PART TWO

Monographic Substitution Systems

CHAPTER 3

MONOALPHABETIC UNILATERAL SUBSTITUTION SYSTEMS USING STANDARD CIPHER ALPHABETS

Section I

Basis of Substitution Systems

3-1. Substitution Systems

The study of analysis of substitution systems begins with the simplest of systems. The systems explained in Part Two are monographic substitution systems. The systems in Chapters 3 and 4 are further categorized as monoalphabetic unilateral substitution systems.

a. Both *monographic* and *unilateral* mean *one letter* by their construction. The prefixes *mono-* and *uni-* mean one, and *graphic* and *literal* refer to *letters* or other characters. Monographic systems are those in which one plaintext letter at a time is encrypted. Unilateral systems are those in which the ciphertext value is always one character long. Note that the term monographic refers to single plaintext letters and the term unilateral refers to single ciphertext letters.

b. Monoalphabetic systems are those in which a given ciphertext value always equals the same plaintext value. One alphabet is used. “

c. Chapter 5 deals with monoalphabetic multilateral systems, which substitute more than one ciphertext character for each plaintext character. Later parts of this manual present the analysis of polygraphic and polyalphabetic systems. Polygraphic systems substitute values for more than one plaintext letter at a time. In polyalphabetic systems, a given ciphertext character will have different plaintext equivalents at different times through the use of multiple alphabets.

d. The techniques used with these simplest of systems carry over to the more complicated systems. Whether or not you will ever see the very simple systems in use, the same skills are used in combination with other techniques to solve more secure systems as well.

3-2. Nature of Alphabets

A cipher alphabet lists all the plaintext values to be enciphered paired with their ciphertext equivalents. Cipher alphabets can take many different forms from a simple listing of 26 letters with 26 equivalent letters to much more complex charts. Chapters 3 and 4 deal with the simple 26 letter for 26 letter types and Chapter 5 introduces some of the more complex chart type multilateral systems.

a. The simple 26 letter for 26 letter cipher alphabets are composed of two sequences of letters: the plain component sequence and the cipher component sequence. The letter sequences can be in standard A through Z order, systematically mixed order, or randomly sequenced. Alphabets are classed as standard, mixed, or random according to the types of sequences they contain. The techniques used to solve the system depend to some extent on the type of alphabet. Alphabets in which both components are standard A through Z sequences are called standard alphabets.

b. A standard sequence does not have to be written beginning with A and ending with Z. A sequence is considered to have no beginning or ending, but continues as if it were written in a circle. The letter that follows Z in a standard sequence is A. Each of the following examples is a standard sequence.

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

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

c. If the alphabetic progression is in the normal left to right order, it is called a direct standard sequence. If the alphabetic progression proceeds from right to left, it is called a reverse standard sequence. Each of the following examples is a reverse standard sequence.

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

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

d. Standard alphabets are also classed as direct or reverse. If the two standard sequences (plaintext and ciphertext) run in the same direction, the alphabet is called a direct standard alphabet. Each of the following alphabets is a direct standard alphabet. Notice that the second one has the identical equivalents to the first and can be rewritten in left to right order without changing its substitution at all.

p: 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
c: 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

p: z y x w v u t s r q p o n m l k j i h g f e d c b a
c: Q P O N M L K J I H G F E D C B A Z Y X W V U T S R

p: j i h g f e d c b a z y x w v u t s r q p o n m l k
c: Z Y X W V U T S R Q P O N M L K J I H G F E D C B A

e. If the two standard sequences (plaintext and ciphertext) run in opposite directions, the alphabet is called a reverse standard alphabet. Notice that the two following examples of reverse standard alphabets are also equivalent.

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

p:	g	f	e	d	c	b	a	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h
c:	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

f. An alphabet, in which the plain component is shown in A through Z order, is called an enciphering alphabet. The first alphabet after paragraph 3-2e is an enciphering alphabet. If the cipher component is in A through Z order, it is called a deciphering alphabet. The second alphabet is a deciphering alphabet.

g. Standard alphabet cryptograms are the easiest to solve. The rest of Chapter 3 explains the techniques of cryptography and cryptanalysts of standard monoalphabetic ciphers.

Section II
Monoalphabetic Unilateral Substitution

3-3. Cryptography

The users of a monoalphabetic unilateral substitution system must know three things about the keys to the system. They must know what sequence of letters is used for the plain component, what sequence is used for the cipher component, and how the two components line up with each other. The alignment is termed the *specific key.* Whatever keys are put into use by the originating cryptographer must be known by the receiving cryptographer, too. The key selection must either be prearranged or sent along with the cryptogram itself.

a. Prearranged keys are normally included in published operating instructions, known variously as the Signal Operation Instructions (SOI) or Communications-Electronics Operation Instructions (CEOI). For example, an SOI might specify the use of direct standard sequences for an extended period and a new alignment of the two sequences at regular shorter intervals. A portion of an SOI might look like this example.

31 May 1989, 0001-0600Z

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

31 May 1989, 0601-1200Z

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

Another way to provide exactly the same information in a more abbreviated form is shown below.

31 May 1989

Plain component: Direct standard sequence.
Cipher component: Reverse standard sequence.

0001-0600Z: Ap = Qc
0601-1200Z: Ap = Tc

In this example, the alphabet construction is left to the cryptographer, who writes out the sequences and aligns them with each other according to the specific keys for each key period.

b. Transmitted keys are used whenever the cryptographer is given some choice of the specific key selections. For example, if the alignment of the sequences were left to the cryptographer, the alignment would need to be transmitted. One way to do this is to agree that the first group of the message is always the cipher equivalent of plaintext A repeated five times. This group then tells the receiving cryptographer how to align the alphabet. The example is simple, but more complex systems can be used for greater security.

3-4. Message Preparation

The cryptographer normally prepares a message for encryption by writing the plaintext in regular length groups. Four or five letter groups are common for this type of system.

a. Word lengths are not preserved normally, because they provide strong clues to the plaintext when they appear. It is easier for a cryptanalyst to figure out the plaintext for example 1 in Figure 3-1 than example 2.

p:	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
c:	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

Plaintext to be enciphered: ATTACK AT DAWN

● Example 1: Word length encipherment.

p: attack at dawn
c: JCCJLT JC MJFW

Resulting cryptogram: JCCJLT JC MJFW

● Example 2: Four letter group encipherment.

p: atta ckat dawn
c: JCCJ LTJC MJFW

Resulting cryptogram: JCCJ LTJC MJFW

Figure 3-1. Word and group length encipherment.

b. In writing out the message for encipherment with a simple system, any numbers in the text must be spelled out or left in the clear. Punctuation must be spelled out or omitted. At the end of sentences, PD or STOP is often used in English. Commas are replaced by COMMA or CMA.

c. Whenever the text does not break evenly into groups, the text will generally be padded to fill out the groups. The filler letters are usually added at the end of the last group. For clarity, they are often just a repeated low frequency letter such as X or Z. The above cryptogram, broken into five letter groups, appears below.

JCCJL TJCMJ FWXXX

Section III

Solution of Monoalphabetic Unilateral Ciphers Using Standard Cipher Alphabets

3-5. Methods of Solution

Because of the extreme simplicity of standard alphabets, cryptograms enciphered with them can always be solved. There are two general approaches to solving these simple ciphers. One makes use of the frequency characteristics discussed in Chapter 2. The other uses the orderly progression of the alphabet to generate all possible decipherments from which you can pick the correct plaintext. Each method is explained in the following paragraphs.

3-6. Frequency Matching

The first approach consists of matching expected plaintext letter frequencies with the observed ciphertext letter frequencies.

a. As explained in Chapter 2, monoalphabetic unilateral ciphers preserve exactly the same letter frequencies as found in plaintext. The frequencies occur with the cipher equivalents, not the plaintext letters, but the numbers are unchanged. If E was the most common plaintext letter in a cryptogram, then E's replacement will be the highest frequency ciphertext letter.

b. With standard alphabets, another characteristic is preserved in addition to the individual letter frequencies. The order of highs and lows is also preserved. With a direct standard alphabet, the pattern of peaks and troughs remains, although shifted to the right or left. With a reverse standard alphabet, the pattern also remains, but it runs in the opposite direction. Figure 3-2 illustrates the expected frequency distribution of 100 letters of plaintext. It then shows what happens to the distribution when it is enciphered by a direct and a reverse standard alphabet.

c. As shown in Figure 3-2, there are several recognizable patterns in plaintext. First is the three peak pattern formed by the letters A through I. The pattern is a peak (A), a three letter trough (BCD), a peak (E), a three letter trough (FGH), and a peak (I). The second easy to recognize pattern is formed by the letters N through T. The pattern is a double peak (NO), a trough (PQ), and a triple peak (RST). When you compare the plaintext distribution with the two ciphertext distributions, the patterns are still evident.

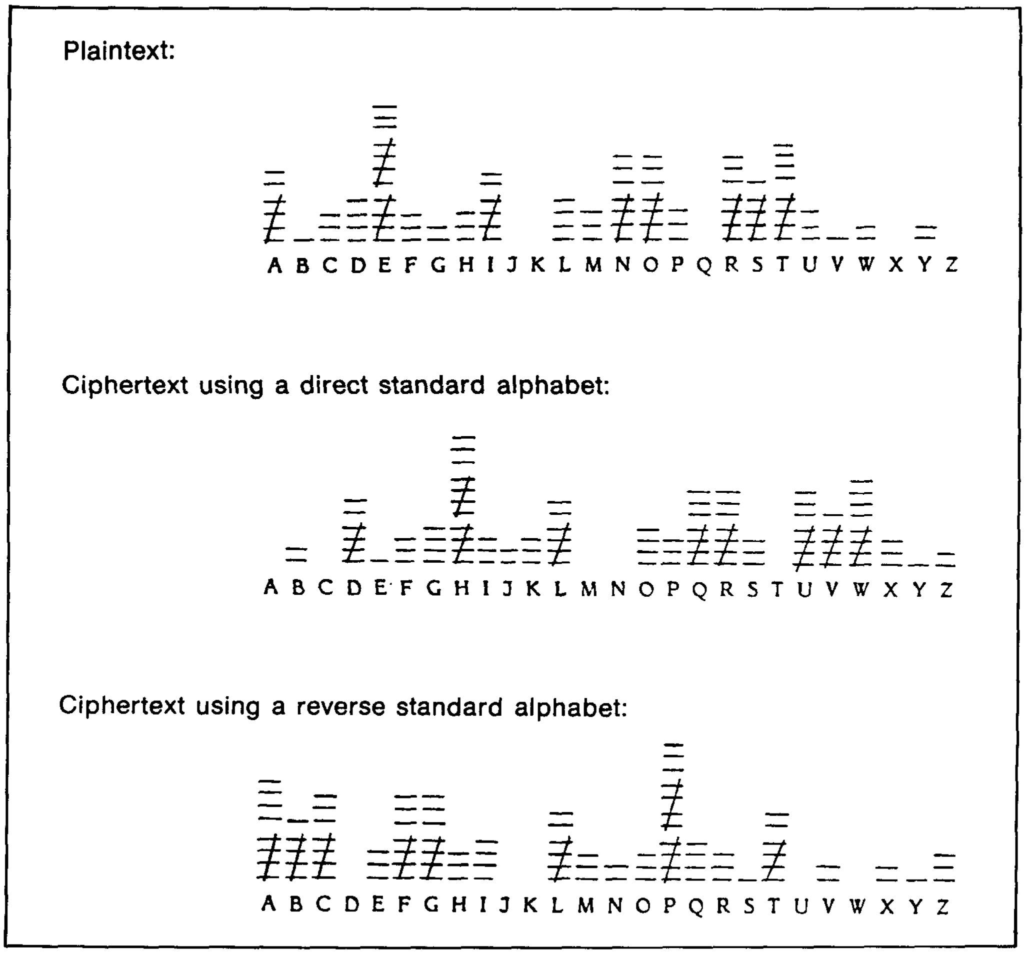
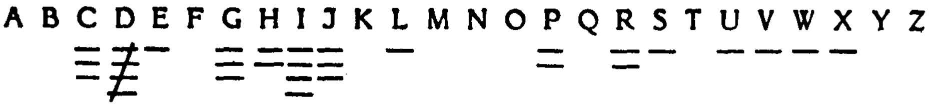


Figure 3-2. Frequency distributions.

d. Not all plaintext frequency distributions show the patterns clearly. The examples in Figure 3-2 show a perfect 100 character frequency distribution with every letter appearing exactly as many times as expected. Actual frequency counts will vary considerably, particularly with small samples. It is easier to recognize the overall patterns by their frequency than it is to recognize individual letters, however. If you can recognize even a partial pattern, it is easy to write the whole alphabet and see if the frequencies are close to expectations. Consider the cryptogram shown below.

CDRDC IPRIS JGXCV EPHII LDUDJ GWDJG HXXXX



The four Xs at the end are almost certainly fillers, so they are not counted. The cryptogram is too short for the complete pattern to appear. The cluster of higher frequency letters from C through I could represent the N through T pattern, though. We will write the full sequence of letters on that assumption.

p: 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
c: 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

The frequency match fits the plaintext letters reasonably well. E does not appear at all, but other vowels make up for it, keeping the vowels near the expected 40 percent. No low frequency letters appear with unexpectedly high frequency. The confirmation of the match occurs when the alphabet is tried with the cryptogram.

nocon tactd uring pastt wofou rhour s
CDRDC IPRIS JGXCV EPHII LDUDJ GWDJG HXXXX

or

NO CONTACT DURING PAST TWO FOUR HOURS

e. This method depends on knowing or suspecting that standard alphabets are used. With a long message, the frequency count will usually make it obvious. The A-E-I and the NO-RST peaks will stand out. With a short message like the above example, it is not obvious, but it is an easy step to try if you think you spot a partial match.

3-7. Generating All Possible Solutions

The frequency matching technique only works if the text is long enough to produce a recognizable frequency count. A second technique always leads to the solution. With a known standard alphabet, there are only 26 different ways the alphabet can be aligned. It does not take very long to try all 26 settings to find the correct solution.

a. As an example, consider the solution of the following cryptogram.

SIZUX VJFLK

With no repeated letters, frequency matching is not likely to help. Suppose the alphabet was a direct standard with p:a=c: Z.

p: 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
c: 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

Using the above alphabet, SIZUX VJFLK *deciphers* as TJAVY WKGML. Obviously, this is not the correct plaintext. The text the trial decipherment produces is called *pseudoplaintext* or *pseudotext.* Suppose the alphabet used p:a=c:Y.

p:	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
c:	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

This alphabet produces **UKBWZ XLHNM.**
The next alphabet with p:a=c:X gives the text **VLCXA YMION.**
The next alphabet with p:a=c:W gives the text **WMDYB ZNJPO.**
The next alphabet with p:a=c:v gives the text **XNEZC AOKQP.**

Clearly, not one of these is the correct setting, but notice the effect of trying each alphabet in turn. The columns of letters from each successive trial alphabet are in alphabetical order. You can achieve the same effect as trying each alphabet in turn by listing the letters vertically in alphabetical order. Figure 3-3 lists the results of trying all possible alphabets.

```
SIZUX VJFLK

TJAVY WKGML
UKBWZ XLHNM
VLCXA YMION
WMDYB ZNJPO
XNEZC AOKQP
YOFAD BPLRQ
ZPGBE CQMSR
AQHCF DRNTS
BRIDG ESOUT
CSJEH FTPVU
DTKFI GUQWV
EULGJ HVRXW
FVMHK IWSYX
GWNIL JXTZY
HXOJM KYUAZ
IYPKN LZVBA
JZQLO MAWCB
KARMP NBXDC
LBSNQ OCYED
MCTOR PDZFE
NDUPS QEAGF
OEVQT RFBHG
PFWRU SGCIH
QGXSV THDJI
RHYTW UIEKJ
```

Figure 3-3. All possible decipherments.

The plaintext, *BRIDGES OUT,* appears about halfway down the columns. In practice, you would only write enough to recognize the plaintext. Generally, write a column at a time, and only write as many columns as you need. Once you have spotted plaintext, set up the alphabet and complete the decipherment.

b. With a reverse standard alphabet, another step must be added. You cannot generate the columns until you try deciphering first at any alphabet setting of your choice. Then generate the columns starting with your trial decipherment. As you will see in the following chapters, this technique can be used with any known alphabets, not just standard ones. The procedures, which will be illustrated in Chapter 4, are—

- Set up the known alphabet at any alignment.
- Perform a trial decipherment to produce pseudotext.
- Using the trial decipherment as the letters at the head of the columns, generate all possible decipherment by listing the plain component sequence vertically for each column.