Secret Codes II

BY

John Loxton

There are essentially only two operations involved in the encryption of a message : substitution and transposition. The first article in this series (Parabola, volume 18, number 3), dealt with simple substitution systems and now I move on to transposition. In addition to these basic systems and various refinements of them, there are schemes for shortening the message by replacing commonly occurring phrases by shorter code groups. For example,

ADAUX	am awaiting arrival of
ADAXA	arrived all right, address letter care of

and so on. Schemes like this are designed primarily for economy; their illusions of secrecy have been shattered forever by Thurber's account of the secret life of a code clerk in the State Department during the first world war.
("The beast in me and other animals" by James Thurber.) He explains, "in enciphering messages in one code, in which the symbol for 'quote' was (to make up a group) 'ZOXIL,' we were permitted to use 'UNZOXIL' for 'unquote' .... The Department may have comforted itself with the knowledge that even the most ingenious and complex codes could have been broken down by enemy cipher experts. Unzoxilation just made it a little easier for them .... I doubt that we could have got through a second world war shouting 'ZOXIL Here we come ready or not UNZOXIL.'"

Transposition systems

A transposition code tries to conceal the message by scrambling the order of the letters. In the simplest scheme, the key is a permutation P of the numbers $1,2,\ldots,n$. The message is coded by breaking it into blocks of n letters and rearranging each block using the permutation P. For example, take $n = 5$ and let P be the permutation

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 5 & 2 & 4 \end{pmatrix}$$

This means that, in each block of five letters, the first letter is moved to the third position, the second letter to the first position and, in general, the j-th letter goes to the position shown under j in the permutation P. With the permutation given above, the message

THIS I|S ANOT|HER AR|TICLE| ON SEC|RET CO|DES XX

would be enciphered

HSTII AOSTN EAHRR ILTEC NEOCS ECROT EXDXS .

(Note that two dummy letters XX must be added to the message to fill up the last group of five letters. Decoding is just a matter of unscrambling the cipher text by using the inverse permutation P^{-1} which undoes the effect of the permutation P. You can check that the unscrambling permutation for the example above is

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 5 & 3 \end{pmatrix}$$

and that this takes the message HSTII ... back to THIS I ... . Experiment with other transpositions. Devise a computer program to do the coding and decoding. What is the effect of combining two permutations of the same length or of different lengths ? Does this make the code more secure ?

The number of possible keys is very large. For example, the number of permutations of length 5 is $5! = 1 \times 2 \times 3 \times 4 \times 5$ and the number of permutations of length 20 or less is $1! + 2! + 3! + \ldots + 20! \approx 10^{17}$. However, the simple transposition cipher is easy to crack by exploiting the peculiarities of the English language. Let us solve the code

HSTII AOSTN EAHRR ILTEC NEOCS ECROT EXDXS .

Since the message has 35 letters, we are looking for a permutation of length 5 or 7. Let us try 5. From the table in the first part of this article, the most common digraph (two-letter combination) in English is TH, so we might start by setting the T and H in the first group HSTII together. This means the decoding permutation has the shape

$$P^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ k+1 & - & k & - & - \end{pmatrix}$$

Using this on every group gives the combinations

TH SA HE TI ON RE DE .

Since these digraphs are all quite common, we are probably on the right track. Now the digraph HE suggests the word THE with T coming from the second group. If we assume this, we get

$$P^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & - & 1 & 5 & - \end{pmatrix}$$

and we can write down the skeleton of the message :

TH--I SA--T HE--R TI--F ON--C RE--O DE--X

Now it is easy to see that we must have

$$P^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 5 & 3 \end{pmatrix} \quad \text{and} \quad P = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 5 & 2 & 4 \end{pmatrix} .$$

What happens if you assume the permutation has length 7 ? The following BASIC program devised for the APPLE II Computer can be used to perform the manipulations. Can you devise something more efficient ?

PROGRAM FOR SOLVING SIMPLE TRANSPOSITION CODES

```
100  HOME : DIM P(10)
110  PRINT "    SIMPLE TRANSPOSITION SOLVER": PRINT
120  PRINT "TYPE THE CIPHER MESSAGE"
130  PRINT "TEXT MUST NOT EXCEED 250 CHARACTERS"
140  PRINT "CHARACTERS NOT A TO Z ARE IGNORED"
150  PRINT "FINISH WITH 'RETURN'": PRINT
160  INPUT C$:L =  LEN (C$):D$ = ""
170  FOR J = 1 TO L:X =  ASC ( MID$ (C$,J,1)) - 64
180  IF X > 0 AND X < 27 THEN D$ = D$ +  MID$ (C$,J,1)
190  NEXT J:C$ = D$:L =  LEN (C$): PRINT
200  PRINT "START DECODING": PRINT
210  INPUT "LENGTH OF PERMUTATION (AT MOST 10) = ";N:K = L / N: PRINT
220 D$ = "":X = 1: FOR J = 1 TO K
230 D$ = D$ +  MID$ (C$,X,N) + " ":X = X + N
240  NEXT J: PRINT D$: PRINT
250  PRINT "DECODING PERMUTATION"
260  PRINT "TO SEND I TO POSITION J TYPE J UNDER I"
270  PRINT "TYPE * UNDER UNDETERMINED ENTRIES": PRINT
280 A$ = " ": FOR J = 1 TO N:A$ = A$ + " " +  STR$ (J) + " ": NEXT J
290  PRINT A$: INPUT B$:B$ = "  " + B$: PRINT
300 X = 2: FOR J = 1 TO N:P(J) =  VAL ( MID$ (B$,X,3)):X = X + 3: NEXT J
310 M$ = "":R$ = "":X = 0
320  FOR J = 1 TO K:Y = N + 1
330  FOR I = 1 TO N:M$ = M$ + " ": NEXT I
340  FOR I = 1 TO N
350  IF P(I) < 1 OR P(I) > N THEN  GOTO 390
360 T = X + J + P(I)
370  IF T = 2 THEN M$ =  MID$ (C$,X + I,1) +  MID$ (M$,T): GOTO 400
380 M$ =  LEFT$ (M$,T - 2) +  MID$ (C$,X + I,1) +  MID$ (M$,T): GOTO 400
390 R$ = R$ +  MID$ (C$,X + I,1):Y = Y - 1
400  NEXT I:M$ = M$ + " "
410  FOR I = 1 TO Y:R$ = R$ + " ": NEXT I
420 X = X + N: NEXT J
430  PRINT "CIPHER TEXT": PRINT D$: PRINT
440  PRINT "RECONSTRUCTED MESSAGE": PRINT M$: PRINT
450  PRINT "RESIDUAL LETTERS IN EACH GROUP": PRINT R$: PRINT
460  PRINT "CHOOSE ALTERNATIVE REQUIRED:"
470  PRINT  TAB( 5);"1. CHANGE LENGTH OF PERMUTATION"
480  PRINT  TAB( 5);"2. CHANGE PERMUTATION"
490  PRINT  TAB( 5);"3. EXIT"
500  INPUT "ENTER 1,2 OR 3 AND PRESS 'RETURN' ";I: PRINT
510  IF I = 1 THEN  GOTO 210
520  IF I = 2 THEN  GOTO 280
530  END
540  END
```

There are many variations of the simple transposition scheme. One of these, widely used in ancient times, is columnar transposition. This works as follows. The message is first inscribed in a rectangle :

```
3 2 7 1 8 4 6 5
A L L R I G H T
H A V E I T Y O
U R O W N W A Y
Y O U H E A R D
A S E A L B A R
K
```

The key consists of the number of columns in the rectangle and the numbers at the top of the columns. These numbers are a permutation whose length is the same as the number of columns. Now the columns are transcribed in the order given by the numerical key :

```
REWHA    LAROS    AHUYA    KGTWA    BTOYD    RHYAR
ALVOU    EIINE    L .
```

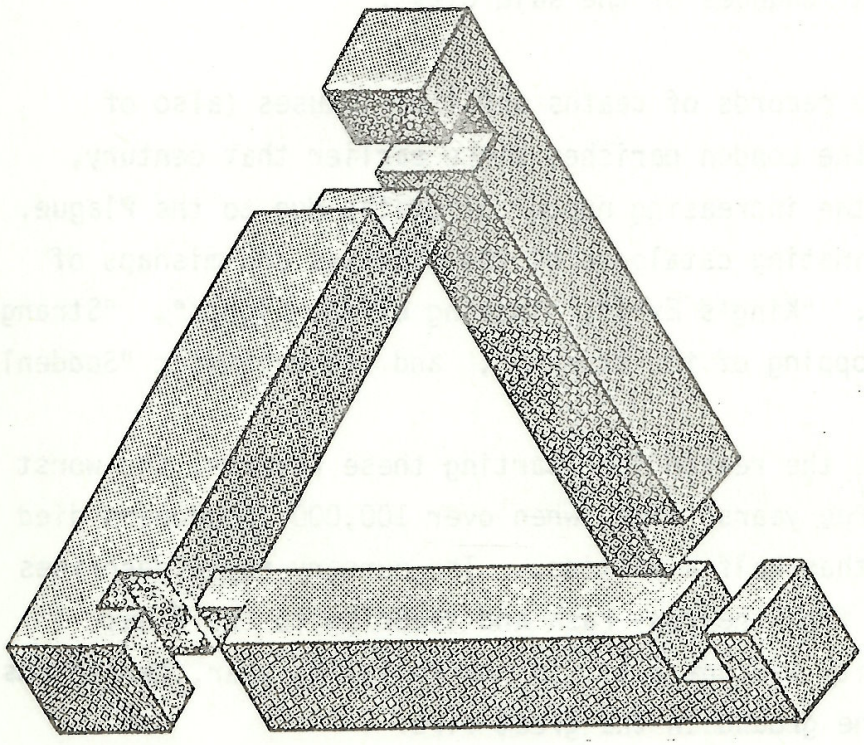
Can you work out how to decipher this code, assuming you have been given the key ? How would you set about cracking the code ? Investigate what happens when you make a double transposition, that is the message is first coded by a columnar transposition and then the result is put through a second columnar transposition ? Does this make the system more secure ?

The following examples have been coded by simple transposition. The solutions are on page 32.

(1)	RAAPP	YELTN	ATLEH	YTRET	OESRA	IFVLE
	SSERP	ELWYE	TRARE	RHGRE	YILZZ	
(2)	OHNFM	TITAI	SECCT	ERKSA	HETEE	MOTLT
	WHSRI	IIOSG	TPESI	BATSL	HSIET	ICTHR
	TOETE	NKMSC	EATKA	MOLES	WISAN	SROHE
(3)	IFHKC	CIRND	NSAEE	CKHOT	KGIES	RGEDO
	LOUNW	CCASR	KTEOT	XFNO		

Incidentally, if you intercepted a cipher which might be either a simple transposition or a substitution code, how would you decide which it was ?

So far, the solver is definitely on top. In the next part, I will describe how the basic processes of substitution and transposition can be mixed up and generally disguised to the benefit of the code clerk.



The impossible triangle made possible ???