COMPUTER-AIDED SOLUTION
OF MONOALPHABETIC
SUBSTITUTION
CIPHERS

By

GERALD DALE SMITH

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1970

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 1978

COMPUTER-AIDED SOLUTION OF MONOALPHABETIC SUBSTITUTION CIPHERS

Thesis Approved:

J P Chandler
Thesis Adviser

D. E. Hedrick

J.R.VanDoren

Norman N Durham
Dean of the Graduate College

PREFACE

This thesis describes a system of algorithms which have been implemented on a digital computer in an interactive system designed to aid in the solution of monoalphabetic substitution ciphers. The programs aid the user by recognizing short words of four letters or less and words containing patterns of repeated letters.

The author wishes to express his appreciation to his major advisor, Dr. John P. Chandler, for his interest, guidance and valuable suggestions. Also thanks to Dr. Don D. Fisher for his interest and encouragement.

Finally, a special thanks to Durand B. Lugar who first introduced me to the "Daily Cryptoquote".

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Since the beginning of written communications man has been devising methods of disguising his messages. In most cases this has been immediately countered by other men developing methods to recover the hidden messages.

In recent years a new tool has entered this picture. The computer has proven to be a powerful aid in all areas of cryptography. The principal advantage in the encoding and decoding processes is the speed with which it can perform very complicated processes. The speed of the computer is also very important in the cryptanalysis process.

The cryptanalysis process has many facets. It includes many statistical analyses which can be done quickly and accurately by a computer. Examples of these include frequency counts (number of occurrences of each letter), contact tables (occurrences of letters immediately preceding or following each letter), and position tables (relative letter position within words).

Another important area in cryptanalysis is pattern recognition. This includes the recognition of reversals, common endings, and patterns of repeated letters. The speed and large memory of the computer are both very useful in these areas.

This project deals with one specific area of pattern recognition, namely word recognition. Certain characteristics of a word such as its length and composition (in terms of repeated letters and present letters) are used to identify words. This is applied in the solution of a specific kind of simple substitution cipher.

CHAPTER II

BACKGROUND

One of the earliest methods of secret writing was a substitution cipher. W.F. Friedman's (6) definition is:

> A substitution cipher is a cryptogram in which the original letters of the plain text, taken either singly or in groups of constant length, have been replaced by other letters, figures, signs, or combinations of them, in accordance with a definite system and key (p. 7).

One type of substitution cipher is a monoalphabetic substitution cipher. In a monoalphabetic substitution cipher the same character or symbol is always substituted for the same letter. For example, an A is always used for a Q, a B is always used for an R, and so forth.

There are three different types of monoalphabetic substitution ciphers. The simplest type is a Caesar cipher. In a Caesar cipher the cipher alphabet is shifted relative to the plain alphabet. For example,

```
plain  - ABCDEFGHIJKLMNOPQRSTUVWXYZ
CIPHER - PQRSTUVWXYZABCDEFGHIJKLMNO
```

The cipher alphabet can also be reversed:

```
plain  - ABCDEFGHIJKLMNOPQRSTUVWXYZ
cipher - FEDCBAZYXWVUTSRQPONMLKJIHG
```

The second type of monoalphabetic substitution cipher is a keyword cipher. In this type a keyword is selected to define part of the substitution alphabet and then the remainder of the alphabet is used to define the rest. For example:

plain - ABCDEFGHIJKLMNOPQRSTUVWXYZ
cipher - NPQSTUBXZKEYWORDABCFGHIJLM

Of course the keyword may be placed anywhere under the plain alphabet. Also the remaining letters may be used in reverse order. For example:

plain - ABCDEFGHIJKLMNOPQRSTUVWXYZ
cipher - IHGFCBAKEYWORDZXVUTSQPNMLJ

The only restriction on keywords is that no letters may be repeated in the word. Obviously this would result in two plain letters for the same cipher letter.

The third type of monoalphabetic substitution cipher is a random cipher. With this type the letters in the cipher alphabet can be arranged in any order. Thus there is no relation between one letter and the one next to it. That makes this type of monoalphabetic substitution cipher the most difficult to solve.

Monoalphabetic substitution ciphers have been recorded at several different times in history. The Caesar cipher is credited to Julius Caesar although there is some doubt that he actually used it. However it was known at that time. Much later, in 1839, Edgar Allen Poe brought monoalphabetic substitution ciphers into public attention again by challenging his readers to submit cryptograms for him to solve. According to David Kahn (12) in The Codebreakers, he received many cryptograms and published solutions to most of them. In one case he printed an elaborate proof that a submitted cryptogram was nothing but a collection of letters submitted by someone who wanted to see what he would read into it.

Monoalphabetic substitution ciphers have also appeared in literature. Edgar Allen Poe's (18) The Gold-Bug, published in 1843, contained such a cipher. Various numbers and special characters were used for substitutions instead of letters but the method was the same. Sir Arthur Conan Doyle (3) used a similar cipher in his short story The Adventure of the Dancing Men.

Ironically, monoalphabetic substitution ciphers in all their simplicity were still being used by some governments as late as the early part of the twentieth century. According to Herbert O. Yardley (25) in his book The American Black Chamber the Mexican government was using a simple substitution cipher as late as 1917. W.F. Friedman (6) reports in his book Elementary Cryptanalysis that a German general used a simple substitution cipher for a short time in 1918 before he was informed of its vulnerability.

Of the three types of monoalphabetic substitution ciphers, the Caesar cipher is the easiest type to solve. Counting both forward and reversed alphabets there are only 51 possible solutions. Thus it is realistic to consider solving these by brute force. That is, by listing all 51 possible solutions and selecting the correct one by inspection. This is the method employed by Frederick W. Chesson (2) in his program described in his article in Datamation.

Keyword and random ciphers can not be solved so easily. The big reason is that the number of possible solutions is much larger. Specifically, the number of possible solutions of a random cipher is 26! which is equal to 403,291,461,126, 605,635,584,000,000. Obviously it would be impossible to list all the possible solutions and select the correct one by inspection. However other methods have been developed which are more effective.

There are several terms involved in the solution of ciphers that should be defined at this point. The ciphertext is the encoded message. The plaintext is the corresponding decoded message. A ciphertext letter is a letter from the ciphertext. A plaintext letter is a letter from the plaintext. A plaintext substitution is the assignment of a letter from the plaintext alphabet to a specific letter in the ciphertext alphabet. For example, ciphertext letter A corresponds to plaintext letter R.

One of the earliest methods of solution was the use of the frequency count. With messages in excess of 200 to 300 characters the number of times various characters appear usually gives a good indication of what letters they represent. Extensive research has been done to develop tables of frequencies. These include tables constructed after counting the frequencies in a text of 10,000 letters done by Ohaver (17) and Hitt (6).

With messages of shorter length, 30 to 150 characters, the frequency count is less reliable and therefore less useful. However there are other techniques that can be used. Some of these methods will be described in Chapter III.

The fact that a unique solution exists in cryptograms of such short length was shown by Claude Shannon (21) in his article "Communication Theory of Secrecy Systems." He defined a quantity called the unicity point which is the number of letters required in a cryptogram to give a unique and unambiguous solution.

The calculation of the unicity point is based on the structure of a language. The basic measure of structure is a quantity called entropy. Entropy measures the lack of structure or randomness of a language. Shannon found statistically that English has an entropy of about one bit per word. The maximum entropy (e) for an alphabet of 26 letters is 4.7 bits per letter. Using these values the redundancy (D) of English is given by the equation

$$D = 1 - 1/e$$

This gives a value of approximately 75 percent. The unicity point is defined by the equation

$$up = H(K)/D$$

Where H(K) is a measure of the key space. For a simple substitution cipher in English

$$H(K) = \log_{10}26! = 20$$

Thus the unicity point of a simple substitution cipher in English is approximately 27 letters.

According to Shannon any simple substitution cryptogram in English having more than 27 or 28 letters has a unique and unambiguous solution. It has been verified experimentally that such a point exists between 20 and 30 letters.

CHAPTER III

METHOD OF SOLUTION

Solving cryptograms of relatively short length, 30 to 150 characters, often requires special techniques. Two very useful techniques involve the use of short words and pattern words.

For this thesis <u>short words</u> are defined as words of one to four letters where no letters are repeated. The cutoff at four letters was an arbitrary decision based on the memory restrictions of the IBM 1130 computer used in this project.

Short words are helpful because the number of possibilities is much less than for longer words. For instance, the total number of possible two-letter words is 650. However, many of the combinations, such as "qp" and "xy" do not occur as words in the English language. It is estimated that the actual number of valid two-letter words in English is less than 50.

As words get longer the number of possibilities increases but it remains manageable for a while. The total number of possible three-letter words is 15,600. It is estimated that there are less than 550 that are valid English words. Likewise, there are 358,800 possible four-letter words but less than 1750 of them are valid English words.

Since the number of possibilities for short words is reasonably small it is feasible to construct tables listing all the possibilities. These tables can then be used to determine possible words for use in a solution. Such tables are especially useful if one or more plaintext letters have been deciphered in a short word. When one or more plaintext letters are known a properly constructed table will quickly give the possibilities for the remaining letters. With short words of four letters or less the number of possibilities will often be very small. Thus the right solution will be fairly easy to pick.

Various techniques for utilizing short words are described in Gaines (7), Smith (21), Pratt (19), and Ohaver (17). The solution techniques used with short words vary somewhat depending on the length of the words. One technique is used for one-letter words. A second is used for two-letter words. A third is used for three and four-letter words.

The one-letter word technique is very simple. When a one-letter word is encountered a list of unused plaintext letters is checked. If A or I is available, but not both, then it is substituted. If both are available no substitution is made. If neither is available the solution may be suspect and the solution process may be stopped or continued as desired.

The technique used for two-letter words is more complicated. Substitutions are made from two-letter words only when one plaintext letter is present. The known plaintext letter and the position of the unknown letter are used to get the possible words from a table of all possible two-letter words.

For instance, if O_ is encountered then the possible words from the table might be "of", "on", and "or". The letters F, N, and R are then checked against the unused plaintext letter list. If only one letter is available then it is substituted. If two or more letters are still available then no substitution is made. If none of the letters is available then no substitution is made and the absence may be noted as an error indication.

The same technique is used for both three and four-letter words. This technique also depends on one or more plaintext letters being present in a word. The first plaintext letter present in a word is used to extract the list of possible words from the table of all possible words of that length. If more than one plaintext letter is present, those words in the extracted possible word list that do not fit are eliminated. The same type of substitution process is used that is used in the two-letter technique. However each unsolved letter is considered separately. Thus there can be from none to two substitutions with a three-letter word and none to three substitutions with a four-letter word. However, if no substitutions exist for one unsolved letter then no substitutions are made for any other unsolved letters in that word.

Pattern words are words containing one or more repeated letters. There are various methods for representing a pattern in a word. The method employed in this project is described by Harris (9). A word is represented by an equal length sequence of digits. All the nonrepeating letters are represented by zeros. The first repeating letter is represented by a one in each position where it occurs. The second repeating letter, if present, is represented by a two in each position where it occurs. This process continues until all letters are represented by a digit.

The following examples illustrate this representation system. The pattern 000001001 represents a nine-letter word with the sixth and ninth letters the same. This pattern would represent the word "knowledge" or the word "important". The pattern 12342530415 represents an eleven-letter word where the first and tenth letters are the same, the second and fifth letter are the same, the third and seventh letters are the same, the fourth and ninth letters are the same, and the sixth and eleventh letters are the same. This pattern would represent the word "unconscious".

Pattern words are helpful in the solution process because having a pattern greatly reduces the number of possibilities for a word. At least 50% of English words are pattern words. More than 70% of the patterns represent single words. Thus a list or file of pattern words is very useful. If it is properly ordered, the possible words for a given pattern can be quickly obtained.

Various techniques for utilizing pattern words are described in Gaines (7), Pratt (19), Harris (9), and Ohaver (17). The substitution technique used with pattern words in this project is the same as the three or four-letter word technique. The only difference is that no plaintext letters need to be known because the pattern is used to reduce the universe of possibilities.

These techniques are particularly powerful when they are used together on an entire cryptogram. As each substitution is developed it is made throughout the entire cryptogram. This reduces the number of possibilities for remaining words which leads to more substitutions. This process is continued until the entire cryptogram is solved. This process is illustrated in the example in the next chapter.

CHAPTER IV

EXAMPLE OF SOLUTION PROCESS

The solution process is best illustrated by going through an example, step by step. Note, in the following example the short word tables and pattern word file used were those available in the computer programs to be described later. The following cryptogram is a typical example of those often found in newspapers.

```
KWU QPGDZUYNU KWRK DWRKUAUF WRJJUPI KG EGC VI HGF
    000001001 1001 00001010 0011000

EGCF NGGY, FRVIUI EGC KG KWU WUVNWKI GH ZVAVPN VP
     0110  000101            1000100    010100

JRFRYVIU. - WRIVYVO IREVPN
01010000
```

Notice that the pattern words have their associated patterns under them.

Since no plaintext letters are known initially the first substitutions must be developed from pattern words. The first pattern word in the cryptogram is the second word. The only word in the pattern word file matching the pattern 000001001 is the word "knowledge". Since only one possible word is known it is substituted into the cryptogram giving the following:

```
  E KNOWLEDGE      W   E E      EN   O  O      O
KWU QPGDZUYNU KWRK DWRKUAUF WRJJUPI KG EGC VI HGF
    000001001 1001 00001010 0011000

 O   GOOD      E   O   O   E  E G    O  L   NG  N
EGCF NGGY, FRVIUI EGC KG KWU WUVNWKI GH ZVAVPN VP
     0110  000101           1000100    010100

    D  E        D       NG
JRFRYVIU. - WRIVYVO IREVPN
01010000
```

After the substitutions have been made processing continues with the third word. Since it is a pattern word the pattern word file is checked. Although eight possibilities are found only the words "sits", "sons" and "that" fit with the available unused plaintext letters. Since there are no common letters no substitutions can be derived.

The fourth word is also a pattern word. Checking the pattern word file gives the four possible words: "achieved", "whatever", "learning" and "virtuous". With the known plaintext letters the only word that will fit is "whatever". Making those substitutions gives the following:

```
THE KNOWLEDGE THAT WHATEVER HA  EN  TO  O      OR
KWU QPGDZUYNU KWRK DWRKUAUF WRJJUPI KG EGC VI HGF
    000001001 1001 00001010 0011000

 O R GOOD  RA  E   O  TO THE HE GHT  O  L V NG  N
EGCF NGGY, FRVIUI EGC KG KWU WUVNWKI GH ZVAVPN VP
     0110  000101           1000100    010100

 ARAD  E    HA  D    A  NG
JRFRYVIU. - WRIVYVO IREVPN
01010000
```

With these substitutions the fifth word is over half solved. Checking the pattern word file gives the following six possibilities: "freedom", "freeing", "collide", "fellows", "pillars" and "telling". Since the known plaintext letters do not correspond with any of thepossible words no additional substitutions can be developed from this word.

The sixth word has been completely solved so consideration moves on to the seventh word. The seventh word is a three-letter word with one plaintext letter known. Since a plaintext letter is known the three-letter word table can be used to find possible words. Considering the unused plaintext letters available gives four possibilities: "boy", "joy", "you" and "job". Again there are no common letters in any position so no substitutions can be made.

No substitutions have affected the eighth word at this point so processing continues to the ninth word. The ninth word is a three-letter word with two plaintext letters known. The only three-letter word that fits these conditions is the word "for". Making the substitution gives the following:

```
THE KNOWLEDGE THAT WHATEVER HA  EN  TO  O     FOR
KWU QPGDZUYNU KWRK DWRKUAUF WRJJUPI KG EGC VI HGF
    000001001 1001 00001010 0011000

 O R GOOD  RA  E   O  TO THE HE GHT  OF L V NG  N
EGCF NGGY, FRVIUI EGC KG KWU WUVNWKI GH ZVAVPN VP
     0110  000101            1000100     010100

 ARAD  E    HA  D    A  NG
JRFRYVIU. - WRIVYVO IREVPN
01010000
```

The tenth word is a four-letter word with two plaintext letters currently known. Checking the four-letter word table and taking into account the available plaintext letters yields two possibilities, "pour" and "your". In this case the first letters of the possible words are different but the third letters of both words are the same. Since there is only one possibility for the third letter it is substituted giving the following:

```
THE KNOWLEDGE THAT WHATEVER HP  EN  TO  OU    FOR
KWU QPGDZUYNU KWRK DWRKUAUF WRJJUPI KG EGC VI HGF
    000001001 1001 00001010 0011000

 OUR GOOD  RA  E   OU TO THE HE GHT  OF L V NG  N
EGCF NGGY, FRVIUI EGC KG KWU WUVNWKI GH ZVAVPN VP
     0110  000101            1000100    010100

 ARAD  E    HA  D    A  NG
JRFRYVIU. - WRIVYVO IREVPN
01010000
```

At this point the eleventh word has been completely solved by previous substitutions. The twelfth word is half solved but no possibilities are listed in the pattern word file for that pattern, so no substitutions can be derived from it. The thirteenth word is a partially solved three-letter word. Using the known plaintext letters and the three-letter word table gives only one possible word, "you". Since the letter Y is still available it is substituted into the cryptogram giving the following:

```
THE KNOWLEDGE THAT WHATEVER HA  EN  TO YOU    FOR
KWU QPGDZUYNU KWRK DWRKUAUF WRJJUPI KG EGC VI HGF
    000001001 1001 00001010 0011000

YOUR GOOD  RA  E  YOU TO THE HE GHT  OF L V NG  N
EGCF NGGY, FRVIUI EGC KG KWU WUVNWKI GH ZVAVPN VP
     0110  000101            1000100    010100

 ARAD  E    HA  D   AY NG
JRFRYVIU. - WRIVYVO IREVPN
01010000
```

The sixteenth word is the next one still containing some unsolved letters. It is a pattern word but checking the pattern word file gives the word "already" as the only word matching the pattern. Since the known plaintext letters in the sixteenth word do not match the letters in the word "already" no additional substitutions can be developed from this word. The same problem is encountered with the eighteenth word. Although some possible words are known, none of them fit with the known plaintext letters.

The nineteenth word is a two-letter word ending in the letter N. The two-letter word table lists three possible words: "an", "in", and "on". Since the letters A and O are already used the only letter left to substitute is I giving the following:

```
THE KNOWLEDGE THAT WHATEVER HA  EN  TO YOU I  FOR
KWU QPGDZUYNU KWRK DWRKUAUF WRJJUPI KG EGC VI HGF
    000001001 1001 00001010 0011000

YOUR GOOD  RAI E  YOU TO THE HEIGHT  OF LIVING IN
EGCF NGGY, FRVIUI EGC KG KWU WUVNWKI GH ZVAVPN VP
     0110  000101                1000100    010100

 ARADI E    HA IDI   AYING
JRFRYVIU. - WRIVYVO IREVPN
01010000
```

The twentieth word is another pattern word that has not been completely solved. Checking the pattern word file gives the word "paradise" for that pattern. The known plaintext letters match those in the possible word and the letters P and S are still available so they are substituted giving the following:

```
THE KNOWLEDGE THAT WHATEVER HAPPENS TO YOU IS FOR
KWU QPGDZUYNU KWRK DWRKUAUF WRJJUPI KG EGC VI HGF
    000001001 1001 00001010 0011000

YOUR GOOD  RAISES YOU TO THE HEIGHTS OF LIVING IN
EGCF NGGY, FRVIUI EGC KG KWU WUVNWKI GH ZVAVPN VP
     0110  000101                1000100    010100

PARADISE    HASIDI  SAYING
JRFRYVIU. - WRIVYVO IREVPN
01010000
```

At this point the only unsolved letter remaining is in the area following the last hyphen. This area generally contains the source of the quotation which precedes it. This area is not checked for pattern or short words because the vocabulary is mostly proper names and is a completely different vocabulary from the regular text. In this case it is fairly easy to see that the missing letter must be C which makes the source "Hasidic saying". If any letters in the text were still unsolved a second pass could be made and, probably, more substitutions developed because of the additional known plaintext letters and the reduced number of letters still available.

CHAPTER V

PROGRAM DESCRIPTION

A collection of programs has been written to aid in the solution of monoalphabetic substitution ciphers. These programs form an interactive system which aids the user by developing substitutions from short words and pattern words. This frees the user so he can concentrate on more complex decisions dealing with tense, antecedents, and other grammatical restrictions and questions of context. This system has proved to be quite effective in the majority of cases.

A cryptogram is processed one word at a time. When a short word (of length four or less) is encountered and one or more plaintext substitutions have been made the appropriate table is searched for additional possible substitutions. When a word with one or more repeated letters is encountered the pattern word file is searched for possible substitutions. As each word is processed any additional substitutions developed are made prior to processing the next word. Thus as each substitution is made it is utilized to develop more substitutions.

When the entire cryptogram has been processed without developing any new substitutions the program lists the cryptogram with all current substitutions. At this point the program cannot arrive at more substitutions until some additional input is received. When one or more new substitutions have been made, the program can be allowed to try again to find more substitutions. Often the additional information will lead to additional substitutions.

Cryptograms are input into the system on cards. A single cryptogram can be punched on one or two cards in the format shown in Table I. The date and sequence fields are optional. The date field is printed above the initial listing of the cryptogram for identification. The sequence field is not used at all. With the two card limit the maximum length of a cryptogram that can be solved at one time is 148 characters including spaces and punctuation. The end of the text of the cryptogram is flagged by a dollar sign ($).

TABLE I

INPUT FORMAT

CARD COLUMNS	FIELD
1 - 5	Date
6	Sequence Number
7 - 80	Text of Cryptogram

The hyphen is the only other special character with special significance. It is used to separate the cryptogram source from the text of the cryptogram. This is not particularly restrictive except when an author has a hyphenated name. In that case some other symbol should be used in his name. If a hyphenated word occurs in the cryptogram and there is no author then a hyphen should be appended on the end of the cryptogram. That part of the cryptogram following the last hyphen, normally the author's name, is not checked for patterns or short words because it generally encompasses a different vocabulary, principally proper names.

After a cryptogram has been read, the rest of the input comes from the console typewriter. The commands from the console typewriter control the operation of the program. The instruction codes allowed are listed in Table II and described in the following paragraphs.

The code A is used to print the available letter list. This command lists those plaintext letters which have not been assigned a cipher equivalent, in other words, the plaintext letters that have not been used yet.

The code END is used to terminate the program. This code terminates the program and returns control to the monitor system of the computer.

The code F is used to print the frequency count. Although a frequency count is of limited use, it is available if desired. The frequency count includes all letters in the cryptogram including the source, if present.

TABLE II

CONSOLE INSTRUCTION CODES

CODE	MEANING
A	Print Available Plaintext Letter List
END	Terminate the Program
F	Print the Frequency Count
G	Make All Possible Substitutions and Print
I	Print Instructions
NEW	Read New Cryptogram
P	Print Cryptogram with Current Substitutions
R	Restart Current Cryptogram
S	Print Current Substitutions
Wnn	Print Possible Words for Word nn
X=Y	Assignment Statement

The code G is the most powerful command. It allows the program to develop substitutions from short words and pattern words. When all possible substitutions have been made the cryptogram is listed with all of the current substitutions.

The code I is used to get an instruction listing. The list describes the various codes allowed and the Console Entry Switch (CES) settings allowed.

The code NEW is used to read a new cryptogram. All of the tables are reset and a new cryptogram is read in on the card reader.

The code R is used to restart processing on the same cryptogram. All of the tables are reset and processing is resumed as if the cryptogram had just been read.

The code S is used to list all of the current substitutions. This code prints the cipher alphabet and the plain alphabet, showing the substitutions that have been made. This can be particularly useful when working on keyword ciphers.

When the S command is used, two lines are printed. Each line is divided into two parts. The first half of the second line is the complete cipher alphabet. The first half of the first line contains those plain alphabet letters that have been substituted thus far. They are printed over the corresponding cipher letters. The letters in the second line that have no letter appearing over them are the cipher letters that have not had plaintext letters substituted for them yet.

The second half of each line contains similar information. The second half of the top line contains the complete plaintext alphabet. The second half of the second line contains those ciphertext letters that have plaintext letters substituted for them. They are printed under the corresponding plaintext letters. The letters in the top line that do not have letters under them are the plaintext letters that are still available for substitutions.

This command is especially useful in the solution of keyword monoalphabetic substitution ciphers because of the order preserved in the cipher alphabet. Often when only a few substitutions have been made more can be deduced from the list printed. This can best be illustrated with the following examples.

Suppose that the following list is obtained by using the S command.

```
        W J   N   D  G      ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ     S  V  K   O        I
```

Most often the keyword will appear in the second half of the listing. If that is the case in this situation then the S and V with two spaces between them are a strong indication that T=E and U=F are valid substitutions.

After a few more substitutions the list might appear like this:

```
   P    W J   N  ODEFG  L   ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ     STUV  K Y ORD      I
```

In this case the keyword used has become obvious giving the substitutions E=K and W=M. These substitutions lead to more substitutions such as X=H and Z=I. These last two substitutions are developed again from the order of the letters in the cipher alphabet.

The code W is used to list possible words. The code is used in the form Wnn where nn is a one or two digit number specifying the word for which the possibilities should be listed. This code only works for two, three, and four-letter words and pattern words. Only those words that fit when all of the current suostitutions have been made are listed.

The W code can be very helpful in some cases. Suppose, for example, that one segment of a partial solution to a cryptogram is ". . . MEN AN_ WOMEN . . ." Using the W command on the second word of the segment yields two possibilities, "and" and "any". Although the program cannot differentiate between the two words, to the human user the word "and" is obviously much more likely in this context. Even in less clear cut cases the W code can often be useful.

The final command is the assignment statement. It is of the form X=Y where X is any ciphertext letter and Y is the plaintext letter to be associated with it. The Y may be blank to "undo" an assignment. The substitutions made by the program are also printed in this form.

The Console Entry Switches (CES) can also be used to control various operations of the program. The use of these switches is not required. Their use is merely to eliminate various functions if that is desired. The switches with their uses are listed in Table III and in the following paragraphs.

CES 0 can be used to inhibit the pattern word routines. If it is on during the reading of a cryptogram then the possible words for the pattern words will not be listed. If it is on when a G code is entered then no substitutions will be generated from pattern words.

CES 2 can be used to inhibit the two-letter word routine. If it is on when a G code is entered then no substitutions will be generated from two-letter words.

CES 3 can be used to inhibit the three-letter word routine. If it is on when a G code is entered then no substitutions will be generated from the three-letter words.

CES 4 can be used to inhibit the four-letter word routine. If it is on when a G code is entered then no substitutions will be generated from four-letter words.

CES 14 can be used to prevent the listing of the instructions when the program is started. The switch must be set before the printing starts, to be effective. It will not inhibit the I code if it is used during processing.

TABLE III

CONSOLE ENTRY SWITCH USES

CES NUMBER	USE
0	Pattern Word Routine
2	Two-Letter Word Routine
3	Three-Letter Word Routine
4	Four-Letter Word Routine
14	Instruction Listing

CHAPTER VI

SUPPLEMENTAL PROGRAMS

In addition to the main system there are two other programs involved. One is used to build the files of short words. The other is used to build the pattern words file.

There are three files of short words. One contains two-letter words; one contains three-letter words; and one contains four-letter words. All three files have the same format. Each file actually consists of three arrays. The arrays are set up to impart a Multilist organization to the files. The Multilist structure consists of a directory to facilitate accessing the file. The directory entries consist of a key, in this case a specific letter in a specific position, and a pointer to the first relevant entry in the file. Each entry also contains the number of words matching the condition. After the first word is accessed, subsequent words are located using linked lists set up in the file. This organization permits the desired words to be located in the file very quickly.

The third array contains the actual file. It also contains some link pointers to indicate different orders of the words. The first link pointer with each word links the words in alphabetical order. The second link pointer with each word links the words in alphabetical order by their second letters. With the three and four-letter word files a third link pointer with each word links the words in alphabetical order by their third letters. The four-letter word file contains a fourth link pointer with each word. It connects the words in alphabetical order by their fourth letters.

The first array contains pointers for finding words with a given letter in a given position. The second array contains entries giving the number of words containing a given letter in a given position. These two arrays are used to locate the words containing the given letter in the given position.

This can best be illustrated by an example. To keep it simple a two-letter word will be used. The two-letter word tables are shown in Figure 1. If the word M_ is encountered the letter M and its position, first, are used with the first table to find the two-letter words fitting that form.

Using the START table go down to the M row. Since the letter M occurs in the first position of the word take the first number in the M row, 15. The first word in the two-letter word file of the form M_ is word number 15. The corresponding entry from the NUMBER table is 2. Therefore, there are two words starting with the letter M in the table.

Final processing occurs in the WORDS table. It starts with the fifteenth entry. The fifteenth entry in the WORDS

	START		NUMBER		WORDS			
A	1	6	5	0	AM	2	2	1
B	6	6	2	0	AN	3	12	2
C	8	6	0	0	AS	4	13	3
D	8	6	1	0	AT	5	14	4
E	9	6	0	4	AX	6	7	5
F	9	11	0	2	BE	7	10	6
G	9	1	1	0	BY	8	16	7
H	10	1	1	0	DO	9	9	8
I	11	1	4	0	GO	10	17	9
J	15	1	0	0	HE	11	15	10
K	15	1	0	0	IF	12	18	11
L	15	1	0	0	IN	13	19	12
M	15	1	2	1	IS	14	24	13
N	17	2	1	3	IT	15	5	14
O	18	8	3	5	ME	16	25	15
P	21	23	0	1	MY	17	0	16
Q	21	20	0	0	NO	18	21	17
R	21	20	0	1	OF	19	1	18
S	21	3	1	3	ON	20	8	19
T	22	4	1	2	OR	21	3	20
U	23	5	2	0	SO	22	22	21
V	25	5	0	0	TO	23	23	22
W	25	5	1	0	UP	24	20	23
X	25	5	0	1	US	25	4	24
Y	25	7	0	2	WE	0	11	25
Z	25	25	0	0				26

Figure 1. Two-letter word file

table is the word ME. Since there is more than one entry desired, the next one can be found by using the link pointers following the word ME. Since the first letter is the known letter the first link pointer is the one to be used. It points to word number 16 which is MY. Thus the two possible words fitting M_ are ME and MY.

A similar process is used when the second letter is the given letter. The only difference is that the second numbers and link pointers are used instead of the first ones.

The same type of arrays and lookup techniques are used for the three-letter and four-letter word files also. There are just additional columns added for the additional letters.

This system of word lookup has the obvious advantage of being very fast. It is also equally fast with any possibility. The position of a word in the file does not affect the time required to find it.

The pattern word file has a simpler structure. It is a sequential file organized as shown in Figure 2. The file is in sequence by the length of the pattern words. That is, the shorter words come first. Within each group of a given length, the words and patterns are arranged in descending order by their patterns as described earlier. Because of the varying number of words fitting a given pattern and the varying size of the patterns, a variable length record scheme is used. This gives much better disk space utilization than a fixed length record scheme would.

Because of the sequential organization of the pattern word file a sequential search must be used to find a desired pattern. This means that the search time depends on the size of the pattern. However, since the file is in ascending sequence, the short words occur first and, as a general rule, more short pattern words occur than long pattern words.

LENGTH	PATTERN	NUMBER	WORDS
6	001001	4	CREATE, TWELVE, URGING, WORKER
6	000110	4	ASLEEP, REALLY, SCHOOL, PRETTY
6	000101	1	RAISES
6	000011	2	ACROSS, UNLESS
7	1233201	1	SORROWS
7	1231023	1	PREPARE
7	1221002	1	ESSENCE
7	1221000	3	AFFAIRS, ATTACKS, OPPOSED
7	1202101	1	DIVIDED

Figure 2. Section of the pattern word file

The program that builds the pattern word file uses a card file for input. The cards contain the pattern and the words matching that pattern. The program does some error checking during the building of the file to guarantee the sequence and to assure that the words match the patterns.

CHAPTER VII

LIMITATIONS AND EXPANSION

The system is currently implemented on an IBM 1130 computer. It is written in A.N.S.I. Standard Basic Fortran(11) except for the nonstandard DATA and disk I/O statements required for the 1130. For that reason it should be fairly easy to implement it on a different machine. The extensive use of the Console Entry Switches should be no problem because they are all accessed through a subroutine called DATSW which can be substituted on a different machine. If no external switches were available the subroutine could be dummied and the programs would still function properly. The Console Entry Switches are only used to eliminate functions, so their absence would not hinder the programs.

The relatively small 8k memory of this 1130 has forced several restrictions in the programs. The restrictions include limits on table sizes and structuring of the program.

The main system consists of a main program and seven subroutines. In order to fit into memory, the subroutines have to be overlaid. This is accomplished entirely through the Job Control Language for the 1130 without the use of any special Fortran statements such as CALL LINK. If more memory were available these subroutines could all be resident in memory, which would allow the program to run much faster.

The short word table sizes are also limited by the memory size. The routines are implemented with room for 54 two-letter words, 80 three-letter words, and 174 four-letter words. The particular figures are based on the size of available memory and also the optimum utilization of disk space to store the files. While the space for the two-letter words is adequate, the three and four-letter word tables should be bigger. That is, there are more words than there is space available now.

With a larger machine there are several other things that could be added for further improvement. Tables of five, six, and possibly even seven-letter words could be added to the vocabulary of the program. There are several other aids that could also be added. These include contact tables showing occurrences of letters immediately preceding and following each letter, position tables showing relative letter positions within words, and sequence tables showing all trigrams and terminal digrams.

There are also a couple of processing changes that would speed up processing and reduce the number of incorrect substitutions. The number of incorrect substitutions could be reduced by changing the order of processing words in a cryptogram. Instead of just taking them in order all of the two-letter words should be processed first; then all of the three-letter words should be processed and so forth. When a substitution is made, processing should start over at the two-letter words again. This will reduce the number of incorrect substitutions because observation has shown that the shorter word tables are more complete than the longer word tables and the pattern word file. This is especially true with the current limited table sizes.

The processing time could be reduced by introducing a two-level store concept for the pattern word file processing. The first time the pattern word file is accessed the relevant records could be saved in a secondary storage, either on disk or in high speed memory. Then subsequent references would be to the smaller file containing only the relevant patterns and words. As the pattern word file continues to get larger, this would become more and more significant.

A final improvement, probably the most extensive of all, would be dynamic table updating. With dynamic table updating a code could be entered following a successful solution, and any words not currently in the tables or pattern word file would be automatically added. This would eliminate the manual updating process and keep all of the tables and file up to date.

CHAPTER VII

SUMMARY AND CONCLUSIONS

The solution of monoalphabetic substitution ciphers, particularly with short text length, is not a well-defined process. Much of the success or failure of a solution depends on an individual's memory and his ability to recall specific words. The programs described above are designed to be an aid in these areas.

The programs have proven to be extremely successful. Over the course of more than six months they have been used on cryptograms from the "Daily Cryptoquote" feature in the newspaper. In at least 95 percent of the cases they have been helpful to one degree or another. In some cases they have been able to achieve a nearly complete solution without any user intervention.

All of the methods described thus far utilize only one word at a time. In the manual solution process it is sometimes necessary to consider two or more words concurrently to narrow down the range of possibilities and, eventually, to arrive at the solution. Thus an area for future work is to develop algorithms to duplicate this process.

SELECTED BIBLIOGRAPHY

(1) Bryan, William G. *Practical Cryptanalysis Vol. IV*. American Cryptogram Association (1967).

(2) Chesson, Frederick W. "Computers and Cryptology." *Datamation*, XIX (January, 1973), 62-64, 77, 80-81.

(3) Doyle, Sir Arthur Conan. "The Adventure of the Dancing Men." *The Annotated Sherlock Holmes*. ed. William S. Baring-Gould. New York: Clarkson N. Potter, 1967.

(4) Edwards, D.J. *OCAS-On-line Cryptanalytic Aid System*. Massachusetts Institute of Technology (1966).

(5) Feistal, Horst. "Cryptography and Computer Privacy." Scientific American, CCXXVIII (May, 1973), 15-23.

(6) Friedman, W.F. *Elements of Cryptanalysis*. Washington: Government Printing Office, 1924.

(7) Gaines, Helen Fouché. *Cryptanalysis*. New York: Dover, 1956.

(8) Hammer, Carl. "Signature Simulation and Certain Cryptographic Codes." Communications of the ACM, XIV (January, 1971), 3-14.

(9) Harris, Frances A. *Solving Simple Substitution Ciphers*. American Cryptogram Association (1943).

(10) *IBM 1130 Disk Monitor System, Version 2, Programmer's and Operator's Guide*. GC26-3717-9, International Business Machines Corporation (May, 1972).

(11) *IBM 1130/1800 Basic Fortran IV Language*. GC26-3715-8, International Business Machines Corporation (January, 1973).

(12) Kahn, David. *The Codebreakers*. New York: Macmillan, 1967.

(13) Mellen, G. E. "Cryptology, Computers, and Common Sense." AFIPS Conference Proceedings, Vol. 42 (1973), 569-579.

(14) Meyer, D. H. "Design Considerations for Cryptography." AFIPS Conference Proceedings, Vol. 42 (1973), 603-606.

(15) Minot, O. N. and Neil Macdonald. "Communications and Ciphers." Computers and Automation (September, 1971), 36-38.

(16) Minot, O. N., R. A. Sobieraj and K. D. Streetman. "Computers, Ciphers, and Cryptography." Computers and Automation, XXI (February, 1972), 47-48.

(17) Ohaver, M. E. Cryptogram Solving. Etcetera Press (1973).

(18) Poe, Edgar Allen. "The Goldbug." Complete Stories and Poems of Edgar Allen Poe. Garden City: Doubleday & Company, 1966.

(19) Pratt, Fletcher. Secret and Urgent, The Story of Codes and Ciphers. Idianapolis: Bobbs-Merrill, 1939.

(20) Schatz, Bruce R. "Automated Analysis of Cryptograms." Cryptologia, I (April, 1977), 116-142.

(21) Shannon, C. E. "Communication Theory of Secrecy Systems." Bell System Technical Journal, XXVIII (October, 1949), 656-715.

(22) Smith, Laurence Dwight. Cryptography. New York: Dover, 1955.

(23) Stahl, Fred A. "A Homophonic Cipher for Computational Cryptography." AFIPS Conference Proceedings, Vol. 42 (1973), 565-568.

(24) The Cryptogram, published bimonthly by the American Cryptogram Association, 9504 Forest Road, Bethesda, MD 20014.

(25) Wiltbank, Henry C., Lewis S. Sutliff and Lisle J. Maxson. Three Ways of Solving Cryptograms. American Cryptogram Association (1963).

(26) Yardley, Herbert O. The American Black Chamber. Indianapolis: Faber & Faber Limited, 1931.

VITA 2

Gerald Dale Smith

Candidate for the Degree of

Master of Science

Thesis: COMPUTER-AIDED SOLUTION OF MONOALPHABETIC SUBSTITUTION CIPHERS

Major Field: Computing and Information Sciences

Biographical:

Personal Data: Born in Enid, Oklahoma, August 13, 1948, the son of Mr. and Mrs. Wilbur E. Smith

Education: Graduated from Enid High School, Enid, Oklahoma, in May, 1966; received Bachelor of Science degree in Mathematics from Oklahoma State University in May, 1970; completed requirements for the Master of Science degree at Oklahoma State University in July, 1978.

Professional Experience: Part-time applications programmer, Champlin Petroleum Company, 1967-1970; applications programmer, Champlin Petroleum Company, 1970-1972; part-time applications programmer, Champlin Petroleum Company, 1972-1973; software systems programmer, Champlin Petroleum Company, 1974-present.

Professional Organizations: Association for Computing Machinery.