüpfung (voran- oder nachgestellt) oder
b) mit weiteren Operationen verbunden.

Nach den vorstehenden Grundsätzen hat der Autor eine Reihe von Programmen entwickelt, die in der folgenden Liste zusammengestellt sind.

System Basis	System Alphabet	Basis-Coding (ohne XOR-Funktion) einfache Matrix	Verbund-Coding (mit XOR-Funktion) einfache Matrix	Längen-verhältnis
1	2	Crypto01	MonoCode	1:8
2	4	Crypto02	ZweiCode	1:4
3	8	Crypto03	DreiCode	1:2,66
4	16	Crypto04	VierCode	1:2
5	32	Crypto05	QuinCode	1:1,6
6	64	Crypto06	CM64Code	1:1,33
7	128	Crypto07	DataCode DynaCryp CodeData[1)] QuadCode[2)]	1:1,143 1:1,143 1:1,143 1:1,143
8	256	Crypto08 CMCode8D System08	PlanCode MyCode08	1:1 1:1 1:1
9	512	Crypto09 System09	NeunCode MyCode09	1:1,79 1:1,79
10	1024	Crypto10 System10	ZehnCode MyCode10	1:1,6 1:1,6
11	2048	Crypt11A Crypt11B System11	ElvaCode MyCode11	1:1,46 1:1,46
12	4096	Crypto12 System12	MegaCodA MegaCodB MyCode12	1:1,33 1:1,33 1:1,33
13	8192	System 13	MyCode13	1:1,23
14	16384	System14	MyCode14	1:1,143

[1)] Programm mit drei Operationen (XOR – bit conversion - exchange),
[2)] Programm mit vier Operationen (dyn24 – XOR – bit conversion – exchange).

Die Programme können einzeln oder in Gruppen mit oder ohne Quellcode beim Autor per e-mail angefordert und im Rahmen der **CMLizenz** getestet oder weiter entwickelt werden.
Alle Programme sind DOS-Programme und laufen nur noch unter Windows XP.
Sie können auch das Analyseprogramm **"StepConD.exe"** herunter laden und damit die genannten Programme testen.
E-mail Adresse: eschnoor@multi-matrix.de

München, im April 2013

Hinweis

[#1] Alle Aussagen beruhen auf den Forschungsergebnissen des Autors. Sie sind zur Erweiterung der Wissenschaft der digitalen Kryptographie bestimmt. Erläuterungen dazu im whitepaper " Neue Techniken in der digitalen Kryptographie"