Alternatives in digital Cryptography

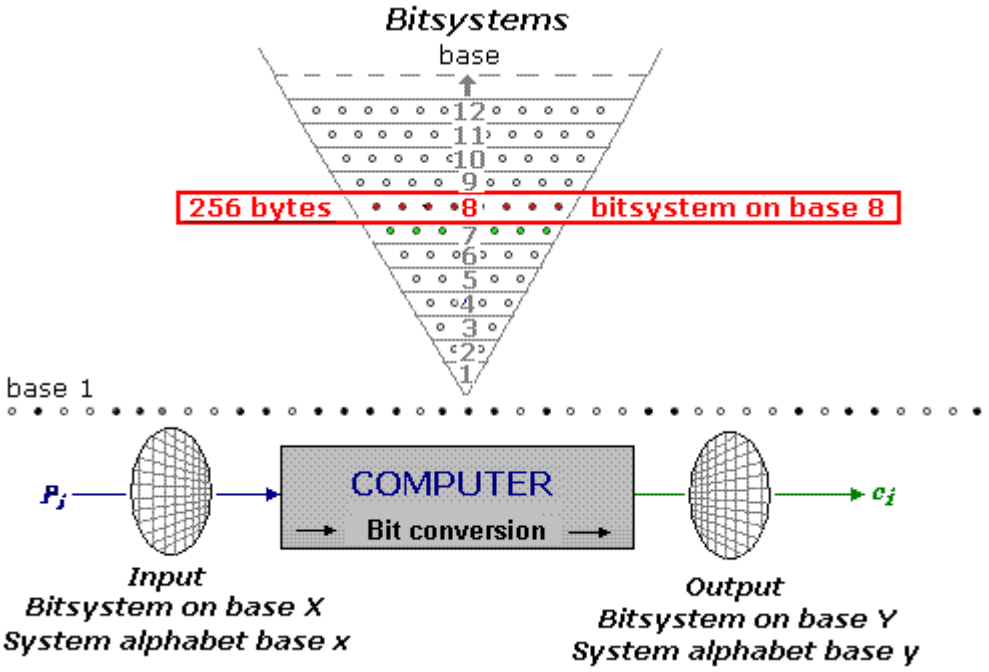
(Ernst Erich Schnoor)

Since centuries cryptographers used for enciphering exclusively single characters of their concerned language. When computers came up methods of ciphering changed to digital techniques. But computer on principle only can distinguish between two states: „present“ (voltage) and „not present“ (no voltage), in digits of number system on base 2: „**one**“ or „**zero**“. This connection is defined as „**bit“** as everybody knows.

An information is multilateral. It comprises two bits, at least, in general: a **series** of **bits**. Their quantity is unlimited (1 to ∞). In order to work with bit series systematically they have to be systemized, that means: to be scaled and devided into definite segments (**units**). It is a similar phenomenon compared with the quantity of all numbers. Comparable to numerical theory bit series can be systemized in a significance order system. Hence, 8 bit sequences may be named as „**bitsystem on base 8**“. Abridged:

				system alphabet
bit sequences:	1-bit	= bitsystem on base 1	= 2^1 signs =	2 units
	3-bit	= bitsystem on base 3	= 2^3 signs =	8 units
	6-bit	= bitsystem on base 6	= 2^6 signs =	64 units
	7-bit	= bitsystem on base 7	= 2^7 signs =	128 units
	8-bit	**= bitsystem on base 8**	**= 2^8 bytes =**	**256 bytes**
	11-bit	= bitsystem on base 11	= 2^11 signs =	2048 units
	13-bit	= bitsystem on base 13	= 2^13 signs =	8192 units
	16-bit	= bitsystem on base 16	= 2^16 signs =	65536 units

Circumstances are detailed explained in whitepaper “www.telecypher.net/ParadigmaEn.pdf [#1]. Structure of the significance order system is best described by the system pyramid turned on its head.



1 System base of current processes

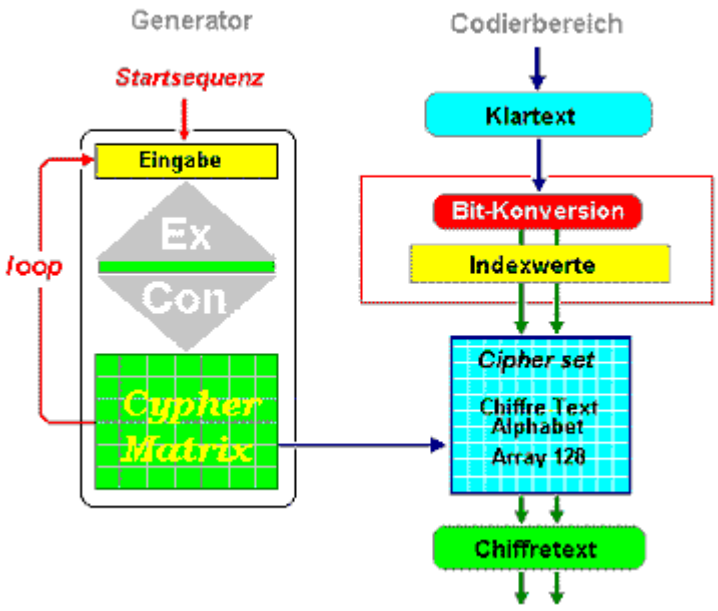
In present cryptography almost all operations are processed in bitsystem on **base 8.** Inputs are performed in bitsystem on base 8 with system alphabet of 256 signs (00 up to FF) and encrypted results (ciphertexts) are issued in bitsystem on base 8 with system alphabet of 256 signs, as well. If manifold longer bit series are manipulated between input and output (coding function) this will be of no significance for the order system. Insofar all encrypting procedures – independently of numbers of bits in the code function – take place in a uniform **order system,** in bit system on **base 8**.

By this matter of fact and the demand plaintext and ciphertext must have the same length [#2,#9], the **action range** of current cryptography is narrowed to the field of bitsystem on **base 8** including the system alphabet (cipher alphabet) of **256 bytes.** This narrowness can be avoided by new alternatives – the CyperMatrix procedure (named by the author) [#3].

2 Alternative procedures

In CypherMatrix procedures – explained in article: "Basis of CypherMatrix procedure" - encryption processing is relative simple:

> **Generator** creates the necessary **system alphabet** and
> in **coding area** ciphertext will be written by **bit conversion.**



2.1 „System alphabet“

The System Alphabet is the most important component of computer techniques. It is the base for visualization of bit series content. Without definition of an appropriate alphabet a computer will not be able to work, at all.

2.2 „Bit conversion“

Bit conversion is changing of bit series from one bitsystem to another bitsystem. Number of bits and their order remain unchanged. No bit will be added and no bit will be removed. Number of bits in one unit will change only and by this the structure of the bit series too. Decimal values of the new units link to indexation of the assigned system alphbet. Bit conversion may be performed for all bitsystems from base 1 up to base 14 (and higher).

3 Changing bitsystems

Changing bitsystems presuppose the feature that length of bit series to be converted can be devided by number of foregoing units and by number of aimed units, as well, because otherwise some bits may be lost. For example a conversion from base 8 to base 7 perform as follows:

foregoing units:
01000010 01101100 01110101 01100101 00100000 01101000 01100101 = 7 units
aimed units:
0100001 0011011 0001110 1010110 0101001 0000001 1010000 1100101 = 8 units

Length of bit series with 56 bits is devisible by 7 and by 8 as well. As a result of conversion ciphertext will be longer than plaintext by ratio **8/7**, that means: from one plaintext character results **1,143** ciphertext character. Ciphertext and plaintext have no equal length, any longer.

3.1.1 Bit conversion from base 1 to base 8

In historical view conversion has already been developed in 1963 at beginning of digital performances when a 7-bit code combination was established for character recognition (ASCII-standard code). When system alphabet of 128 signs became to narrow an 8-bit code with an extended ASCII-character set of 256 elements (system alphabet) was introduced , which furthermore is used as a standard character representation.

By particular approach this handling already implements a coding procedure. It will scarcely be perceived as such but rather as basis of digital communication (standard code). Conversion from base 1 to base 8 in detail occurs as follows:

bit series base 1:
01000010011011000111010101100101001000000110100001100101011000010111011 0
bit series base 8:
01000010 01101100 01110101 01100101 00100000 01101000 01100101 01100001 0111

index:	66	108	117	101	32	104	101	97
system alphabet	B	l	u	e	□	h	e	a

system alphabet (ASCII): decimal hexa

alphabet$(32) = □ 20
alphabet$(66) = B 42
alphabet$(97) = a 61
alphabet$(101) = e 65
alphabet$(104) = h 68
alphabet$(108) = l 6C
alphabet$(117) = u 75

index (hexa) = system alphabet (hexa):
42 6C 75 65 20 68 65 61
In plaintext the output reads: „**Blue heaven**“
in hexa: „**426C756520686561. . .**“

The underlying bit series in bitsystem on base 1 remain unchanged (no bit is added and no bit is removed). The example elucidates that the system alphabet only has to be changed while the unstructured bit series (number and order of sequences) on principle remain the same at each conversion.

3.1.2 Variation of system alphabet

With **generator** and start sequence „**Donky racing on the banks of San Bernadino**“ a permutated system alphabet is created, which leads to following variation of the base encryption:

```
  1 ®Žf E™ï¿½Ì"c$°Î'#ã£ï¿½eú,²¥ áRgjÔ¦å§USTÏA#çNX,Vk#¨#JCIÈ÷õÉùGôK›.nÊ@   64
 65 0èý5¤Do tpx©€#‹:#Y[üÚº„ZWy6ï□]ÞÐÒhË½Åê%I\#(±ß´z^8×…q³ï¿½æ·rÀÁéˆ9s=  128
129 i#äÝ_O#ò1)`â###&¬##Íþša#‡»d«#ï¿½'¶ÑHµàøœb¯/Œ Æ;Q2Ó###vP—¡4ìÜ{ž¼¢Ÿ 192
193 #ï¿½!LM¾<ë3BÛw##û#*˜|#mÖÕ¿ªƒ Ø#7‰†ö?íF~>uŠ#¸#î}+Ãñ“Â#”ð•ÙÇ#Äó-–  256
```

new system alphabet:

alphabet$(32) = å
alphabet$(66) = è
alphabet$(97) = Ò
alphabet$(101) = Å
alphabet$(104) = l
alphabet$(108) = ±
alphabet$(117) = ³

Plaintext „Blue heaven“ leads to cipertext „**è±³ålÅÒ**“ **. . . . .**

This example demonstates the simpliest mode of encryption. The system alphabet only has to be changed in text blocks of e.g. 16 bytes (new round). For testing purposes you may download the program: „System08.exe“ and test it by yourself. The source code is included as well.

3.2 Bit conversion from base 8 to base 13

The conversion is performed by the program: „**System13.exe**“. Despite of converting from bitsystem on base 8 to bitsystem on base 13 number of bits and their order remain unchained. No bit ist removed and no bit is added. Only distances of the bits change. 8-bit sequences modify to 13-bit sequences. Dezimal values of the new units become index values for the signs in system alphabet on base 13. For this 2^13 signs = 8192 signs are necessary. Because of this size there are no single characters available the digits of **number system** on **base 128** – that are 16384 digits – are used as double signs. The system alphabet covers a permutated range beginning with Kappa+1, that is from digit 5193 up to 13385 = 8192 double signs.

bit series base 8:
01000010 01101100 01110101 01100101 00100000 01101000 01100101 01100001 011
index: 66 108 117 101 32 104 101 97
system alphabet
B l u e □ h e a
bit series base 13:
0100001001101 1000111010101 1001010010000 0011010000110 0101011000010 11
index: 2125 4565 4752 1670 2754
index+1: 2126 4566 4753 1671 2755
base 13 vN ëV ì† rï &C

system alphabet (base 13): Alphabet$(2126) = vN
Alphabet$(4566) = ëV
Alphabet$(4753) = ì†
Alphabet$(1671) = rï
Alphabet$(2755) = &C

In bitsystem on base 13 „Blue heaven“ is coded as follows: **vNëVì†rï&C . . .**

3.3 Bit conversion from base 8 to base 7

Bit conversion from base 8 to base 7 is the main application for encryption.
In following example a conversion is demonstrated by program: **Crypto07.exe**.

bit series base 8:
01000010 01101100 01110101 01100101 00100000 01101000 01100101 01100001 011
index: 66 108 117 101 32 104 101 97
system alphabet
B l u e □ h e a
bit series base 7:
0100001 0011011 0001110 1010110 0101001 0000001 1010000 1100101 0110000 101
index:33 27 14 86 41 1 80 101 48
(+1) 34 28 15 87 42 2 81 102 49
cipher alphabet base 7:
Ç 3 ù ï Ã t T ¬ ³

Plaintext „Blue heaven“ reads in bitsystem on base 7 as follows: **Ç3ùïÃtT¬³**

System alphabet on base 7 with 128 characters is created by the generator and the start sequence: „**Blue heaven over Minnesota all the day long**“. Index values are added by (+1), because the program does not recognize index „0“.

cipher alphabet (Base 7)

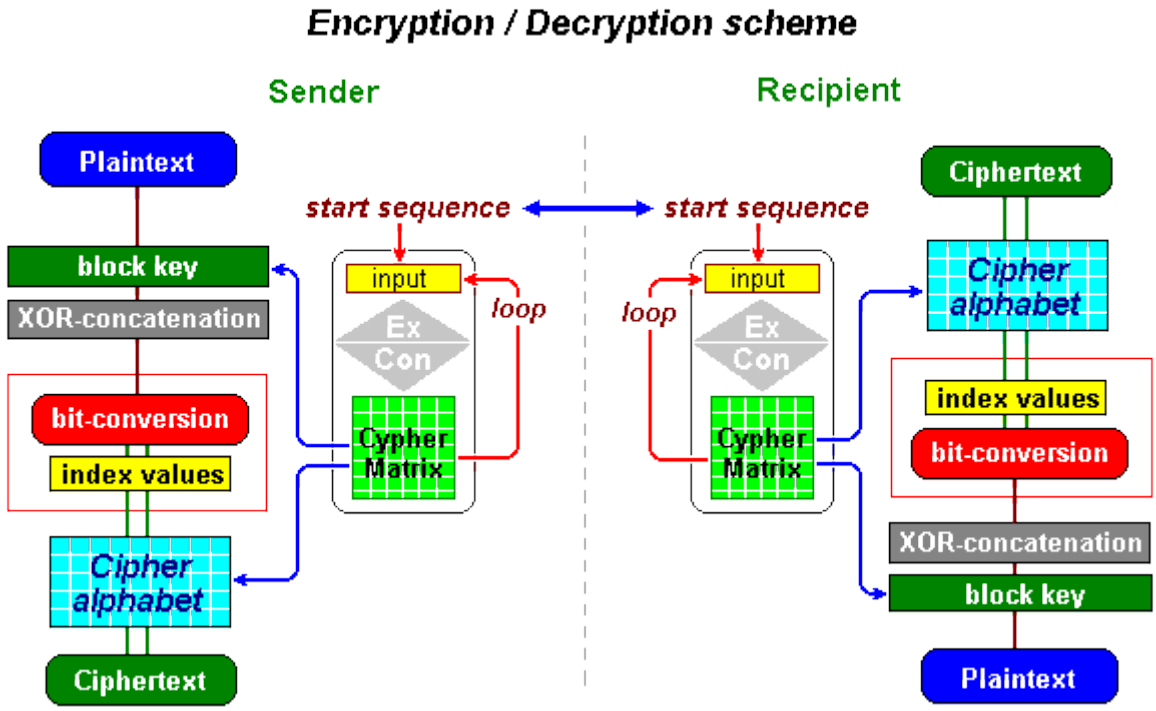
```
  1   P t    ¨ 1 ‹ ø Ÿ ƒ ¯ Ä [ ¡ Â ù ¢ ¼ é % ú □ å ‘ | ö û „ 3 } ’ ¤ '      32
 33  © Ç B “ W ; Å 7 e Ã Æ , – 9 ì L ³ ü ý ã ¥ R ¶ ª > & 8 þ 6 Ë µ æ    64
 65   — m ž ( ½ ˜ f « ~ ê à u : ë Y □ T † c È î º ï g ¡ £ * w € · < ¸      96
 97  ¥ # - á l ¬ Î ¾ U X • ‡ D H _ Í ð … ¿ À É Ó ! + Ï ˆ Z $ E Ô Á Ì  128
```

Cipher aphabet (hex)

```
  1   50 74 A0 A8 31 8B F8 9F 83 AF C4 5B 69 8F F9 A2   16
 17   BC E9 25 FA 7F E5 91 7C F6 FB 84 33 7D 92 A4 27   32
 33   A9 C7 42 93 57 3B C5 37 65 C3 C6 82 96 39 EC 4C   48
 49   B3 FC FD E3 9D 52 B6 AA 3E 26 38 FE 36 CB B5 E6   64
 65   97 6D 9E 28 BD 98 66 AB 7E EA E0 75 3A EB 59 7F   80
 81   54 86 63 C8 EE BA EF 67 A1 A3 2A 77 80 B7 3C B8   96
 97   A5 23 2D E1 6C AC CE BE 55 58 95 87 44 48 5F CD  112
113   F0 85 BF C0 C9 D3 21 2B CF 88 5A 24 45 D4 C1 CC  128
```

4 Encryption

Encryption – writing and reading of secret informations – is exclusively performed in the coding area. By inserting an identical start sequence at sender and recipient es well an equal course and identic control parameters are generated. The following scheme shows the connections:



Ciphering is performed by the following alternatives:

1. **Basic Coding:** bit conversion without further operations or
2. **Compound Coding:** bit conversion with additional operations ,

   a) with XOR-concatenation (foregoing or succeding) or
   b) connected with further operations (see: "Combinierte Operationen")

According to this principles the Author developed a couple of programs which are shown in the following list:

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)] program with three operationens (XOR – bit conversion - exchange),
[2)] program with four operationens (dyn24 – XOR – bit conversion – exchange).

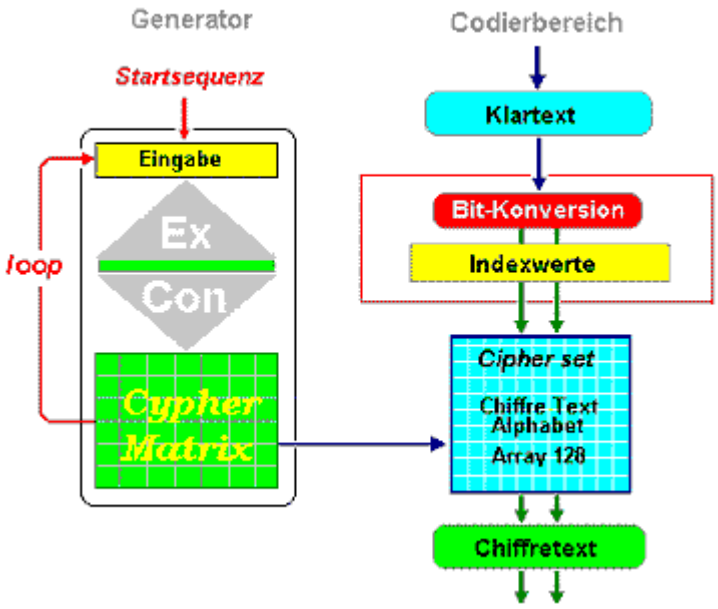
Programs may single or in groups with or without sourcecode per e-mail requested at the author and tested or further developed within the scope of the **CMLizenz.** All programs are Dos-programs and will run under WindowsXP only.
They have to be converted to **C#,** as already done under direction of Prof. **Bernhard Esslinger** (University of Singen) and his CrypTool-team (especially: **Michael Schäfer)** with the programs **DataCode** and **DynaCode** [#4]. All further programs have still to be converted.

4.1 Basic Coding

In „basic coding“ modus the bit conversion is performed directly from the forgoing bitsystem to the aimed bitsystem. As example: from **base 8** to **base 7**:

Changing from plaintext to ciphertext are processed by two functions:

1. Bit conversion
   8-bit plaintext values → 7 bit index values (0 ...127)
2. Destination of ciphertext
   7-bit Index values → cipher alphabet (0...127) → ciphertext.



As an example of **Basic Coding** a text from **Mark Twain** is encrypted with program „**System12.exe**“ and start sequence: „**Donkey racing on the banks of San Bernadino**“ (file: language.txt/ content: 612 bytes).

In German, all the Nouns begin with a capital letter. Now that is a good idea; and a good idea, in this language, is necessarily conspicuous from its lonesomeness. I consider this capitalizing of nouns a good idea, because by reason of it you almost able to tell a noun the minute you see it. You fall into error occasionally, because you mistake the name of a person for the name of a thing. and waste a good deal of time trying to dig a meaning out of it. German names almost always do mean something, and this helps to deceive the student.

The Awful German Language, Mark Twain, A Tramp Abroad, 1880

As first plaintext block 36 bytes are inserted:

In German, all the Nouns begin with□

49 6E 20 47 65 72 6D 61 6E 2C 20 61 6C 6C 20 74 68 65
20 4E 6F 75 6E 73 20 62 65 67 69 6E 20 77 69 74 68 20

Plaintest in bitsystem on **base 8** (36x8=288) is converted into segmemts of aimed bitsystem on **base 12** (288:12=24).

base 8: I n G e r m a
01001001 01101110 00100000 01000111 01100101 01110010 01101101 01100001 . . .
base 12:
010010010110 111000100000 010001110110 010101110010 011011010110 0001 . . .

Index:	1174	3616	1142	1394	1750
(+1)	1175	3617	1143	1395	1751
ciphertext:	îi	Žs	îC	{8	}˜

system alphabet base 12:

Alphabet$(1143) = îC
Alphabet$(1175) = îi
Alphabet$(1395) = {8
Alphabet$(1751) = }˜
Alphabet$(3617) = Žs

Ciphertext reads as follows:

îiŽsîC{8}˜è4ê„å£}ˆŠs€ˆ,§éQ<u>5€˜</u>9éSé§}C…4éT€¬€ˆ,sjLX'jQX˜lèZ›b9v"
léf"lbzZg•#¦bAf—jQfZkQdZjJxDj¥#ž,êéž,î€–{xé–ƒ›|f,Réœƒ¬"™íFˆ™,‹ër
íFˆ£íG|ƒƒWïfƒîë£c¦a<u></u>c¥alU9i¢U9s"cva¢evS¡dPo©U9Wžd–WŸdPW¤d¦a¢R}k•
X'oiNäcpX<u></u>KhXrmaX¡oiWLmaX¢P™Nâb‹V¡ojX¡cZWMN‹Y5ae•'ˆN"êˆT•¦ŠW"êœT

```
EE 69 8E 73 EE 43 7B 38 7D 98 E8 34 EA 84 E5 A3 7D 88 8A 73 80
88 82 A7 E9 51 8F 35 80 98 8F 39 E9 53 E9 A7 7D 43 85 34 E9 54
80 AC 80 88 82 73 6A 4C 58 92 6A 51 58 98 6C E8 5A 9B 62 39 76
94 0D 0A 6C E9 66 94 6C 62 7A 5A 67 95 23 A6 62 41 66 97 6A 51
66 5A 6B 51 64 5A 6A 4A 78 44 6A A5 23 9E 82 EA E9 9E 82 EE 80
96 7B 78 E9 96 83 9B 7C 66 82 52 E9 9C 83 AC 94 99 ED 46 88 99
82 8B EB 72 ...
```

The encrypted file **Language.ctx** comprises 816 characters.
Through bit conversion 12 characters in bitsystem on base 8 comprise 96 bits which will be assigned to 8 double signs in bitsystem on base 12. Insofar there is a ratio of 1:1,333.

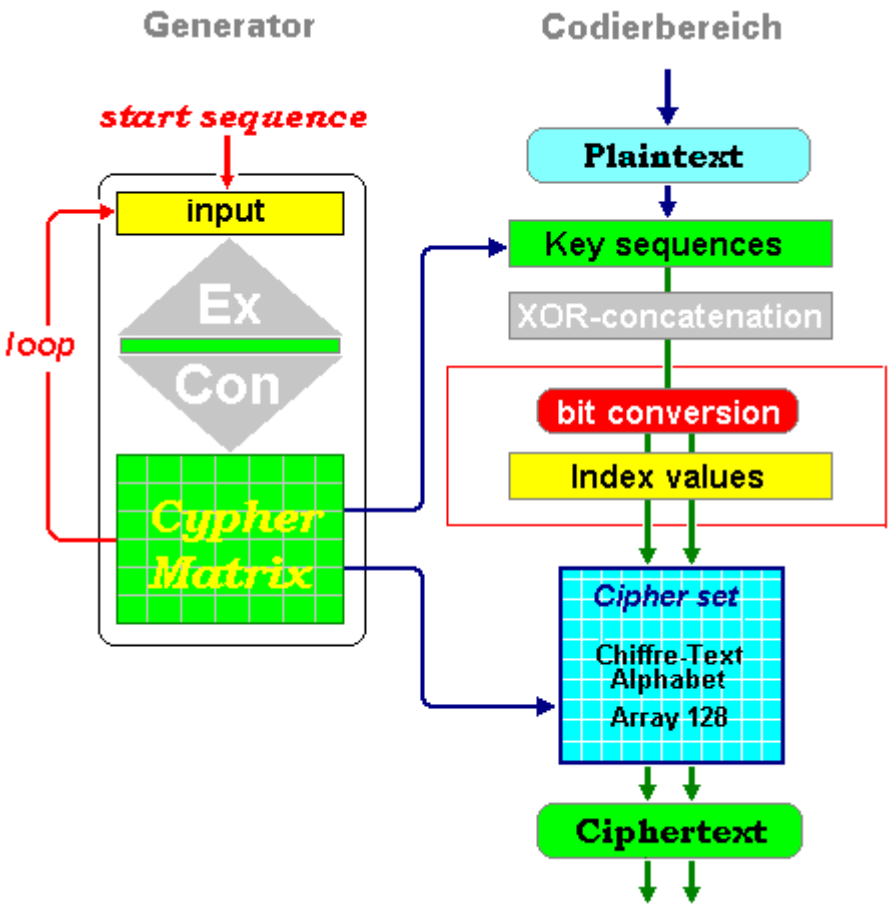
The system alphabet on base 12 needs 4096 signs, which are not available by single characters. So, they are generated as double signs by digits of number system on base 128 – that are 16384 digits.
Partial sourcecode:

```
SUB Alphabet
    SHARED Alphabet$(),  Kappa
    Kappa = 7979, 1202, 142, 8330, 1428  (anew generated in each round)

    FOR C=1 TO 4096
      X## = C + Kappa
      CALL DezNachSystem (128, X##, Zeichen$)
      Digit$ = „00“+Zeichen$
      Digit$ = RIGHT$(Digit$,2)
      Alphabet$(C) = Digit$
   NEXT C
END SUB
```

4.2 „Compound Coding“

By „Compound coding“ several operations are combined in succession with bit conversion and further operations (e.g. XOR concatenation, dyn24, exchange).



XOR concatenation is installed before bit conversion. Encryption then works in three functions:

1. partial dynamic „one-time-pad“
   plaintextblock → block key → 8-bit XOR concatenation
2. bit conversion
   8-bit XOR concatenation → 7 bit index values (0 ...127)
3. destination of ciphertext
   7-bit index values → cipher alphabet (0...127) → ciphertext.

Moduls are passing the procedure in succession.

```
. . . . . .
CALL  XORGenerator (InputData$,TransData$)          for „XOR“
CALL BitConversion (TransData$,DataTrans$)          for „bit conversion“
CALL Substitution (DataTrans$,ReportData$)          for „dyn24“
CALL ByteWechsel (ReportData$,OutputData$)          for „exchange“
. . . . . .
```

4.2.1 Encryption from base 8 to base 7

The following example demonstrates encryption steps of file **Museum.txt** with program **Datacode.exe** and the start sequence „**Yellowstoneparkbridge is under construction**“ (43 bytes)**.** Plaintext file contains 596 bytes.

> *The science of bird study in all its aspects is known as ornithology. A major task of the ornithology is to describe and name the birds of the world and to arrange them into species, genera, families, and higher categories of kinship. About 8700 species are known. There is still much to learn concerning the evolutionary relationships of the families and orders of birds. New methods in systematics, as applied to populations of closely related birds, are constantly leading to a better understanding of the process of evolution.*
>
> The American Museum of Natural History, New York 1972

In each round a plaintext block of 42 bytes (42x8 = 336 bits) is XOR concatenated with a block key of equal length. As first plaintext block the program reads:

The science of bird study in all its aspec (42 bytes)

54 68 65 20 73 63 69 65 6E 63 65 20 6F 66 20 62 69 72 64 20 73
74 75 64 79 20 69 6E 20 61 6C 6C 20 69 74 73 20 61 73 70 65 63

The block key extracted at position 115 from current CypherMatrix comprises:

†9ÉóB£WkÒ¥®”æ@#Ðë‡Í`Î‹#ÛIÏO5#ÑYd'ZVÇXTiû\c

86 39 C9 F3 42 A3 57 6B D2 A5 AE 94 E6 40 1D D0 EB 87 CD 60 CE
8B 1C DB 49 CF 4F 35 12 D1 59 64 27 5A 56 C7 58 54 69 FB 5C 63

plaintext base 8:
01010100 01101000 01100101 00100000 01110011 01100011 01101001 01100101 . . .
block key base 8:
10000110 00111001 11001001 11110011 01000010 10100011 01010111 01101011 . . .

XOR contatenation base 8:
11010010 01010001 10101100 11010011 00110001 11000000 00111110 00001110 . . .

As result arises a partial dynamic „**one-time-pad**“. Plaintext and block key are of equal length and the key will not be repeated. Each round gets another key from the respective CypherMatrix.

4.2.2 Bit conversion

The result of XOR-concatenation in bitsystem on base 8 (42x8 = 336 bits) is converted to characters of bitsystem on base 7 (336 bits = 48x7) . Decimal values of the changed signs (base 7) are index values to positions of cipher characters in the system alphabet on base 7 (cipher alphabet of 128 elements). Indexes have to be added with (+1) because the cipher alphabet array does not recognize the value „**0**“

XOR base 8:
11010010 01010001 10101100 11010011 00110001 11000000 00111110 00001110 . . .
conversion base 7:
1101001 0010100 0110101 1001101 0011001 1000111 0000000 0111110 0000111 0 . . .

index:	105	20	53	77	25	71	0	62	7
(+1)	106	21	54	78	26	72	1	63	8
cipher:	[	™	N	“	M	‘	i	¦	e

System alphabet on base 7

1	**i** û \ c ^ ³ o **e** Ž % ¨ Ú ˜ f Ë © v t ð ™ E) × • **M** Å Ä á „ » è	32
33	s Ì b í , n Á ÷ a Ù ¸ A Å C ‰ À (H · ¥ = **N** I J ¶] ~ m Ÿ Â ¦ Q	64
65	Ø / ¼ « Œ º ¾ ‘ ô D Ö ½ ª “ ¡ q à ƒ 7 Ã Ì Ó ú ; ý . w ù R ø Ô	96
97	x ï { þ ˆ y z □ â **[** | : } 2 S # ! < □ 4 $ 8 F □ G È ¬ ã u ? î ž	128

System alphabet (hex)

1	69	FB	5C	63	5E	B3	6F	65	8E	25	A0	A8	DA	98	66	CB	16
17	A9	76	74	F0	99	45	29	D7	95	4D	C5	C4	E1	84	BB	E8	32
33	73	8D	62	ED	82	6E	C1	F7	61	D9	B8	41	8F	43	89	C0	48
49	28	48	B7	9D	3D	4E	6C	4A	B6	5D	7E	6D	9F	C2	A6	51	64
65	D8	2F	BC	AB	8C	BA	BE	91	F4	44	D6	BD	AA	93	A1	71	80
81	E0	83	37	C3	CC	AD	D3	FA	3B	FD	2E	77	F9	52	F8	D4	96
97	78	EF	7B	FE	88	79	7A	7F	E2	5B	7C	3A	7D	32	53	23	112
113	21	3C	7F	34	24	38	46	90	47	C8	AC	E3	75	3F	EE	9E	128

Encryption of the plaintext file **Museum.txt** results in ciphertext file **Museum.ctx** with 720 characters, as follows:

Ciphertext (base 7)

[™N“M‘i¦eÀ•}R7tÁ»}àÀCnûÂž.J4eŸªwM□oàØázb.»o*ƒ*Ì□□i
ùe%§Š·¾-%û…ö*3í\„4T3ä□¨*ƒ*T,©–*ƒ*˜¾ •ä×ÍÎsG5□/&¥é³å0
?Çµ€Í?ržZ»¶ÒÒÃØõ¦IAæ{D¦;ëx#ÒYØ«nfñÂ>³6W½¡€€#}ÒèÄ
ª/´Ú”3¹Vû0ªþŠoÈ4Žú9Ë2\Š´‰·êKHÍˆ0ê□Òn¿þ1&è?Ž”’õÅn
l¥%FãÑD÷UcîÌOªü•ZE»Mh¸Øè„úØK\÷£NÎíÄDˆ×®nÍeÔªÄ´O. . . . . .

Cipher file (hex)

5B 99 4E 93 4D 91 69 A6 65 C0 95 7D 52 37 74 C1 BB 7D E0 C0 43 6E FB C2 9E
2E 4A 34 65 9F AA 77 4D 8F 6F E0 D8 E1 7A 62 2E BB 6F 83 CC 8F 7F 69 F9 65
25 A7 8A B7 BE 2D 25 FB 85 F6 2A 33 ED 5C 84 34 54 33 E4 7F A8 83 54 82 A9
96 83 98 BE A0 95 E4 D7 49 CE 73 47 35 8F 2F 26 A5 E9 B3 E5 30 3F C7 B5 80
CD 3F 72 9E 5A BB B6 D2 D2 C3 D8 F5 A6 6C 41 E6 7B 44 A6 3B EB 78 23 D2 59
D8 AB 6E 66 F1 C2 3E B3 36 57 BD A1 80 80 23 7D D2 E8 C4 AA 2F B4 DA 94 33
B9 56 FB 30 AA FE 8A 6F C8 34 8E FA 39 CB 32 5C 8A B4 89 B7 EA 4B 48 CD 88
30 EA 90 D2 6E BF FE 31 26 E8 3F 8E 94 92 F5 C5 6E 6C A5 25 AD 46 E3 D1 44
F7 55 63 EE CC 4F AA FC 95 5A 45 BB 4D 68 B8 D8 E8 84 FA D8 4B 5C F7 A3 4E
CE ED C4 44 88 D7 AE 6E CD 65 D4 AA C4 B4 4F . . . . . .

5 Deciphering

For deciphering the generator performes an identic course equal to the encryption process. Deciphering is performed in the coding area, but in reverse order, only.

1. Analysing ciphertext
   ciphertext → cipher alphabet (0...127) → 7 bit index values
2. bit conversion
   7 bit index values (0 ...127) → 8-bit XOR concatenation
3. XOR concatenation
   8-bit XOR concatenation → block key → plaintext block

The program searches in blocks of **48** cipher characters the decimal index values of single characters in an identical created cipher alphabet (system alphabet) and connects them to a series of 336 bits. This series will be devided into **42** 8-bit sequences (336 bits) in bitsystem on base 8 and concatenate with the respective block key. The original plaintext becomes visible.

As an example deciphering the ciphertext file **Museum.ctx:**

At start the program reads the first ciphertext block with 48 characters:

[™N“M‘i¦eÀ•}R7tÁ»}àÀCnûÂž.J4eŸªwM□oàØázb.»o*ƒ*Ì□□i

Index:	106	21	54	78	26	72	1	63	8
(-1)	105	20	53	77	25	71	0	62	7

base 7:
1101001 0010100 0110101 1001101 0011001 1000111 0000000 0111110 0000111 . . .
conversion to base 8:
11010010 01010001 10101100 11010011 00110001 11000000 00111110 00001110 . . .

block key base 8:
10000110 00111001 11001001 11110011 01000010 10100011 01010111 01101011 . . .
XOR concatenation base 8:
01010100 01101000 01100101 00100000 01110011 01100011 01101001 01100101 . . .
Index base 8:

	84	104	101	32	115	99	105	101
ASCII:	T	h	e	□	s	c	i	e

As plaintext results: **The scie**nce of bird . . . .

6 Cryptanalysis

As is known cryptanalysis encompasses all attempts to draw any conclusion out of the ciphertext intended to recovery the plaintext. To noticeable events of any language belong statistically repetition patterns and word combinations, frequency structures and two-digit-groups and bigramms [#5]. But this circumstances require that plaintext and ciphertext are in a ratio of 1:1, ciphertext should be as long as the plaintext "Congruence of Length" [#5]. The most known attacks are analysing structures, „known plaintext attack“ and „chosen plaintext attack“, possibly still „differential“ and „linear“ analysis [#6]. By this attacks it is intended to find conspicuous connections in order to discover a way to the plaintext, possibly. To analyse this features it is necessary that ciphertext and plaintext share certain structures which can be compared really. Therefore a unit order system must exist which works in plaintext and ciphertext as well.

6.1 Current procedures

If ciphertext and plaintext have an equal length it follows that there must exist for each single plaintext character a specific ciphertext character. Both areas – input and output - work within the same system alphabet. Therefore a uniform order system exist and no system changes. Particularities in plaintext must be effective in ciphertext as well and might be found.

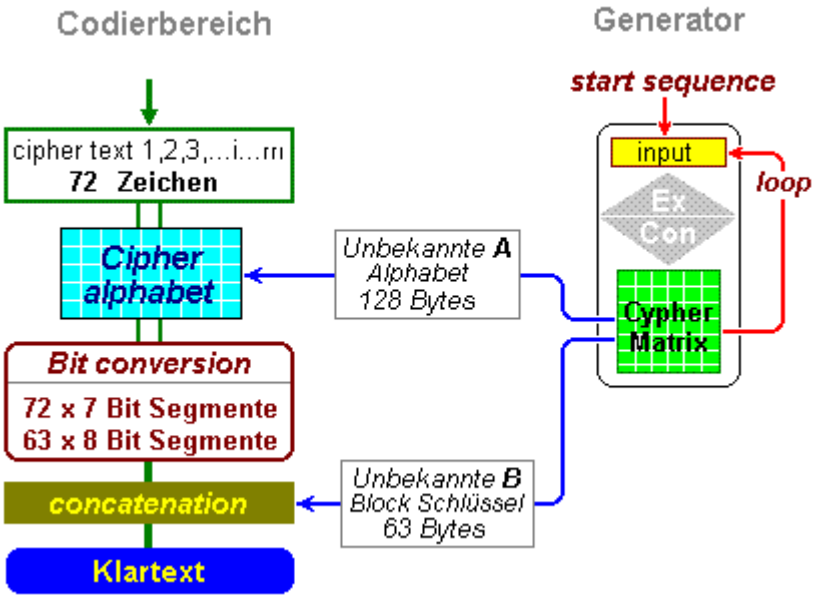
6.2 „CypherMatrix“ procedure

This conditions are not implemented in CypherMatrix procedures. System changing occurs: conversion of plaintext characters in bitsystem on base 8 to ciphertext characters in bitsystem on base 7. Both areas cannot be compared any longer. Congruence of length is missing. Due to conversion for each plaintext character there is a ciphertext character, which is by factor **1,143** longer than the causative plaintext character. Besides the independent ciphertext alphabet contains 128 elements, only and the plaintext alphabet comprises 256 characters. A united order system for both areas is missing and by this the basis for all special attacks mentioned above. They fall into abeyance and we may forget them. In order to test this you may download some with CypherMatrix programs encrypted messages as ZIP-file and try to break the cipher with todays usual attacks [#7].

6.3 Security of the procedure

At least there is still the possibility to break the cipher by a „brute force attack“. An attacker in principle knows the ciphertext and the CypherMatrix procedure. But the respective program and single control parameters, including start sequence he does not know.
He may try an iteration of all possibilities. An attack on the start sequence with **42 bytes** length results to an entropie of **336** and an exponential complexity of **O(2^336) = 1.4E-101**
An attacker using the ciphertext may then try to find single parts of encryprion steps. But there are no starting points which give some expectation for success.

Decryption happens in each round as follows:



The procedure includes three functions:

1. Plaintext block --> **block key** --> -8 bit XOR sequences
2. 8-bit XOR sequences --> 7-bit index values
3. 7-bit index values --> **chipher alphabet (128)** --> ciphertext

In this functions the parameters **block key** and **cipher alphabet** are two variables independent from each other. Effective are:

cm = f [f1 (an, **k1**), f2 (b1, b2), f3 (b2, **k2**)]
an = f [f3 (cm, **k2**), f2 (b2, b1), f1 (b1, **k1**)]

fx = functional connection
an = plaintext
k1 = **block key**
b1 = 8-bit sequence
b2 = 7-bit index value
k2 = **cipher alphabet (128)**
cm = ciphertext

Ascertaining the ciphertext (**cm**) and retrograde searching for the plaintext (**an**) points out as an equation with two unknown veriables: **k1** und **k2.** That leads to a definite solution, only, if one unknown variable can be derived from the other unknown variable or if there are two equations with the same unknown variables.

But between the respective block key = k1 and the cipher alphabet (128) = k2 generated in the same round there are no connections. Nevertheless, even if both are extracted out of the current CypherMatrix yet they have no functional connection: (**k1** --> (**Hk MOD 169**)**+1**) and **k2** --> (**Hk +Hp**) **MOD 255**)**+1**). The current CypherMatrix itself is derivated from the initial start sequence only. But thereunto is no way back (two one-way-functions prevent this). Hence, there are many couples of cipher alphabets / block keys which result from an „brute force“ attack and deliver any legible texts, but one does not know which is the right one: Angriff mit "brute force" . Thus „brute force“ cannot have success, too.

7 Summary

The foregoing explanations show some alternative principles:

1. An unity order system of bit series (structure and system alphabets) are not defined up to now,
2. Description of cryptography are confined to coding in bitsystem on base 8 (input and output) and
3. almost all attacking scenarios presuppose a comparable order system for plaintext and ciphertext (bitsystem on base 8 and system alphabet with 256 characters).

More details of **CypherMatrix** procedures under: www.telecypher.net/ [#8]
Who wants to deal with the procedures more intensively may request single programs – with or without source code – from the author per e-mail and work with them under the CMLizenz **.** (eschnoor@multi-matrix.de).

Munich, in Febrary 2013

Notes

[#1] Paradigmenwechsel in der Kryptographie, www.telecypher.net/ParadigmaDe.pdf
[#2] Schneier, Bruce, Angewandte Kryptographie (dt. Ausgabe), Bonn … 1996, S.229
[#3] Basisfunktion.pdf, www.telecypher.net/Basisfunktion.pdf
[#4] Esslinger, Bernhard, Uni Siegen, http://www.cryptool.org/de/
[#5] Strukturvergleich Klartext und Geheimtext, www.telecypher.net/Equilang.pdf
[#6] Bauer, Friedrich L., Entzifferte Geheimnisse – Methoden und Maximen der Kryptologie. Berlin Heidelberg New York, 1995, S. 186 ff.,
[#7] Download, www.telecypher.net/ZUSENDEN.HTM
[#8] Der Kern des CypherMatrix Verfahrens, www.telecypher.net/CYPHKERN.HTM
[#9] Schmeh, Klaus, Safer Net, Kryptografie im Internet und Intranet, Heidelberg 1998, S.61.