Date: Monday, September 16, 2002
Prof.: Dr Jean-Yves Chouinard

Design of Secure Computer Systems CSI4138/CEG4394

Notes on Classic encipherment methods

1 Simple substitution ciphers

The message encipherment is done by applying the transformation algorithm $E(\bullet)$ with key K to the plaintext message M:

Encryption transformation:

$$C = E_K(M)$$

where C is the resulting ciphertext.

The decryption operation is performed on the ciphertext, using the same key K in conjunction with the decryption transformation $D_K(\bullet)$ on the received ciphertext data.

Decryption transformation:

$$M = D_K(C) = D_K[E_K(M)]$$

The message string M consists in a string of single plaintext letters, or symbols, $M = m_1$, m_2, ..., m_i, ... taken from a plaintext sample space $m_i \in \mathcal{M}$, the key string K is also a string of symbols $K = k_1$, k_2, ..., k_i, ... from a key sample space $k_i \in \mathcal{K}$, and the encrypted ciphertext string $C = c_1$, c_2, ..., c_i, ... where each ciphertext symbol $c_i \in \mathcal{C}$, the ciphertext sample space.

1.1 Monoalphabetic substitution ciphers

Consider the plaintext alphabet $\mathcal{M}$ (or sample space):

$$\mathcal{M} \equiv \{a_1, \ldots, a_n\} \quad \text{where } n \text{ is the alphabet size}$$

The encryption transformation $E_K(\bullet)$ can be viewed as a single mapping function $f(\bullet)$ from the plaintext alphabet $\mathcal{M}$ to the ciphertext alphabet $\mathcal{C}$:

$$f(\bullet) : \mathcal{M} \rightarrow \mathcal{C}$$

for which $\mathcal{C} \equiv \{f(a_1), \ldots, f(a_n)\}$. If the message string M is:

$$M = m_1, m_2, \ldots, m_i, \ldots$$

then the corresponding ciphertext string C is:

$$C = E_K(M) = f(m_1), f(m_2), \ldots, f(m_i), \ldots$$

Example*(Caesar's cipher):* A very simple monoalphabetic substitution cipher is the Julius Caesar's cipher. The transformation algorithm $E_K(\bullet)$ is: *"replace each letter in the plaintext by the third one following it in the standard alphabet"*, whereas the key k is simply the amount of "shift" between the original plaintext letters and the ciphertext letters. It is called a *shifted-alphabet cipher*. Assume that $k = 3$, for instance.

Encryption algorithm:

$$f(i) = (i + k) \bmod n$$

where $k = 3$ and $n = 26$. $f(i)$ represents the letter index in the ciphertext sample space $\mathcal{C}$.

If Caesar wanted to encipher the plaintext message M:

$$M = \text{“}brutus''$$

then the ciphertext C would be:

$$C \quad = \quad E_3(\text{“}brutus'') = f(b), f(r), f(u), f(t), f(u), f(s) = f(1), f(17), f(20), f(19), f(20), f(18)$$

in terms of indexes. Since

$$\begin{array}{lll} f(1) & = (1+3) \bmod 26 & = 4 \\ f(17) & = (17+3) \bmod 26 & = 20 \\ f(20) & = (20+3) \bmod 26 & = 23 \\ f(19) & = (19+3) \bmod 26 & = 22 \\ f(20) & = (20+3) \bmod 26 & = 23 \\ f(18) & = (18+3) \bmod 26 & = 21 \end{array}$$

the ciphertext is:

$$C \quad = \quad 4, 20, 23, 22, 23, 21 = \text{“}EUXWXV''$$

Decryption algorithm:

The legitimate message recipient, having the encryption key k and knowing the encryption transformation (i.e. shifted-alphabet cipher transformation) can perform the decipherment of ciphertext C:

$$f^{-1}(i) = (i - k) \bmod n$$

$$\begin{array}{lll} D_3(C) & = & D_3[E_3(\text{“}brutus'')] = D_3(\text{“}EUXWXV'') \\ D_3(C) & = & \text{“}brutus'' \end{array}$$

Cryptanalysis of Caesar's cipher (2 cases):

1. The cryptanalyst don't know that the ciphertext C is a *shifted-alphabet cipher*. He computes from C the *relative frequencies of occurrence* of the ciphertext letters (see figure 2). If there is a sufficient amount of ciphertext, the ciphertext letter distribution should approach that of the plaintext alphabet (i.e. figure 1). By comparing both distributions it is obvious that C is a shifted-alphabet cipher and that the amount of "shifting" is 3.

2. The cryptanalyst already suspects that it is a shifted alphabet cipher. He can then try all the 25 possible keys $1 \leq k \leq 25$ on the ciphertext until he obtains a meaningful message (exhaustive key search method).

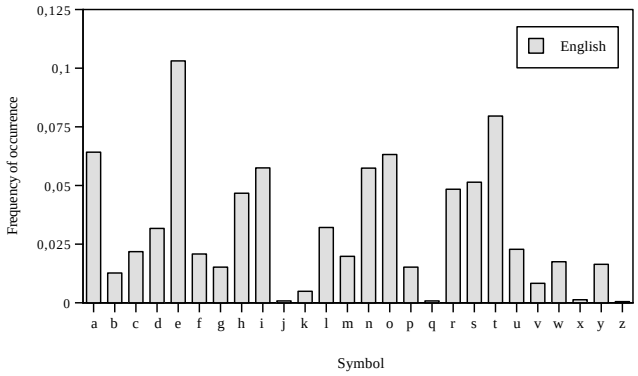


Figure 1: Letter distribution of English plaintext.

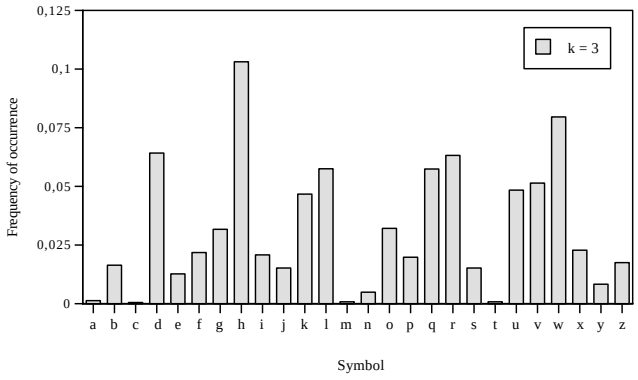


Figure 2: Letter distribution of Caesar's shifted alphabet cipher ($k = 3$).

This example shows that a simple shifted alphabet cipher is extremely weak in preserving the secrecy of a message M.

1.2 Other simple substitution ciphers

Multiplication cipher:

For multiplication ciphers, the encryption transformation is given by:

$$f(i) = (i \times k) \bmod n$$

where k and n must be relatively prime, that is $\gcd(k, n) = 1$. For instance if $k = 3$ then:

$$f(i) = (i \times 3) \bmod n$$

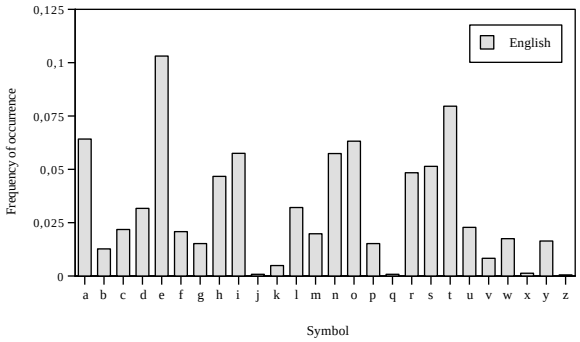


Figure 3: Letter distribution of English plaintext.

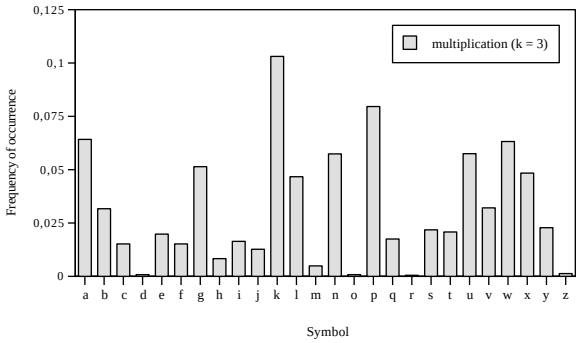


Figure 4: Letter distribution of multiplication cipher ($k = 3$).

Affine transformation:

The encryption transformation combines the shifted-alphabet and the multiplication encryption transformations:

$$f(i) = (i \times k_1 + k_0) \bmod n$$

Table 1: Multiplication cipher.

M:	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
$i =$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$f(i) =$	0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2	5	8	11	14	17	20	23
C:	A	D	G	J	M	P	S	V	Y	B	E	H	K	N	Q	T	W	Z	C	F	I	L	O	R	U	X

Again, k_1 and n must be relatively prime.

Polynomial transformation:

The encryption transformation is a generalization of the affine transformation:

$$f(i) = (i^t k_t + i^{t-1} k_{t-1} + \cdots + i^2 k_2 + i k_1 + k_0) \bmod n$$

$t = 1 \Longrightarrow$ Affine transformation cipher

$t = 0 \Longrightarrow$ Shifted-alphabet cipher

General case (monoalphabetic substitution ciphers):

$$f(a_i) \neq f(a_j) \qquad \text{for all } i \neq j$$

For instance,

$$\begin{aligned} \mathcal{M} &= \{a, b, c, \cdots, x, y, z\} \\ \mathcal{C} &= \{H, X, N, \cdots, A, D, J\} \end{aligned}$$

Note that for the general case, an exhaustive key search over a 26 letter alphabet may prove computationally infeasible: there are $n = 26$ possible choices for the first letter, $(n-1) = 25$ choices for the second, $(n-2) = 24$ the third, and so on... In other words there are $n! = 26! \approx 4 \times 10^{26}$ choices of alphabet permutations. However, looking at the distribution of ciphertext letters, it is fairly simple to determine the encryption mapping function $f(\bullet)$ (assuming sufficient amount of ciphertext). Note that for the above schemes there is a single $f(\bullet)$ function mapping:

$$\begin{aligned} M &\Longrightarrow C \\ i &\Longrightarrow f(i) \end{aligned}$$

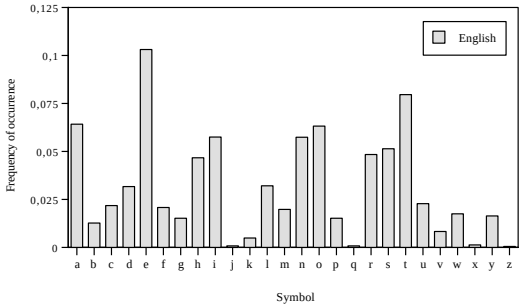


Figure 5: Letter distribution of English plaintext.

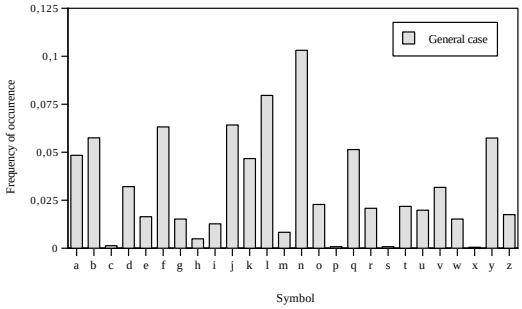


Figure 6: Letter distribution of monoalphabetic substitution cipher (general case).

2 Polyalphabetic substitution ciphers

For polyalphabetic substitution ciphers, the message sequence $M = m_1, \ldots, m_i, \ldots$ is encrypted by applying the transformation algorithm $E(\bullet)$ with a key sequence $K = k_1, \ldots, k_i, \ldots$ to the plaintext message M:

Encryption transformation:

$$C = E_K(M)$$

where $C = c_1, \ldots, c_i, \ldots$ is the resulting ciphertext sequence. The ciphertext sequence decryption is done using the same key sequence $k_1, \ldots, k_i, \ldots$ with the proper decryption transformation $D_K(\bullet)$ on the received ciphertext data:

Decryption transformation:

$$M = D_K(C) = D_K[E_K(M)]$$

There are three types of polyalphabetic ciphers, these are:

1. Periodic substitution ciphers
2. Running-key ciphers
3. One-time pad ciphers

2.1 Periodic substitution ciphers

For this type of polyalphabetic substitution cipher, the key sequence, or key stream, is repeated after a period of d plaintext symbols. The encryption transformation can be expressed as a set of d mapping functions corresponding to the d different keys $K = k_1, \ldots, k_d$:

$$f_i : \mathcal{M} \to \mathcal{C}_i, \quad \text{for } 1 \leq i \leq d$$

$$\begin{aligned}
\mathcal{M} &= \{a, b, c, \ldots, x, y, z\} \qquad \text{(plaintext set)} \\
\mathcal{C}_1 &= \{V, G, D, \ldots, C, X, E\} \qquad \text{(monoalphabetic \#1)} \\
&\vdots \\
\mathcal{C}_i &= \{E, Q, S, \ldots, H, V, T\} \qquad \text{(monoalphabetic \#}i\text{)} \\
&\vdots \\
\mathcal{C}_d &= \{T, B, N, \ldots, P, G, W\} \qquad \text{(monoalphabetic \#}d\text{)}
\end{aligned}$$

Therefore, for a message sequence M given by:

$$M = m_1, m_2, \ldots, m_{d-1}, m_d, m_{d+1}, \ldots, m_{2d-1}, m_{2d}, m_{2d+1}, \ldots$$

the corresponding ciphertext sequence C is derived from the key stream K, with period d, as:

$$\begin{aligned}
C &= E_K(M) \\
C &= f_1(m_1), \ldots, f_{d-1}(m_{d-1}), f_d(m_d), f_1(m_{d+1}), \ldots, f_{d-1}(m_{2d-1}), f_d(m_{2d}), f_1(m_{2d+1}), \ldots
\end{aligned}$$

2.2 Vigenère ciphers

The Vigenère polyalphabetic cipher is a periodic shifted-alphabet substitution cipher. It was designed by Blaise de Vigenère in the 16th century. The encryption transformation is based on the simple shifted-alphabet substitution used for Caesar's cipher, except that the key is changed for each plaintext letter over a period of d letters. There are d mapping functions:

$$f_i(m) = (m + k_i) \bmod n \qquad \text{for } i = 1 \text{ to } d$$

the inverse decryption transformation simply consists in removing the "alphabet shift" from each ciphertext symbol:

$$f_i^{-1}(c) = (c - k_i) \bmod n \qquad \text{for } i = 1 \text{ to } d$$

Example*(Vigenère cipher):*

M	=	p	e	r	i	o	d	i	c	s	h	i	f	t	e	d	a	l	p	h	a	b	e	t
K	=	v	i	g	e	n	e	r	e	v	i	g	e	n	e	r	e	v	i	g	e	n	e	r
C	=	K	M	X	M	B	H	Z	G	N	P	O	J	G	I	U	A	D	X	N	A	O	I	K
M	=	i	c	s	u	b	s	t	i	t	u	t	i	o	n	c	i	p	h	e	r			
K	=	e	v	i	g	e	n	e	r	e	v	i	g	e	n	e	r	e	v	i	g			
C	=	M	X	A	A	F	F	X	Z	X	P	B	O	S	A	G	Z	T	C	M	X			

Here the period of the key stream is $d = 8$. To encrypt using a Vigenère table, for each plaintext symbol m_i, choose the row k_i and find the ciphertext symbol c_i in the column m_i. To decrypt find the ciphertext symbol c_i in the row k_i and decrypt it as the plaintext character at the top of this particular column.

Table 2: Vigenère table.

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
		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
0	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
1	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
2	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
3	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
4	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
5	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
6	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
7	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
8	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
9	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
10	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
11	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
12	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
13	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
14	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
15	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
16	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
17	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
18	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
19	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
20	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
21	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
22	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
23	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
24	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
25	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

2.3 Cryptanalysis of polyalphabetic substitution ciphers

The ciphertext C consists of a string of ciphertext elements:

$$C = c_1, c_2, \ldots, c_i, \ldots$$

1. Determine the period d of the polyalphabetic cipher, if it is possible.
2. Form the following d sub-sequences from the ciphertext stream:

$$\begin{array}{rcccccc} s_1 & = & c_1, & c_{d+1}, & c_{2d+1}, & \ldots \\ s_2 & = & c_2, & c_{d+2}, & c_{2d+2}, & \ldots \\ & \vdots & & & & \\ s_d & = & c_d, & c_{2d}, & c_{3d}, & \ldots \end{array}$$

3. Perform the frequency analysis of each sub-sequence (as for a monoalphabetic cipher), and try to determine the d different substitution mappings:

$$\begin{array}{rcl} \mathcal{M} & \Longrightarrow & \mathcal{C}_1 \\ \mathcal{M} & \Longrightarrow & \mathcal{C}_2 \\ & \vdots & \\ \mathcal{M} & \Longrightarrow & \mathcal{C}_d \end{array}$$

and recover the plaintext.

2.3.1 Measure of roughness

From an amount of collected ciphertext, the cryptanalyst can evaluate the *roughness* of the ciphertext symbol distribution and estimate the period d of the polyalphabetic cipher, if it is periodic at all. The measure of roughness MR [Den82] is defined as:

$$\boxed{MR = \sum_{i=0}^{n-1} \left(p_i - \frac{1}{n} \right)^2}$$

where p_i is the probability that a ciphertext symbol is the symbol a_i in the alphabet (e.g. p_0: probability that ciphertext letter is a). MR is in fact the variance of the distribution.

If the source is equiprobable, then $p_i = \frac{1}{n}$ for all i and then the measure of roughness is:

$$MR = \sum_{i=0}^{n-1} \left(\frac{1}{n} - \frac{1}{n} \right)^2 = 0$$

which indicates that the distribution is flat or uniform and its variance equals zero.

If the source is not equiprobable, then $MR \neq 0$. For English text the measure of roughness can be computed as:

$$\begin{aligned}
MR &= \sum_{i=0}^{25} \left(p_i - \frac{1}{26} \right)^2 = \sum_{i=0}^{25} \left[p_i^2 - 2p_i \frac{1}{26} + \left(\frac{1}{26} \right)^2 \right] \\
MR &= \sum_{i=0}^{25} p_i^2 - \frac{1}{13} \sum_{i=0}^{25} p_i + \sum_{i=0}^{25} \left(\frac{1}{26} \right)^2 = \sum_{i=0}^{25} p_i^2 - 0.0769 + 0.0385 \\
MR &= \sum_{i=0}^{25} p_i^2 - 0.0385 = 0.0680 - 0.0385 \\
MR &= 0.0295
\end{aligned}$$

or equivalently:

$$MR + 0.0385 = \sum_{i=0}^{25} p_i^2 = 0.0680$$

for English text. The probability $p_i^2 = p_i \times p_i$ represents in fact the probability that 2 letters from the ciphertext C are a_i and the sum:

$$\sum_{i=0}^{n-1} p_i^2$$

gives the probability that 2 letters from the ciphertext are the same (for any value of i).

2.3.2 Index of coincidence

Consider a ciphertext stream of length L:

$$C = c_1, c_2, \ldots, c_L$$

1. The total number of possible ciphertext pairs (c_i, c_j) is:

$$\binom{L}{2} = \frac{L!}{(L-2)!\ 2!} = \frac{L\ (L-1)}{2}$$

2. The number of pairs containing only the letter c_i, that is (c_i, c_i) is:

$$\frac{F_i\ (F_i - 1)}{2}$$

   where F_i is the number of occurrences (i.e. an integer number) of ciphertext letter c_i in the block of L ciphertext symbols, in other words,

$$\sum_{i=0}^{n-1} F_i = L$$

3. The total number of identical ciphertext pairs (c_i, c_i), i.e. for all i, equals:

$$\sum_{i=0}^{n-1} \frac{F_i\ (F_i - 1)}{2}$$

4. The index of coincidence IC of the ciphertext C provides an estimate of the measure of roughness MR by considering the probability that two letters chosen at random in a ciphertext are identical, $\Longrightarrow (c_i, c_i)$:

$$\begin{aligned} IC &= \frac{\text{number of identical pairs}}{\text{total number of possible pairs}} \\ IC &= \frac{\sum_{i=0}^{n-1} \frac{F_i\ (F_i-1)}{2}}{\frac{L\ (L-1)}{2}} \\ IC &= \frac{\sum_{i=0}^{n-1} F_i\ (F_i - 1)}{L\ (L-1)} \end{aligned}$$

By computing the index of coincidence of a sufficient amount of ciphertext, the cryptanalyst can determine approximately the period of the polyalphabetic cipher, as long as d is relatively small. Table 3 (from [SJP89]) indicates the index of coincidence as a function of the cipher period d (and also for different languages when $d = 1$), while figure 7 provides a computer program to compute the index of coincidence of a ciphertext.

Table 3: Indices of coincidence (from Seberry).

Period d	Index IC
1	0.0669
2	0.0520
3	0.0473
4	0.0450
5	0.0436
6	0.0427
7	0.0420
8	0.0415
9	0.0411
10	0.0408
11	0.0405
12	0.0403
13	0.0402
14	0.0400
15	0.0399
16	0.0397
17	0.0396
18	0.0396
19	0.0395
20	0.0394

Language	Index IC
Arabic	0.075889
Danish	0.070731
Dutch	0.079805
English	0.066895
Finnish	0.073796
French	0.074604
German	0.076667
Greek	0.069165
Hebrew	0.076844
Italian	0.073294
Japanese	0.077236
Malay	0.085286
Norwegian	0.069428
Portuguese	0.074528
Russian	0.056074
Serbo Croatian	0.064363
Spanish	0.076613
Swedish	0.064489

```
 program IC (input,
output); const
  N = 100000;
  high = 0.066;
  low =  0.038;
var
  d : integer;
  ic: real;
begin
  writeln('    d                  IC');
  writeln(' ======================');
  for d :=  1 to 20 do
  begin
    ic := (1/d)*(N-d)/(N-1)*high+((d-1)/d)*(N/(N-1))*low;
    writeln(d:6, '                    ',ic:1:4);
  end
end
```

Figure 7: Program to compute the index of coincidence (from Seberry [SJP89]).

Example*(Cryptanalysis for a polyalphabetic substitution cipher):*

The ciphertext C on figure 8 consists of $L = 346$ ciphertext symbols [Den82].

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

Figure 8: Example of a polyalphabetic substitution cipher (from Denning).

To estimate the period d, the relative frequencies $\{F_i\}$ of ciphertext symbols are computed (figure 9):

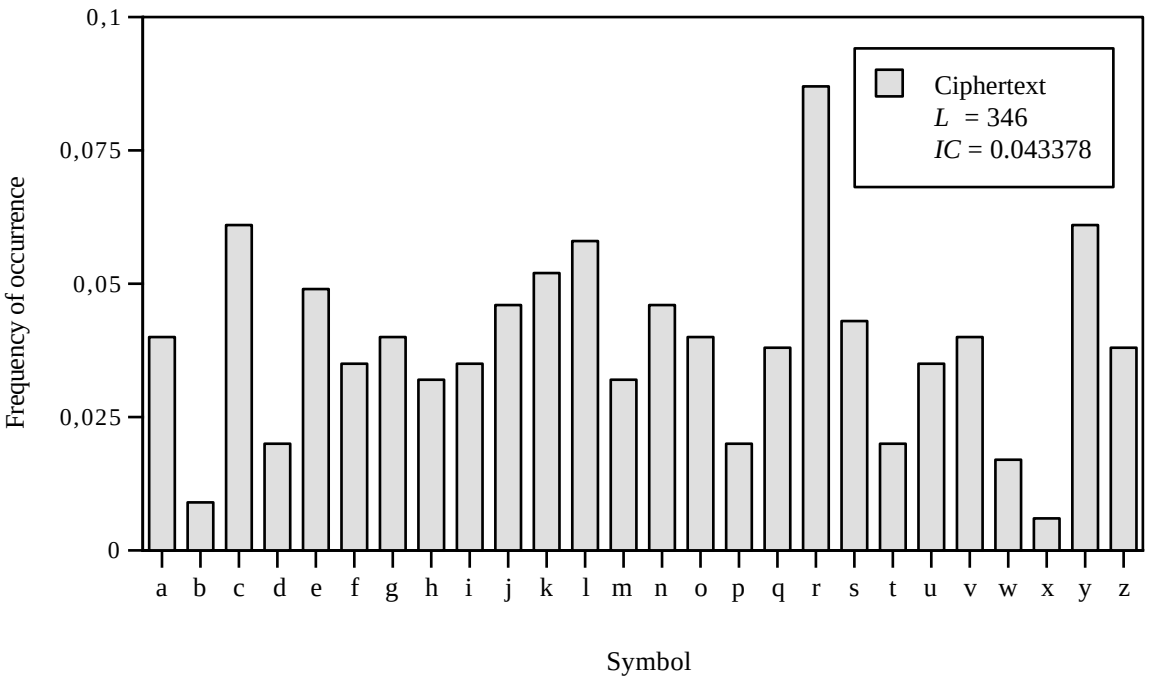


Figure 9: Ciphertext symbol distribution: sequence length $L = 346$ and the index of coincidence $IC = 0.043378$.

$$
\begin{array}{lll}
F_0 & = 14 \rightarrow \frac{14}{346} & \approx 4.04\% \qquad \text{occurrences of } \text{“}A''\\
F_1 & = 3 \rightarrow \frac{3}{346} & \approx 0.87\% \qquad \text{occurrences of } \text{“}B''\\
 & \vdots & \\
F_{25} & = 13 \rightarrow \frac{13}{346} & \approx 3.76\% \qquad \text{occurrences of } \text{“}Z''
\end{array}
$$

The index of coincidence IC is then:

$$
\begin{array}{rcl}
IC & = & \dfrac{\sum_{i=0}^{n-1} F_i\ (F_i - 1)}{L\ (L-1)} \\
 & = & \dfrac{\sum_{i=0}^{n-1} F_i\ (F_i - 1)}{346 \times 345} \\
IC & \approx & 0.0434
\end{array}
$$

From table 3 the cryptanalyst determines that the period d of the cipher C is about $d \approx 5$, and then does the frequency analysis with the 5 sub-sequences.

The exact period can also be determined with the *Kasiski* method (Kasiski, 1863, prussian military officer). By finding identical ciphertext blocks in C, resulting from the substitution of identical plaintext with identical key, the period d is deducted.

For instance, the sequence *"QGOLKALVOS"* appears three times in the ciphertext, that is at location 90, 141 and 213. The offsets between these *chunks* of identical ciphertext are:

$$
\begin{array}{lll}
141 - 90 = 51 & \Longrightarrow & \text{divisors of 51: } (3, 17) \\
213 - 141 = 72 & \Longrightarrow & \text{divisors of 72: } (2, 3, 4, 6, 8, 9, 12, 18, 24, 36)
\end{array}
$$

The cryptanalyst finds out that 3 is the only common divisor of both offsets, i.e. 51 and 72, and then estimates *safely* the cipher period $d = 3$ (instead of 5 from the index of coincidence IC). He then proceeds by forming 3 sub-sequences (figure 10) and analyzing the distributions in each sub-sequence:

$$
\begin{array}{rcl}
s_1 & = & c_1,\ \ c_4,\ \ c_7,\ \ \ldots \\
s_2 & = & c_2,\ \ c_5,\ \ c_8,\ \ \ldots \\
s_3 & = & c_3,\ \ c_6,\ \ c_9,\ \ \ldots
\end{array}
$$

The indices of coincidence IC_1, IC_2, and IC_3 are close to the index of coincidence of a monoalphabetic cipher ($IC \approx 0.0669$). He can then decrypt the three sub-sequences by comparing their distributions to that of English text.

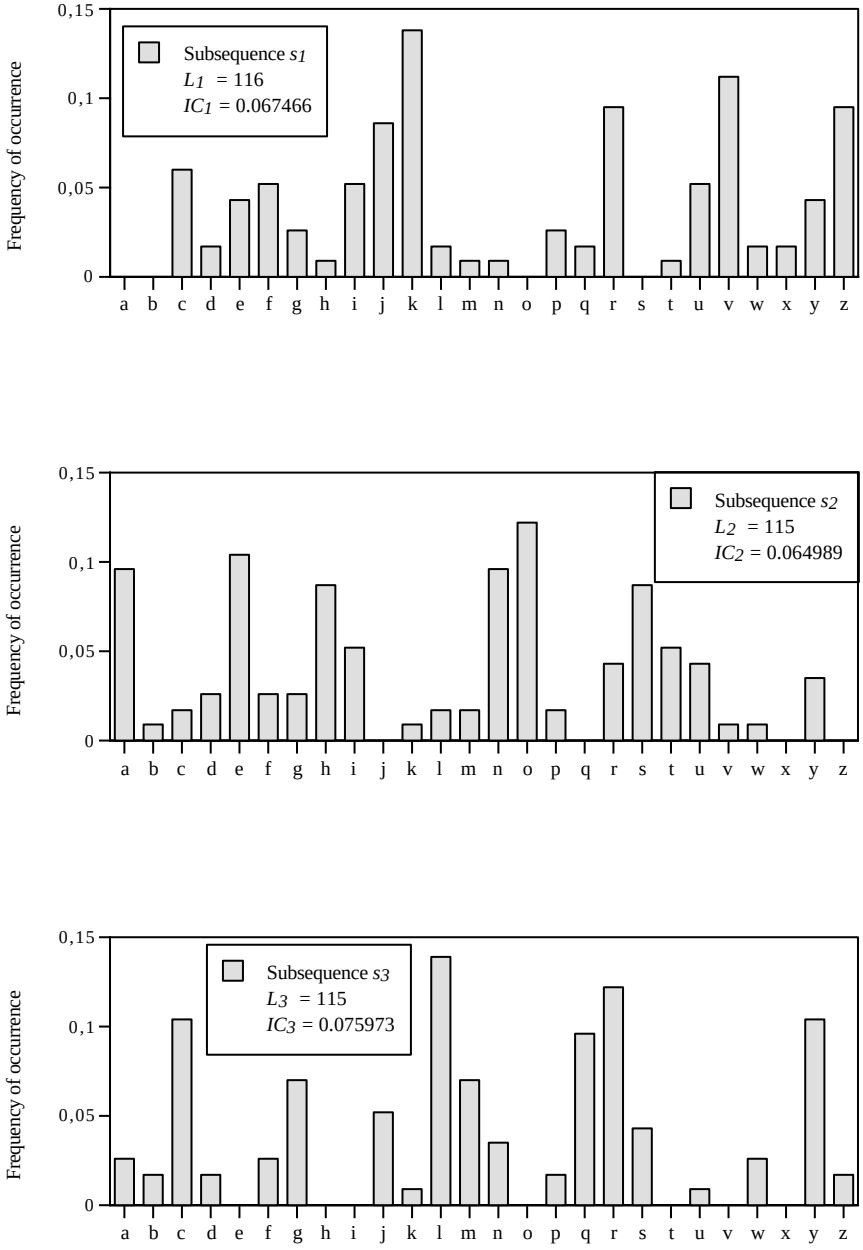


Figure 10: Ciphertext symbol distribution in the three sub-sequences (from Denning): $L_1 = 116$ and $IC_1 = 0.067466$, $L_2 = 115$ and $IC_2 = 0.064989$, $L_3 = 115$ and $IC_3 = 0.075973$.

3 Non-periodic substitution ciphers

3.1 Running-key ciphers

Running-key ciphers are polyalphabetic substitution ciphers which are *non-periodic* (i.e. non-repetitive), or for which the key stream period d is longer than the plaintext message. Historically, to make the use of such ciphers relatively simple, the elements of the key stream were typically taken from a text in a given book at a given page starting at a given line and character; the security depending on the secrecy of the text.

Example*(Running-key cipher):*
The following running-key cipher is based on shifted-alphabet substitutions [Den82].

$$\begin{array}{lllllllllllllllllll}
M & = & t & h & e & t & r & e & a & s & u & r & e & i & s & b & u & r & i & e & d & \ldots \\
K & = & t & h & e & s & e & c & o & n & d & c & i & p & h & e & r & i & s & a & n & \ldots \\
C & = & M & O & I & L & V & G & O & F & X & T & M & X & Z & F & L & Z & A & E & Q & \ldots
\end{array}$$

This is a polyalphabetic (shifted-alphabet substitution) cipher for which there are as many keys elements k_i as there are plaintext symbols.

The key source is English text which as a non-zero redundancy. However, since the key source (taken from the $\mathcal{K}$ sample space) is not an equiprobable source, the cipher is breakable: some ciphertexts symbols will be more likely to happen than others.

Friedman [Den82] (circa 1918) proposed a cryptanalysis method based on the relative frequencies. Many ciphertext symbols $\{c_i\}$ are the result of enciphering a *high frequency* plaintext letter m_i with another *high frequency* key letter k_i (which is not random if English text is used as the running-key) (see figure 11 and table 4 below).

Out of the 19 (m_i, k_i) plaintext-key pairs there are 12 such pairs of high frequency letters:

$$\begin{array}{lllllllllllllllllll}
\{m_i\} & = & t & h & e & t & r & e & a & s & u & r & e & i & s & b & u & r & i & e & d & \ldots \\
\{k_i\} & = & t & h & e & s & e & c & o & n & d & c & i & p & h & e & r & i & s & a & n & \ldots \\
 & & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & & \uparrow & \uparrow & & & \uparrow & & \uparrow & & & \uparrow & \uparrow & \uparrow & & \ldots \\
 & = & 1 & 2 & 3 & 4 & 5 & & 6 & 7 & & & 8 & & 9 & & & 10 & 11 & 12 & & \ldots
\end{array}$$

Since it is a shifted-alphabet cipher, the Vigenère table can be used. What are the high frequency pairs (m_i, k_i) that produces the ciphertext symbol c_i?

For the ciphertext beginning with $C =$ "$MOI\ldots$",

$$\begin{array}{lllllllllllll}
c_1 & = & M & \Longrightarrow & e & i & & i & e & & t & t & \\
c_2 & = & O & \Longrightarrow & a & o & & o & a & & h & h & \\
c_3 & = & I & \Longrightarrow & a & i & & i & a & & e & e & \qquad r \quad r \\
 & & & & \uparrow & \uparrow & & \uparrow & \uparrow & & \uparrow & \uparrow & \qquad \uparrow \quad \uparrow \\
 & & & & m_i & k_i & & m_i & k_i & & m_i & k_i & \qquad m_i \quad k_i
\end{array}$$

Now let's consider the likely trigram combinations that have been used to produce the ciphertext $C =$ "$MOI\ldots$":

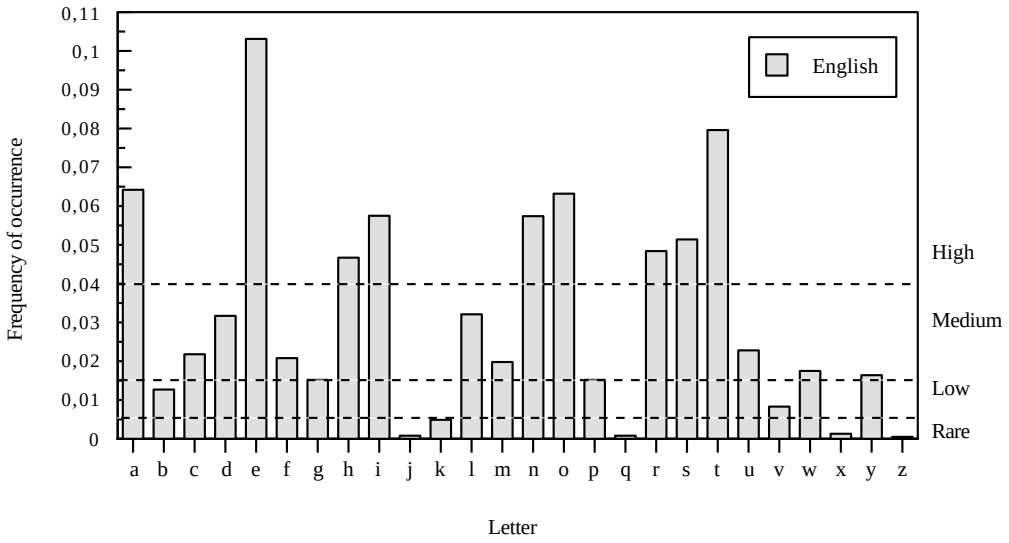


Figure 11: Frequency partitioning of English letters.

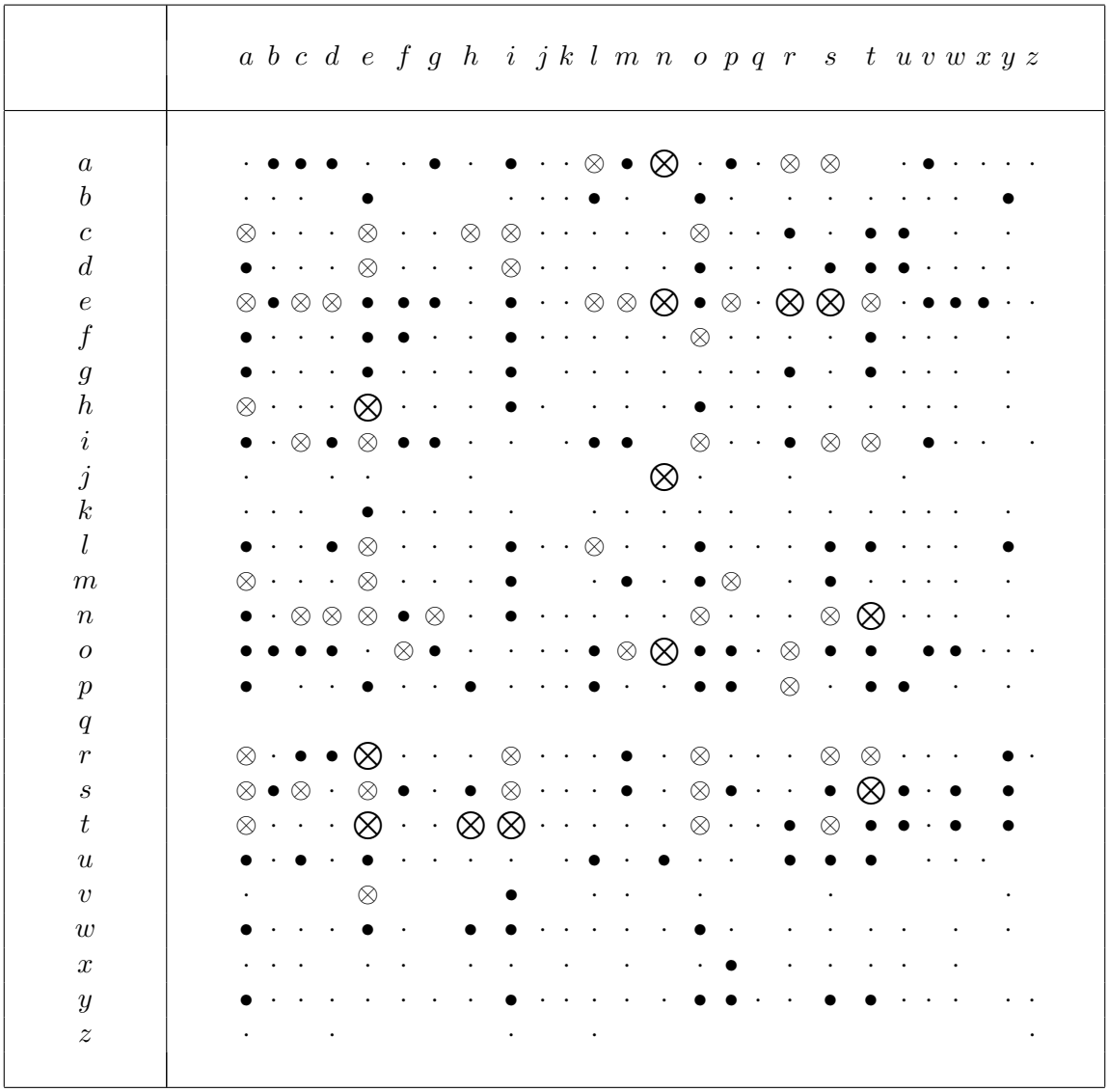
Table 4: Frequency partitioning of English letters.

Probability	Letters
High	$e \quad t \quad a \quad o \quad n \quad i \quad r \quad s \quad h$
Medium	$d \quad l \quad u \quad c \quad m$
Low	$p \quad f \quad y \quad w \quad g \quad b \quad v$
Rare	$j \quad k \quad q \quad x \quad z$

M	O	I	M	O	I	M	O	I	$\ldots$	M	O	I	$\ldots$	M	O	I
e	a	a	e	a	i	e	o	a	$\ldots$	t	h	e	$\ldots$	t	h	r
i	o	i	i	o	a	i	a	i	$\ldots$	t	h	e	$\ldots$	t	h	r

Compare all these trigrams with the meaningful and likely trigrams in English. From these the cryptanalyst keeps the likely trigram "*the*" as the text ($m_i, \quad i = 1, \ldots, 3$) and also for the key ($k_i, \quad i = 1, \ldots, 3$).

Table 5: Frequency of digrams in English text (from Denning [Den82]).



⊗ : High (more than 1.15% of the digrams)

⊗ : Medium (more than 0.46% of the digrams)

• : Low (more than 0.12% of the digrams)

· : Rare (more than 0.10% of the digrams)

□ : No occurrences

3.2 Vernam cipher

Vernam ciphers, or *one time pad* cipher, is another running-key cipher, which has the property of being unconditionally secure. The key stream K is purely random and is never repeated, or used again, hence the name *"one time pad"*.

$$\begin{array}{lcccccc} M & = & m_1 & m_2 & \dots & m_n & \dots \\ K & = & k_1 & k_2 & \dots & k_n & \dots \\ C & = & f_{k_1}(m_1) & f_{k_2}(m_2) & \dots & f_{k_n}(m_n) & \dots \end{array} \quad \text{and}$$

where the key sequence is a non-repeating random sequence. For instance, as shown on figure 12, the plaintext C, the key stream K and the ciphertext can be binary strings defined as:

$$c_i = m_i \oplus k_i$$

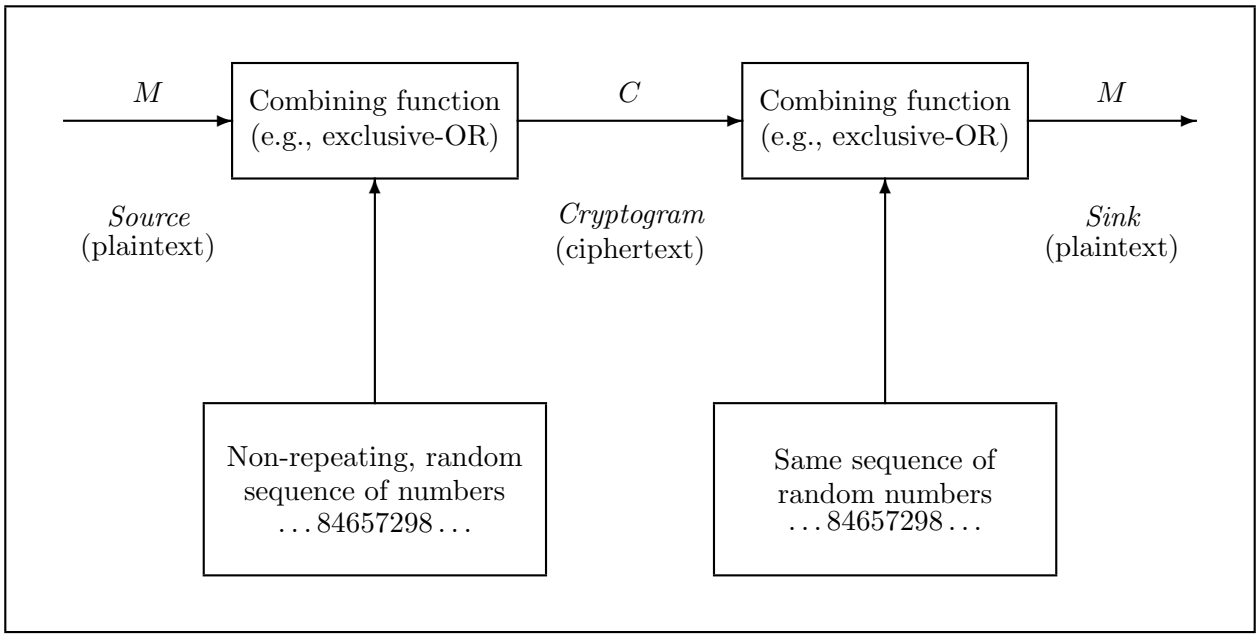


Figure 12: Vernam (one time pad) cipher.

Example*(Vernam cipher):*

Encrypt the message $M = 0011010111\ldots$ with the key $K = 1010110101\ldots$:

$$\begin{array}{lcccccccccccc} M & = & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & \ldots \\ K & = & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & \ldots \\ C & = & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & \ldots \end{array}$$

At the receiving end, the legitimate user, having a replica of the key stream K can decrypt properly the ciphertext C.

$$\begin{aligned} m_i &= c_i \oplus k_i \\ m_i &= (m_i \oplus k_i) \oplus k_i \\ m_i &= m_i \end{aligned}$$

Without this key K, all messages $\tilde{M}$ are equally probable, hence making the cipher unconditionally secure.

The Vernam cipher is unbreakable if:

1. It is a one time pad (never repeated)
2. All the keys are chosen with equal probability

The Vernam cipher thus requires a long random key sequence which must somehow be made available to the legitimate user at the receiving end. A random noise source (e.g. noise from a diode) must be used to generate the key stream which can be recorded on a magnetic tape, for instance. This requires also secure transportation of the magnetic tape. If the key is repeated, then the cipher is no longer unbreakable. Let's assume that two plaintext messages, M and M' are encrypted with the same random key sequence K:

$$c_i = m_i \oplus k_i \qquad \text{and} \qquad c'_i = m'_i \oplus k_i$$

A cryptanalyst having intercepted the two ciphertexts C and C' may add them together to form a third ciphertext C'':

$$c''_i = c_i \oplus c'_i = (m_i \oplus k_i) \oplus (m'_i \oplus k_i) = m_i \oplus m'_i \oplus k_i \oplus k_i) = m_i \oplus m'_i$$

This ciphertext C'' is no longer the result of a random running key cipher and thus may be cryptanalyzed as a running-key cipher (e.g. using the Friedman method for instance). Once the plaintext messages M and M' are decrypted the key string K is easily recovered.

$$k_i = m_i \oplus c_i \qquad \text{or} \qquad k_i = m'_i \oplus c'_i$$

It may even be used to decrypt future messages encrypted by the same sequence K of random sequence!

4 Transposition ciphers

Transposition ciphers, also called permutation ciphers, rearrange the plaintext message symbols in a different order. Often, the permutation of the characters will be done over a fixed period of d symbols:

$$M = m_1, m_2, \ldots, m_d, m_{d+1}, m_{d+2}, \ldots, m_{2d}, m_{2d+1}, \ldots$$

$$\begin{aligned} C &= E_K(M) \\ C &= m_{f(1)}, m_{f(2)}, \ldots, m_{f(d)}, m_{f(d+1)}, m_{f(d+2)}, \ldots, m_{f(2d)}, m_{f(2d+1)}, \ldots \\ C &= m_{f(1)}, m_{f(2)}, \ldots, m_{f(d)}, m_{d+f(1)}, m_{d+f(2)}, \ldots, m_{d+f(d)}, m_{2d+f(1)}, \ldots \end{aligned}$$

where the function $f(i)$ is the permutation of the ith input symbol index.

Example*(Transposition cipher):*

For instance, let the permutation function $f(i)$ be:

$$\begin{array}{lcccccc} i & = & 1, & 2, & 3, & 4, & 5, & 6 \\ f(i) & = & 3, & 1, & 6, & 5, & 2, & 4 \end{array}$$

Then if the original message M is *"... mobile channel is ..."*:

$$\begin{array}{lcccccccccccccc} M & = & m & o & b & i & l & e & c & h & a & n & n & e & l & \ldots \\ & & m_1 & m_2 & m_3 & m_4 & m_5 & m_6 & m_7 & m_8 & m_9 & m_{10} & m_{11} & m_{12} & m_{13} & \ldots \end{array}$$

the periodic transposition cipher C is obtained by the following periodic permutation, (period $d = 6$):

$$\begin{array}{lcccccccccc} C & = & m_{f(1)} & m_{f(2)} & m_{f(3)} & m_{f(4)} & m_{f(5)} & m_{f(6)} & m_{f(7)} & m_{f(8)} & m_{f(9)} & \ldots \\ & = & m_{f(1)} & m_{f(2)} & m_{f(3)} & m_{f(4)} & m_{f(5)} & m_{f(6)} & m_{6+f(1)} & m_{6+f(2)} & m_{6+f(3)} & \ldots \\ & = & m_3 & m_1 & m_6 & m_5 & m_2 & m_4 & m_{6+3} & m_{6+1} & m_{6+6} & \ldots \\ & = & m_3 & m_1 & m_6 & m_5 & m_2 & m_4 & m_9 & m_7 & m_{12} & \ldots \\ C & = & B & M & E & L & O & I & A & C & E & \ldots \end{array}$$

That is C is *"... BMELOIACENHN ..."* after the transposition transformation.

The unicity distance N_U of a periodic transposition cipher is obtained by determining the number of possible permutations, or keys, of d different characters. Since there are $d!$ arrangements of keys, and assuming that for maximum security, these keys are chosen with equal probability, that is:

$$p(k_i) = \frac{1}{d!} \qquad \text{for } 1 \leq i \leq d!,$$

and the key entropy $H(K)$ is

$$\begin{aligned} H(K) &= -\sum_{i=1}^{d!} p(k_i) \log_b p(k_i) = -\sum_{i=1}^{d!} \left(\frac{1}{d!}\right) \log_b \left(\frac{1}{d!}\right) = -\log_b \left(\frac{1}{d!}\right) \\ H(K) &= \log_b d! \end{aligned}$$

therefore the unicity distance N_U is:

$$\boxed{N_U = \frac{H(K)}{D} = \frac{\log_b d!}{D}}$$

For instance if, as above, a transposition cipher of period $d = 6$ is employed to encrypt English language plaintext ($D = 3.2\ Sh$), a cryptanalyst would need theoretically $N_U = 3$ characters to break the code.

$$\begin{aligned} N_U &= \frac{\log_2 6!}{3.2} = \frac{\log_2 720}{3.2} \\ N_U &= 2.922\ symbols \end{aligned}$$

For transposititon ciphers, the cryptanalysis can be done by:

1. Trial and error, in rearranging the letters (e.g. anagrams, crosswords)
2. Finding symbols or letters of *suspected* words
3. By analyzing the digram and trigram distributions (as on figures **??** and 5)

5 Product ciphers

By combining both substitution and transposition transformations, it is possible to increase the security of cryptosystems.

- When transposition, or permutation, is applied to substitution ciphers, digrams, trigrams etc. are broken by the permutation of characters and therefore methods such as Friedman method cannot be used (at least without additional processing) to break the codes.
- On the other hand, when substitution transformation is used on top of transposition; cipher symbols cannot be easily arranged into rows and columns by simple inspection to form anagrams that can then be easily decrypted.

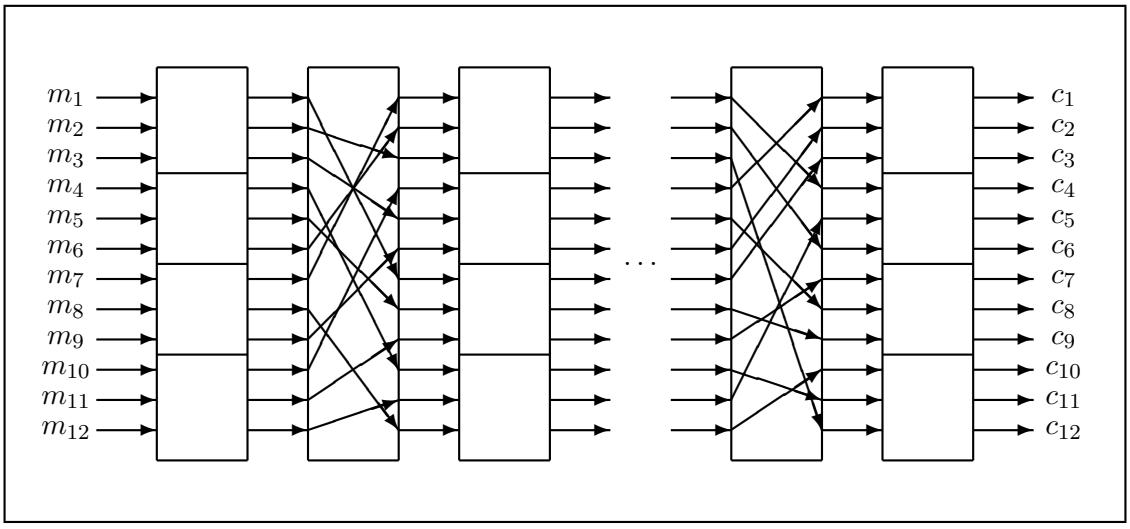


Figure 13: Product (substitutions and transpositions) cipher.

However, there are also disadvantages into using product ciphers; mistakes or errors are frequent since the encryption and decryption process are more difficult to perform by the legitimate parties. For this purpose, mechanical cipher machines, such as the Jefferson Wheel Cipher machine, have been used in the past to facilitate the encipherment and decipherment of product ciphers. The encryption process can be represented as follows:

$$C = E_K(M) = S_t \circ P_{t-1} \circ \ldots \circ P_2 \circ S_2 \circ P_1 \circ S_1(M),$$

while the decryption in done in the reverse order of the transformations:

$$\tilde{M} = D_K(C) = S_1^{-1} \circ P_1^{-1} \circ \ldots \circ P_{t-2}^{-1} \circ S_{t-1}^{-1} \circ P_{t-1}^{-1} \circ S_t^{-1}(C).$$

A common example of a product encryption scheme is the Data Encryption Standard for which a sequence of 16 rounds of substitution and permutation transformations is done on a 64-bit plaintext vector.

6 Rotor cryptographic machines

With the advent of electric typewriters, electromechanical devices were introduced to facilitate the encryption and decryption processes. The use of *rotors* and *stators* wheels allowed for multiple substitution transformations on the plaintext to create ciphertexts.

During the World War II, such encryption machines were used by Germany (the ENIGMA machine) and by Japan (the Purple machine). For those, each rotor performed a substitution transformation while the stator was used a *reflector* to further improve the cryptosystem by reflecting back the ciphertext in the rotors resulting in additionnal substitution transformations.

The basic ENIGMA cryptosystem with three rotors was broken by the British Intelligence during the World War II: Alan Turing developed a machine called the *Bomb* which basically attacked the ENIGMA by exhaustive search (i.e., brute force attack). A fourth and fifth rotor wheels were added at the end of the war making its cryptanalysis even more difficult.

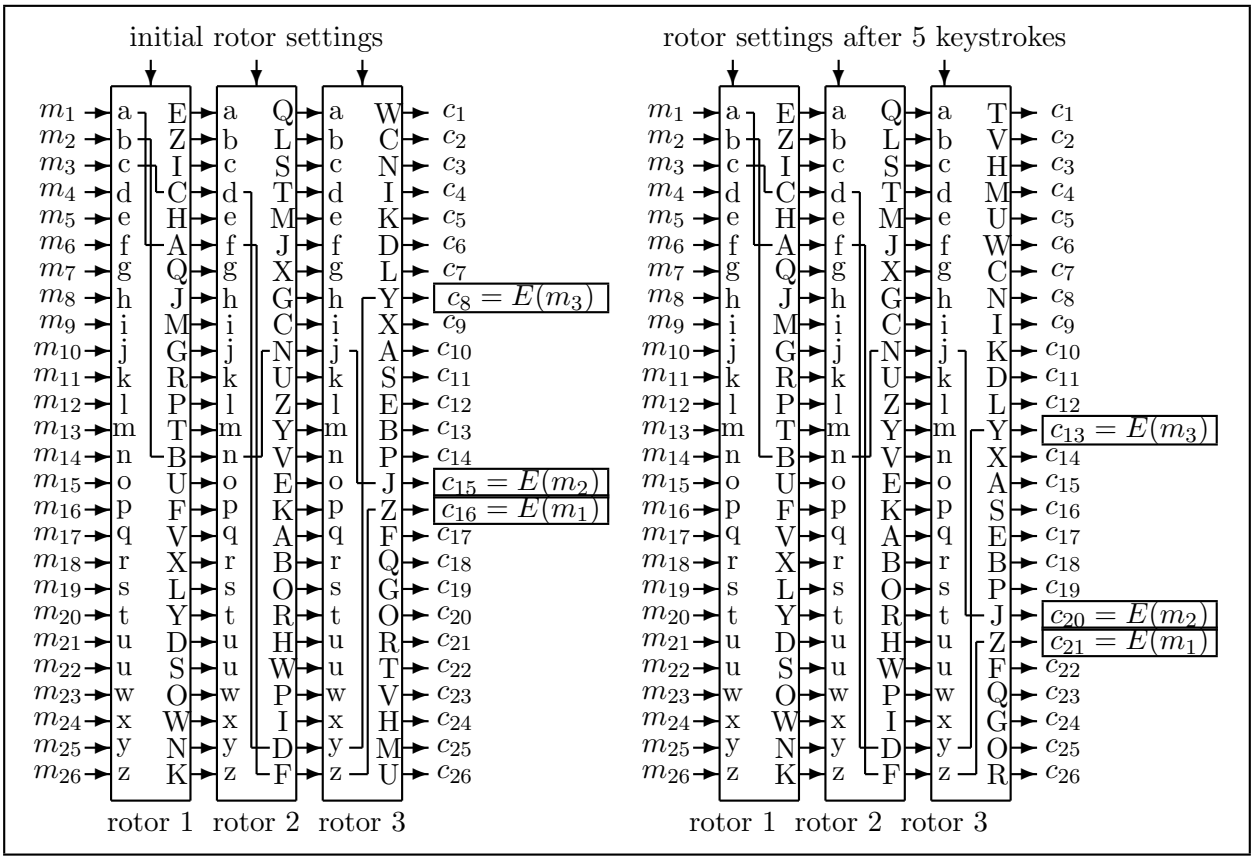


Figure 14: Original rotor settings and settings after 5 keystrokes for the three-rotor machine.

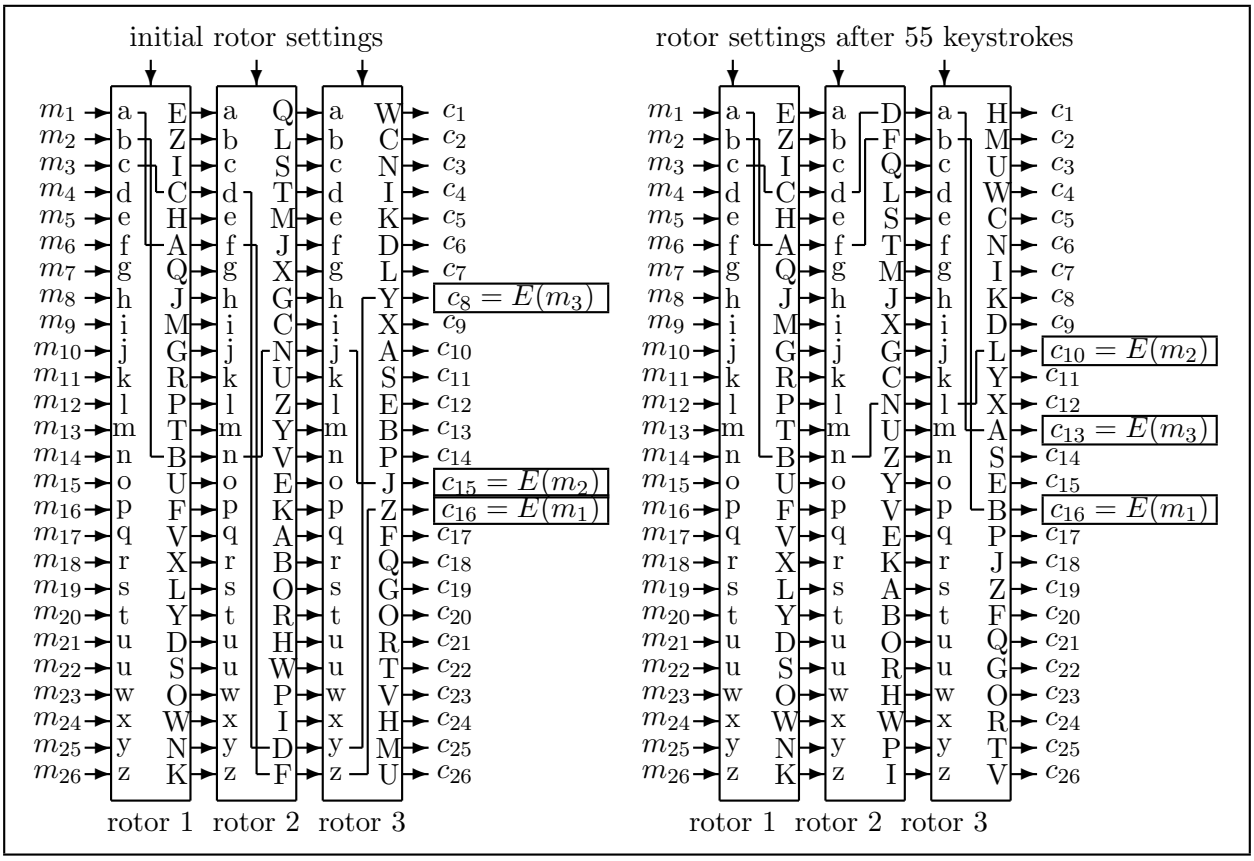


Figure 15: Original rotor settings and settings after 55 keystrokes for the three-rotor machine.

References

[Den82] D.E. Denning. *Cryptography and Data Security.* Addison-Wesley, Reading, Massachusetts, 1982.

[SJP89] J. Seberry and J. J. Pieprzyk. *Cryptography: an Introduction to Computer Security.* Prentice-Hall, Sydney, Australia, 1989.