A SOFTWARE CIPHER SYSTEM FOR PROVIDING SECURITY FOR COMPUTER DATA

by

JOHN CLEVE WALKER

B.S., University of Detroit, 1971

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1977

Approved by:

Richard L. Sincovec
Major Professor

TABLE OF CONTENTS

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

Chapter 1

INTRODUCTION

1. The purpose of this paper is to demonstrate that a cipher system can be designed for software implementation on a computer. Research of this topic revealed that several hardware cipher systems have been produced. The price of these devices ranges from 4,000 to 10,000 dollars. Some research papers discuss proposed software cipher systems, however, none of these are used in practical applications[1]. Additionally, very little information exists on the security of these software cipher systems. A cipher system is an algorithm or an electromechanical device which receives intelligible data as input and performs a transformation or substitution on the input. This action produces unintelligible data as output. These algorithms and hardware devices can be simulated on a computer. Once applied to a computer, these systems can interface with a data base management system to encipher selected data.

2. The proposed cipher system must provide good security for the data to be enciphered and accept data which is represented in the American Standard Code for Information Interchange (ASCII) character code. The security provided by the system should be such that the cost of trying to break the code would be more than the value of the information to be obtained. Another consideration is that the system should be designed for easy employment by the user. In order to meet the above objectives, it was necessary to conduct the research in three phases. First, to evaluate a cipher system's security, a review of cryptanalysis techniques was necessary. All cipher systems appear to provide good security if one has no knowledge of cryptanalysis

techniques. Second, a review and evaluation of various cipher systems was conducted, some of which is listed in Chapter 3. Finally, the selection of one cipher system was made and the system was designed into a software program. For clarity of terminology used within this report a list of definitions is contained in Appendix 2.

Chapter 2

WEAKNESSES OF CIPHER SYSTEMS

1. In the process of studying cipher systems, it behooves one to also study the techniques of cryptanalysis. Thus, the weaknesses of certain cipher systems become readily apparent. It would be foolhardy to study cipher systems and design one for use on computers without studying how easily a potential intruder can break these systems. In discussing some of the more common cryptanalysis techniques, the example below will be referred to to clarify the techniques. Additionally it should be noted that a cryptanalyst would have much more ciphertext to work on than the small amount below. Most of the techniques below can be programmed for use on computers which reduces the time to break the ciphers.

Sample Cipher System

Plaintext alphabet	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Ciphertext alphabet	HARPBJMKCTWZDQELUYFXGIVNOS

Sample of Enciphered Data

Data to be enciphered	THE PROGRAM HAS COMMENTS AND IS IN FORTRAN
Enciphered data	XKB EYLMYHD KHF RLDDBQXF HQP CP CQ JLYXYHQ

a. Single Character Frequency Distribution. The usage of the different letters in a language readily stands out after conducting a frequency count (see Appendix 1). For example, the letter "e" appears 13.05 percent of the time, whereas, the letters "k", "j", and "q" appear less than 0.5 percent. If a cryptanalyst had a lengthy ciphertext which used the cipher key above, he would find that the cipher letter "b" would appear about 13 percent of the time. Thus, the assumption

that the cipher letter "b" was equal to the plaintext letter "e" would be a safe assumption. By utilizing this technique, a potential intruder can break most simple codes.

b. Polygram Frequency Distribution. A study of the English language has produced a listing of digrams and trigraphs as they usually appear by frequency in the normal text. For example, the digrams "th", "he", "an", and "in" appear the most frequently (see Appendix 1). The trigrams "the", "and", and "ent" appear the most frequently (see Appendix 1). In the previous sample ciphertext, the substitution of these most frequent polygrams would assist the cryptanalyst in guessing at the true values of ciphertext "XKB", "HQP", and "CQ".[2]

c. Starting and Terminal Letter Frequencies. What letters are the most frequent as the starting and terminal letters of words was also established by previous studies. The letter "t" is the most frequent initial letter of a word, whereas, the letters "e" and "s" are the most frequent terminal letters (see Appendix 1).

d. Tables of Words by Lengths. This is nothing more than lists of all words, first by length and secondly, alphabetical. These lists can also be compiled alphabetically by the second letter, third letter, and so on. Thus, if an intruder knows the value of the third letter of a five letter word, the task is simply looking up the list of five letter words listed alphabetically by the third letter. The possibilities are then reduced to just a few words. A good example is the ciphertext previously shown, where "EYLMYHD" equates to "PROGRAM". If the value of "M" was known to be "G", then a list of seven letter words with the fourth letter being "G" would reduce the task.

e. Word Structure Patterns. Similar patterns of letter occurrences exist in certain words and is the principle idea of this technique. To illustrate this consider the words caLLing and paDDing. These words have the pattern of --xx---. Having tables, listing these letter patterns, greatly reduces a cryptanalyst's task. Vast possibilities can be reduced to just a few possibilities by use of these tables. In the previous ciphertext consider pRogRam, coMMents, and foRtRan.

f. Tuckerman's u-differencing Method. This cryptanalysis method concerns the Vernam and the Vigenere ciphers. Basically, numerous statistical tests are conducted on the ciphertext to determine the cipher-key length. Then frequency distribution tests are conducted on columns of ciphertext based on the length of the cipher key. Correlation of these distributions then allows a calculation of the differences between the key characters. Then, the previous cryptanalysis techniques are used to break the cipher. However, all the techniques are automated by use of the computer, thus making the task much easier and less time consuming[11].

2. All of the above techniques are merely a means to derive the cipher key. Therefore, if the key can easily be recovered, then the system is worthless. The entire cipher system should depend on the security of the key. This is vividly illustrated by the fact that one ciphertext, which could lead to vast riches, remains unbroken after more than a hundred years. In 1817, Thomas Beale hid gold and silver in an excavation six feet deep somewhere in Virginia. He left several coded messages which describe the location of the riches. Only one of these messages has been broken, by using the Declaration of Independence as the key. The words were numbered from 1 to 1322 and used to encipher the message. Message number one which gives the exact location of the gold has never been

broken because a different cipher key was used. To date, this cipher message, after attacks by cryptanalysts using computers, still remains unsolved[5]. From this example it should be clear that the cipher key is the heart of any cipher system.

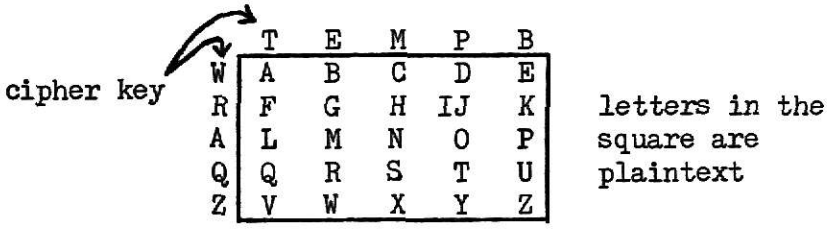
Chapter 3

VARIOUS CIPHER SYSTEMS

By having some knowledge of techniques which can be applied to break ciphers, a better evaluation of various cipher systems can be conducted. Without a basic knowledge of cryptanalysis, all ciphers would appear to provide good security. Listed below are some of the various cipher systems which have been utilized in the past.

a. Monoalphabetic Substitution Ciphers. This cipher is the same as was illustrated in Chapter 2; where only one plaintext alphabet and one ciphertext alphabet exist. Reduction of the ciphertext to plaintext occurrs easily by use of cryptanalysis techniques. This system is also known as Ceasar's Cipher.

b. Matrix Cipher Systems. Variation of this system are many, however, all use the same general scheme. Reducing the ciphertext of these systems to plaintext can be accomplished, but is more time consuming. The cipher key does not change frequently, which in turn is the weakness of the system. These systems essentially are monalphabetic substitution. The word pattern of "SYSTEM" in the illustration relates to the already discussed cryptanalysis methods. Other variations have digits instead of letters in the cipher key.



The ciphertext equivalent of "OPERATING SYSTEM" would be "AP AB WB QE WT QP RP AM RE QM ZP QM QP WB AE".

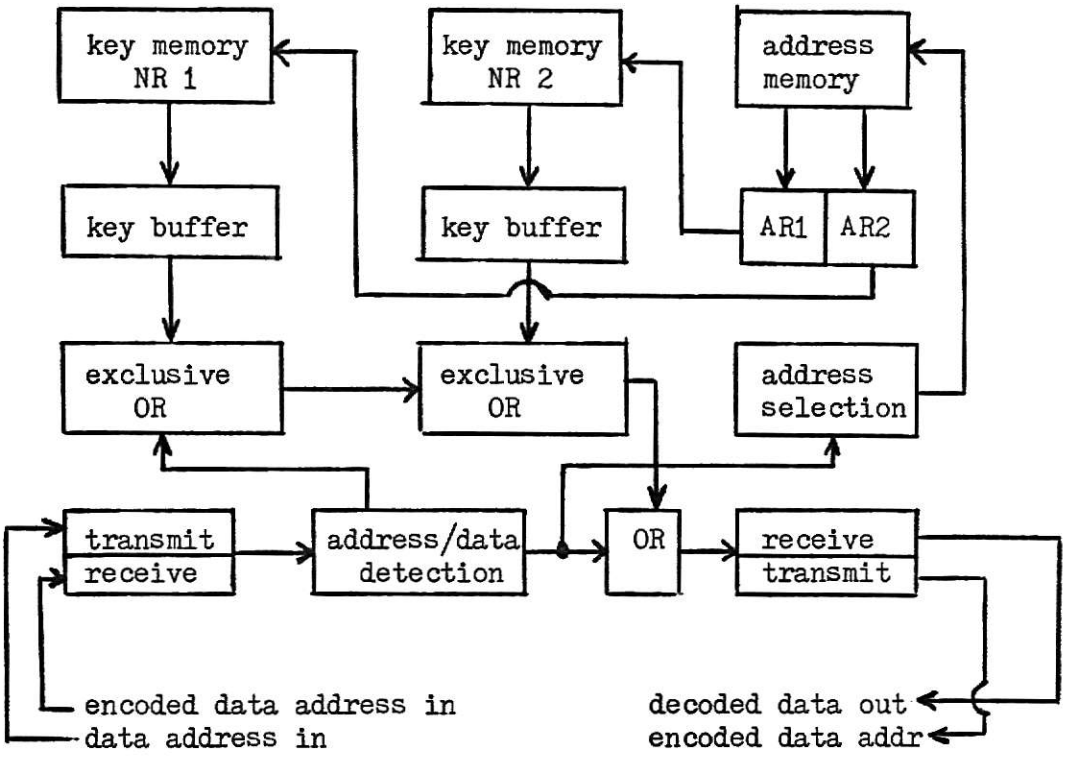
c. Polyalphabetic Substitution System. This system employs more than one alphabet to represent the ciphertext. The plaintext letters appear in the row of letters at the top of the table and the cipher key letters at the side. The more common name for this system is the Vigenere cipher[4]. To encipher the message below, the following steps would be taken. A cipher key word, in this example DATACELL, is repeated over the data to be enciphered. You go down the plaintext column represented by the first letter which is "P". Then come across the row represented by the cipher key letter above the plaintext letter "P" which is "D". The ciphertext letter is the letter where the "P" column and the "D" row intersect, which is "S". This system can be broken, but without the aid of a computer it requires a large amount of time and over 20 million characters of ciphertext. However, with the aid of a computer and using Tuckerman's algorithms for u-differencing, it requires little time to break this complex cipher system since the key is usually short and is repeated over and over.

Cipher key	DATACEL LDAT ACE LLDATAC EL LDATAC
Plaintext	PRIVACY LOCK FOR PAYROLL IS AZTFEM
Ciphertext	SRBVCGJ WRCD FQV ALBRHLN MD LCTYEO

Polyalphabetic Substitution System

	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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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

d. Vernam Cipher System. This system was the first electro-mechanical system. This system used two tapes (cipher keys) which were matched in such a manner to produce a permutation of their contents. The results were then added to produce a secondary cipher key to add with the plaintext to produce a ciphertext. This cipher has many variations, however, the general principle remains the same. A digital substitution variation is shown on the next page[9]. Problems with this system are that medium length keys are used and the storage of all these cipher keys requires considerable computer storage space. Depending on the cipher key, most of these variations can be broken by utilizing Tuckerman's u-differencing method.



Example of Ciphering

	data	110110
encoding	key memory NR 1	100110
	1st level code	010000
	key memory NR 2	111011
	encoded data	101011
decoding	key memory NR 1	100110
	1st level decode	001101
	key memory NR 2	111011
	decoded data	110110

Chapter 4

THE PROPOSED MODIFICATION OF THE VIGENERE CIPHER SYSTEM

1. In selecting a cipher system, all of the previous systems plus many others were evaluated. Additionally, each system's weaknesses were taken into consideration. How the systems could be improved upon was also considered. The cipher system finally selected was the Vigenere cipher. The reasons for this selection are as follows:

a. With modifications, this system provides a degree of security which is much greater than any other.

b. With a modification to the matrix module, the system has the capability to utilize a matrix with a random composition. This differs greatly from the standard Vigenere which uses the normal sequence of the alphabet.

c. The cipher key was changed and is now a sequence of random numbers rather than a short key as used previously.

2. The modifications to the standard Vigenere cipher provide increased security and are explained in detail below.

a. The cipher key for the system consists of a sequence of random numbers generated by two random number generators. Even if the algorithm for this cipher is known, an intruder would have to derive two nine digit keys which are used to seed the generators. The keys must be nine digits and odd, therefore over 400 million different possibilities exist for each key for each random number generator.

b. The ciphertext alphabets in the standard Vigenere are not changed, which is a weakness of that system. In the proposed model, the

system is designed to allow the ciphertext alphabets to be changed periodically. Also, the ciphertext alphabets are generated randomly. An additional problem is now created for the intruder. The configuration of the matrix must be solved. A total of three keys must be determined, which greatly increases the cost factor for the intruder. Significant amounts of manpower and computer time would have to be expended to derive the three keys.

c. Another security factor is that the plaintext in the modified system is not located in the first row of the matrix. In this system, the plaintext row will be selected by the first random number generated, since all the rows contain all of the ASCII characters.

3. The modified Vigenere cipher system will perform as follows (see Appendix 3, Model Diagram and Appendix 4, Fortran Program of Model). The random number generators utilized in this model are based on the model developed by Shan S. Kuo[7]. The assumption was made that Kuo's model meets the requirements for a random number generator as specified by Donald E. Knuth. Knuth recommends that random number generators be subjected to numerous tests, since a generator may pass some tests, but fail others. Specifically, the tests recommended are as follows: equidistribution test, serial test, gap test, partition test, permutation test, run test, subsequence test, serial correlation test, maximum of "t" test, and spectral test[6].

a. Control Module. The control module performs several functions depending on the input. As input, the control module will receive the cipher keys, action command, and data to be enciphered or

deciphered.

1. If the command "111" is received, then the control module calls the encode module and passes control to it.

2. If the command "222" is received, then the control module calls the decode module and passes control to it.

3. If the "333" command is received, the control reverts to the change module.

4. If none of the above commands are received, then an error message is printed out.

5. Global data, which consists of the input array, output array, and the Vigenere matrix, is used to eliminate passing this data to the subroutines as arguments.

b. Encode Module. The encode module receives as arguments the keys for the two random number generators. These in turn are then passed to the random number generators. The random number generators then return the required number of random numbers. An example of the matrix is shown below to simplify the explanation. Assume the first random number received by the encode module is 20. The module notes the character in the first element of the input array and scans row 20 of the matrix to find an equal character. When its position is determined, that position is noted. Let the first character to be enciphered be "F", then its position is noted as 5. The next random number designates a ciphertext alphabet row in which the letter "F" will be enciphered. Assume it is 48, then the module scans position 5 of row 48 and designates "9" as the cipher letter for plaintext letter "F". The "9" is then loaded into the first element of the output array. Assume the third random number is 72,

then row 72 is the ciphertext alphabet for the second letter to be enciphered. If the second letter to be enciphered is "B", then its position is equated to a position in row 72. The cipher letter would be "C". This letter then is loaded into the second element of the output array. This process is repeated until the encipherment is completed, at which time control reverts back to the control module (see Appendix, Sample Input and Output Data). Row 20 is used as the plaintext row during this entire process until new keys are entered.

Sample Matrix

row 20 A W R B F G H ..... 9 0 D 1 C
.
.
.
row 48 Z 2 W 4 9 A M R ..... F Y 7 0 L
.
.
row 72 W X B C M R S 0 Q ..... A 8 F C H

c. Decode Module. The decode module is the reverse of the encode module. The ciphertext letter's position is noted in the ciphertext alphabets. Their positions are matched against positions in the plaintext row to designate the plaintext letters. Then the plaintext letters are loaded into the output array.

d. Change Module. When called, this module will change the cipher alphabets in the matrix. The module will read in as input 128 hexadecimal numbers. The hexadecimal numbers have the same internal representation as the ASCII code characters would have. Each character is then placed in the first row of the matrix, all the while insuring

that no duplication occurrs. Once this is completed, then the remaining rows are copied from the first row, with each row slipping one position. This results in a staggered effect as shown below.

Sample Matrix

row 1 A R S T U 9 a V 2 ..... X B 8 C

row 2 R S T U 9 a V 2 ..... X B 8 C A

row 3 S T U 9 a V 2 ..... X B 8 C A R

4. This system could be utilized to encipher data prior to storage in a data bank. Additionally, use of it could also be made for a computer network. Use in a network environment creates one problem, that of both ends of the network having a requirement for the same cipher keys. The proposed model could be changed to create keys and then transmit them. A higher frequency of transmission will cause a decrease in the efficiency of the computer. The ciphertext produced by this system can withstand cryptanalysis attempts so long as no more than 20 million characters are enciphered by one set of cipher keys[3]. Therefore, to maximize efficiency the cipher keys should be changed only after encipherment of each requested block of data. Additionally, this decreases the number of keys which must be used. The keys could be stored in the computer and fetched when required. However, if the keys are transmitted over the network, then an intruder could maximize his effort toward breaking the first ciphertext. Once, this has been achieved, then all other enciphered data is no problem since he will have the transmitted keys to break this data.

Chapter 5

SUMMARY

1. Prior to making a decision to use a software package to provide security for data, several evaluations should be conducted.

a. Determine if the information to be protected is worth the cost of the computer time required to encipher it.

b. Determine if it is worth the decrease in efficiency.

c. Determine if this information can be obtained elsewhere. All the security in the world is worth nothing, if an intruder can get the same information in an office or trash can. If the cost ratio of the security is high enough, then the intruder will go elsewhere to obtain the same information. If he succeeds, then the elaborate cipher system is for naught.

2. Design of this system was based on the assumption that an intruder would have not only computer resources, manpower, and experience, but also the algorithm for this system. Therefore, the heart of this system is based on the cipher keys. One would have to derive three cipher keys in order to break ciphertext of less than 20 million characters. A task of this size should increase the amount of resources required by an intruder to such a point that it would discourage him. In other words, the value of the information to be obtained would not be worth the cost of the resources required to obtain that information.

3. Most of the time spent on this report was devoted to research to determine the following:

a. What is a cipher system?

b. What are the various cipher systems available?

c. What are the weaknesses of the cipher systems?

d. What cipher systems are being employed or proposed for computer applications?

Two publications required a considerable amount of time due to their volume and complexity. These were the books by Kahn and the report by Tuckerman. Considerable time was spent researching publications in the library, which produced little information. Therefore, two reports by IBM were ordered, however, only the report by Tuckerman arrived in time for referencing in this report[11]. Future work in this area should be conducted by several people. The job of testing a random number generator should be a full time task for one person. Once a cipher system is developed, based on a verified random number generator, then the cipher system should be verified using Tuckerman's cryptanalysis techniques. At least two more people would be required to design and implement all the algorithms discussed in Tuckerman's report. Consideration should also be given to the fact that this will require a vast amount of computer time. Once this has been accomplished, then the cipher system would meet all the requirements of being the perfect cipher system.

REFERENCES

1. Feistel, H., "Cryptography and Computer Privacy", Scientific American, May 1973, pp 15-23.

2. Gaines, H. F., Cryptanalysis, Dover Publications, Inc, New York, N. Y., 1956.

3. Girsdansky, M. B., "Cryptology, The Computer and Data Privacy", Computers and Automation, April 1972, pp 12-19.

4. Hoffman, L. J., Security and Privacy in Computer Systems, Melville Publishing Co., Los Angeles, California, 1973.

5. Kahn, D., The Codebreakers, The MacMillian Co., New York, N. Y., 1967.

6. Knuth, D. E., The Art of Computer Programming, Addison-Wesley Publishing Co., Reading, Massachusetts, 1969.

7. Kuo, Shan S., Computer Applications of Numerical Methods, Addison-Wesley Publishing Co., Reading, Massachusetts, 1972.

8. Minot, O. N., "Communication and Ciphers with a Hexadecimal Alphabet and Variations", Computers and Automation, Sept 1971, pp 36-38.

9. Skatrud, R. O., "The Applications of Cryptographic Techniques to Data Processing", Proc. AFPIS 1969 FJCC, Vol. 34, AFIPS Press, Montvale, N. J., pp 111-117.

10. Winkler, S. and Danner, L., "Data Security in the Computer Communication Environment", Computer, Feb 1974, pp 23-31.

11. Tuckerman, B., A Study of the Vigenere-Vernam Single and Multiple Loop Enciphering Systems, IBM Corporation, RC 2879, May 1970.

APPENDIX 1

1. Single letter frequencies (per 100).

E	13.05	D	4.11	G	1.39
T	9.02	L	3.60	B	1.28
O	8.21	C	2.93	V	1.00
A	7.81	F	2.88	K	.42
N	7.28	U	2.77	X	.30
I	6.77	M	2.62	J	.23
R	6.64	P	2.15	Q	.14
S	6.46	Y	1.51	Z	.09
H	5.85	W	1.49		

2. Order of digrams and trigrams.

Digrams (per 10,000)				Trigrams
TH 315	TO 111	SA 75	MA 56	THE
HE 251	NT 110	HI 72	TA 56	AND
AN 172	ED 107	LE 72	CE 55	THA
IN 169	IS 106	SO 71	IC 55	ENT
ER 154	AR 101	AS 67	LL 55	ION
RE 148	OU 96	NO 65	NA 54	TIO
ES 145	TE 94	NE 64	RO 54	FOR
ON 145	OF 94	EC 64	OT 53	NDE
EA 131	IT 88	IO 63	TT 53	HAS
TI 128	HA 84	RT 63	VE 53	NCE
AT 124	SE 84	CO 59	NS 51	EDT
ST 121	ET 80	BE 58	UR 49	TIS
EN 120	AL 77	DI 57	ME 48	OFT
ND 118	RI 77	LI 57	WH 48	STH
OR 113	NG 75	RA 57	LY 47	MEN

3. Initial letters of words.

Order, as found by M. E. Ohaver[2]: T A O S H I W C B P F D M R etc.
Order, as found by H. O. Yardley[2]: T O A W B C D S F M R H I Y etc.

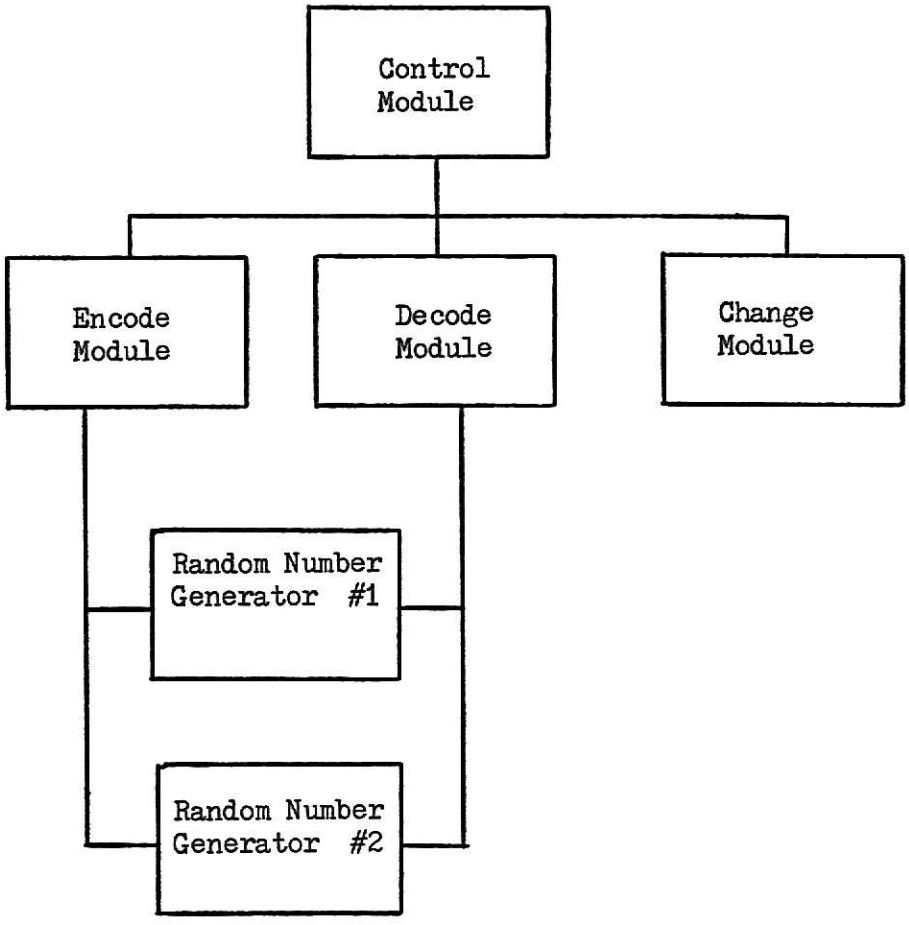
4. Final letters of words.

Order, as found by M. E. Ohaver[2]: E S T D N R O Y etc.
Order, as found by H. O. Yardley[2]: E T D N S R Y etc.

APPENDIX 2

Cipher	any system in which arbitrary symbols or groups of symbols represent units of plaintext of regular length, usually single letters, or in which units of plaintext are rearranged, or both, in accordance with certain predetermined rules.
Ciphertext	unintelligible text or signals produced through the use of cipher systems.
Cryptanalysis	study of encrypted texts; the steps or processes involved in converting encrypted text into plaintext without initial knowledge of the key employed in the encipherment.
Decipher	to convert enciphered text into its equivalent plaintext by means of a cipher system.
Encipher	to convert a plaintext into unintelligible form by means of a cipher system.
Enciphered text	text which conveys no intelligible meaning, or an intelligible meaning that is not the real meaning but actually conveys a hidden meaning.
Key	a symbol or sequence of symbols which controls the operation of encipherment and decipherment.
Plaintext	intelligible text or signals which have meaning and which can be read or acted upon without the application of decipherment.

APPENDIX 3


Control Module
Encode Module
Decode Module
Change Module
Random Number Generator #1
Random Number Generator #2

APPENDIX 4

```
C
C              MODIFIED VIGENERE CIPHER SYSTEM
C
C     THIS PROGRAM WILL PERFORM SUBSTITUTIONS ON DATA RECEIVED AS INPUT.
C     THE RESULTING OUTPUT WILL BE UNINTELLIGIBLE UNTIL PROCESSED THRU
C     THIS PROGRAM USING THE SAME CIPHER KEY. INPUT IS HEXADECIMAL
C     NUMBERS TO SIMULATE ASCII CHARACTER CODE.
C
C     THE DATA DECK MUST BE READ IN THE FOLLOWING ORDER:
C
C          ASCII CHARACTER CODE TO INITIALIZE THE MATRIX
C          SYMBOLS USED TO INITIALIZE THE END ARRAY
C          SYMBOLS USED TO INITIALIZE THE STOP ARRAY
C          CIPHER KEYS USED AS SEEDS FOR GENERATORS
C          SYMBOLS TO INDICATE THE TYPE OF TASK
C          INPUT DATA IN FORM OF ASCII CHARACTER CODE
C          SYMBOLS TO STOP THE PROGRAM OR END ENCIPHERMENT
C
C     VARIABLE DEFINITIONS
C
C          IEND      ARRAY USED TO STORE SYMBOLS TO END ENCIPHERMENT
C                    USING ONE SET OF CIPHER KEYS IN ORDER FOR A NEW
C                    SET OF CIPHER KEYS TO BE LOADED
C
C          ISTOP     ARRAY USED TO STORE SYMBOLS TO STOP THE
C                    PROGRAM. WHEN THE SYMBOLS IN THIS ARRAY
C                    MATCH THE FIRST THREE SYMBOLS IN INPUT
C                    ARRAY THEN THE PROGRAM STOPS
C
C          IIKEY1    SEED FOR THE FIRST RANDOM NUMBER GENERATOR
C                    RAND1
C
C          IIKEY2    SEED FOR THE SECOND RANDOM NUMBER GENERATOR
C                    RAND2
C
C          ITASK     ACTION TO BE TAKEN: IF ITASK EQUALS
C
C                         111  THE INPUT DATA IS ENCIPHERED
C                         222  THE INPUT DATA IS DECIPHERED
C                         333  A NEW MATRIX IS LOADED
C
      COMMON INPUT (128), OUTPUT (128), MATRIX (128,128)
      INTEGER*2 INPUT, OUTPUT, MATRIX IEND(3), ISTOP(3)
      CALL CHANGE
      DO 05 I=1,3
05    READ 40, IEND(I)
      DO 06 I=1,3
06    READ 40, ISTOP(I)
```

```
15    READ 20, IIKEY1, IIKEY2
      READ 30, ITASK
35    DO 36 I=1,128
36    READ 40, INPUT(I)
      IF(INPUT(1).EQ.IEND(1).AND.INPUT(2).EQ.IEND(2).AND.
     1INPUT(3).EQ.IEND(3))GO TO 15
      IF(INPUT(1).EQ.ISTOP(1).AND.INPUT(2).EQ.ISTOP(2).AND.
     1INPUT(3).EQ.ISTOP(3))GO TO 80
      IF(ITASK .EQ. 111) GO TO 50
      IF(ITASK .EQ. 222) GO TO 60
      IF(ITASK .EQ. 333) GO TO 65
      PRINT 71
71    FORMAT(1H0,'ERROR')
      GO TO 80
50    CALL ENCODE(IIKEY1, IIKEY2)
      PRINT 70, (OUTPUT(I),I=1,64)
      PRINT 70, (OUTPUT(I),I=65,128)
70    FORMAT(1H0,(64(Z2)))
      GO TO 35
60    CALL DECODE(IIKEY1, IIKEY2)
      PRINT 70, (OUTPUT(I),I=1,64)
      PRINT 70, (OUTPUT(I),I=65,128)
      GO TO 35
65    CALL CHANGE
      GO TO 15
20    FORMAT(I9,I9)
30    FORMAT(I3)
40    FORMAT(Z2)
80    STOP
      END
C
C
C              SUBROUTINE ENCODE
C
C
C     THIS SUBROUTINE ENCIPHERS THE INPUT DATA AND STORES THE
C     CIPHERTEXT IN THE OUTPUT ARRAY. THE RANDOM NUMBER GENERATORS
C     ARE USED TO INDICATE WHAT ROWS IN THE MATRIX WILL BE USED.
C     THE FIRST NUMBER WILL INDICATE WHAT ROW WILL BE USED FOR THE
C     PLAINTEXT ROW. THE OTHER NUMBERS GENERATED WILL INDICATE THE
C     ROW TO BE USED AS THE CIPHERTEXT ROW. ONCE THE INPUT CHARACTER
C     IS MATCHED IN THE PLAINTEXT ROW, ITS POSITION IS MATCHED IN THE
C     CIPHERTEXT ROW. THE CIPHER CHARACTER IN THAT POSITION IS THEN
C     STORED IN THE OUTPUT ARRAY.
C
C     VARIABLE DEFINITIONS
C
C          KEY11          SEED FOR FIRST RANDOM NUMBER GENERATOR
C                         RAND1
```

```
C
C          KEY22          SEED FOR SECOND RANDOM NUMBER GENERATOR
C                         RAND2
C
C          REKEY1 & 2     INTEGERS RETURNED FROM THE RANDOM
C                         NUMBER GENERATORS FOR THE PURPOSE OF
C                         RESEEDING THE GENERATORS
C
C          NUMB           RANDOM NUMBER RESULTING FROM ADDING THE
C                         OUTPUT OF BOTH GENERATORS
C
C          NUMB1          USED TO DESIGNATE THE PLAINTEXT ROW IN
C                         THE MATRIX
C
C          NUMB2          USED TO DESIGNATE CIPHERTEXT ROWS IN
C                         THE MATRIX
C
C          NOUNT          POINTER TO ELEMENTS IN THE INPUT ARRAY
C
      SUBROUTINE ENCODE(KEY1, KEY2)
      COMMON INPUT(128), OUTPUT(128), MATRIX(128,128)
      INTEGER*2 INPUT, OUTPUT, MATRIX
      INTEGER REKEY1, REKEY2, RAND11, RAND22
      KEY11=KEY1
      KEY22=KEY2
      ICOUNT=0
      NOUNT=1
10    CALL RAND1(KEY11, REKEY1, RANDM1)
      RAND11=RANDM1*64
      KEY11=REKEY1
      CALL RAND2(KEY22, REKEY2, RANDM2)
      RAND22=RANDM2*64
      KEY22=REKEY2
      NUMB=RAND11 + RAND22
      IF(NUMB .EQ. 0) GO TO 10
      ICOUNT = ICOUNT + 1
      IF(ICOUNT .EQ. 1) NUMB1 = NUMB
      IF(ICOUNT .EQ. 1) GO TO 10
      IF(NUMB .EQ. NUMB1) GO TO 10
      NUMB2 = NUMB
      DO 50 I=1,128
      IF(INPUT(NOUNT) .EQ. MATRIX(NUMB1,I)) GO TO 60
50    CONTINUE
60    OUTPUT(NOUNT) = MATRIX(NUMB2,I)
      NOUNT = NOUNT + 1
      IF(NOUNT .EQ. 129) GO TO 70
      GO TO 10
70    RETURN
      END
```

```
C
C
C                        SUBROUTINE DECODE
C
C
C     THIS SUBROUTINE DECIPHERS THE CIPHERTEXT RECEIVED AS INPUT.
C     THE RANDOM NUMBER GERERATORS ARE USED IN THE SAME MANNER
C     AS IN THE ENCODE SUBROUTINE. THE DIFFERENCE IN THIS
C     SUBROUTINE IS THAT THE POSITION OF THE CIPHER LETTER IS
C     NOTED AND MATCHED AGAINST THE PLAINTEXT ROW. THE
C     DESIGNATED LETTER IN THE PLAINTEXT ROW IS THEN LOADED
C     IN THE OUTPUT ARRAY.
C
C     VARIABLE DEFINITIONS
C
C          KEY11          THIS MUST BE THE SAME KEY WHICH WAS
C                         USED TO ENCIPHER THE DATA RECEIVED
C
C          KEY22          SAME AS ABOVE, ADDITIONALLY THE MATRIX
C                         MUST BE IN THE SAME CONFIGURATION AS
C                         IT WAS WHEN THE RECEIVED DATA WAS
C                         ENCIPHERED
C
C          NUMB           SAME AS IN SUBROUTINE ENCODE
C
C          NUMB1          SAME AS IN SUBROUTINE ENCODE
C
C          NUMB2          SAME AS IN SUBROUTINE ENCODE
C
C          NOUNT          POINTER TO ELEMENTS OF THE INPUT AND
C                         OUTPUT ARRAYS
C
      SUBROUTINE DECODE(KEY1, KEY2)
      COMMON INPUT(128), OUTPUT(128), MATRIX(128,128)
      INTEGER*2 INPUT, OUTPUT, MATRIX
      INTEGER REKEY1, REKEY2, RAND11, RAND22
      KEY11 = KEY1
      KEY22 = KEY2
      ICOUNT = 0
      NOUNT = 1
10    CALL RAND1(KEY11, REKEY1, RANDM1)
      RAND11 = RANDM1 * 64
      KEY11 = REKEY1
      CALL RAND2(KEY22, REKEY2, RANDM2)
      RAND22 = RANDM2 * 64
      KEY22 = REKEY2
      NUMB = RAND11 + RAND22
      IF(NUMB .EQ. 0) GO TO 10
      ICOUNT = ICOUNT + 1
      IF(ICOUNT .EQ. 1) NUMB1 = NUMB
```

```
      IF(ICOUNT .EQ. 1) GO TO 10
      IF(NUMB .EQ. NUMB1) GO TO 10
      NUMB2 = NUMB
      DO 50 I=1,128
      IF(INPUT(NOUNT) .EQ. MATRIX(NUMB2,I)) GO TO 60
50    CONTINUE
60    OUTPUT(NOUNT) = MATRIX(NUMB1,I)
      NOUNT = NOUNT + 1
      IF(NOUNT .EQ. 129) GO TO 70
      GO TO 10
70    RETURN
      END
C
C
C                        SUBROUTINE CHANGE
C
C
C     THIS SUBROUTINE IS USED TO INITIALIZE THE MATRIX OR CHANGE
C     THE ORDERING OF THE MATRIX. THE ASCII CHARACTER
C     REPRESENTATIONS ARE READ INTO THE FIRST ROW OF THE MATRIX
C     THEN A DO LOOP IS ENGAGED WHICH WILL FILL IN THE REMAINING
C     ROWS OF THE MATRIX BASED ON THE FIRST ROW. THE RESULTING
C     ROWS WILL BE STAGGERED AS SHOWN BELOW
C
C     EXAMPLE
C
C          ROW 1     A B C D E .....
C          ROW 2     B C D E F .....
C          ROW 3     C D E F G .....
C          ETC
C
C     VARIABLE DEFINITIONS
C
C          M     USED TO KEEP COUNT OF THE ROW BEING CONSTRUCTED
C          N     USED TO DESIGNATE THE COLUMN POSITION TO BE
C                LOADED. THE INTERSECTION OF THE ROW AND COLUMN
C                INDICATE THE ELEMENT OF THE ARRAY (MATRIX) TO
C                LOADED.
C
      SUBROUTINE CHANGE
      COMMON INPUT(128), OUTPUT(128), MATRIX(128,128)
      INTEGER*2 INPUT, OUTPUT, MATRIX
      DO 10 J=1,128
10    READ 20, MATRIX(1,J)
20    FORMAT(Z2)
      M = 1
      DO 50 I=2,128
      M = M + 1
      N = M
      DO 60 J=1,128
      MATRIX(I,J) = MATRIX(1,N)
      N = N + 1
      IF(N .EQ. 129)N=1
```

```
60    CONTINUE
50    CONTINUE
      RETURN
      END
C
C
C                        SUBROUTINE RAND1 & RAND2
C
C
C     SUBROUTINES RAND1 AND RAND2 ARE BASED ON THE RANDOM NUMBER
C     GENERATOR DESIGNED BY SHAN S. KUO IN HIS BOOK TITLED
C     'COMPUTER APPLICATIONS OF NUMERICAL METHODS' ADDISON
C     WESLEY PUBLISHING COMPANY, 1972. THE GENERATORS RECEIVED A
C     SEED FROM THE CALLING SUBROUTINES ENCODE AND DECODE. THE
C     COMPUTATIONS PRODUCE A REAL NUMBER IN THE RANGE OF .0 TO
C     0.1. THIS NUMBER IS THEN CONVERTED BY THE CALLING
C     SUBROUTINES TO AN INTEGER IN THE RANGE OF 1 TO 64.
C     ADDITIONALLY, THE GENERATORS PRODUCE ANOTHER INTEGER, BASED
C     ON THE SEED, TO BE USED TO RESEED THE GENERATOR.
C
C     VARIABLE DEFINITIONS
C
C          IKEY          INTEGER USED TO SEED THE GENERATORS.
C                        THIS NUMBER SHOULD BE COMPOSED OF
C                        NINE DIGITS AND BE AN ODD INTEGER.
C
C          IREKEY        INTEGER TRANSFERRED TO THE CALLING
C                        PROGRAM. IT IS USED TO RESEED THE
C                        GENERATOR.
C
C          RRANDM        THIS IS THE RANDOM NUMBER GENERATED
C                        AND WHICH IS RETURNED TO THE CALLING
C                        SUBROUTINE.
C
C
      SUBROUTINE RAND1(IKEY, IREKEY, RRANDM)
      IREKEY = IKEY * 65539
      IF(IREKEY .GE. 0) GO TO 6
5     IREKEY = IREKEY + 2147483647 + 1
6     RRANDM = IREKEY
      RRANDM= RRANDM * .4656613E-9
      RETURN
      END
```

```
      SUBROUTINE RAND2(IKEY, IREKEY, RRANDM)
      IREKEY = IKEY * 65539
      IF(IREKEY .GE. 0) GO TO 6
5     IREKEY = IREKEY + 2147483647 + 1
6     RRANDM = IREKEY
      RRANDM = RRANDM * .4656613E-9
      RETURN
      END
C
C     END OF PROGRAM
C
```

APPENDIX 5

1. Plaintext to be enciphered:

STXTHIS PROGRAM WHICH WILL ENCODE DATA. BY USING THE SAME CIPHER KEY AND MATRIX ,THE ENCODED DATA CAN BE DECODED. ETXEOTACKCOMPUTERS A

2. The hexadecimal numbers for the above letters and ASCII code is as follows:

10 25 44 4C 1D 00 00 00 05 15 7C 3C 15 0C 6C 00 3D 44 4C 1C 44 00 3D
4C 64 64 00 2C 74 1C 7C 24 2C 00 24 0C 25 0C 72 00 14 40 00 2D 1D 4C
74 3C 00 25 44 2C 00 1D 0C 6C 2C 00 1C 4C 05 44 2C 15 00 5C 2C 4D 00
0C 74 24 00 6C 0C 25 15 4C 45 00 62 25 44 2C 00 2C 74 1C 7C 24 2C 24
00 24 0C 25 0C 00 1C 0C 74 00 14 2C 00 24 2C 1C 7C 24 2C 24 72 18 20
30 1C 7C 6C 05 2D 25 2C 15 1D 00 0C

3. The encipherment was conducted using a matrix constructed with the first row consisting of hexadecimal numbers starting from 00 to 7F. The keys, IIKEY1 and IIKEY2, were set to 425378614 and 354217853. The second command "111" was used for the control module. The enciphered data follows:

5A 66 71 15 47 2C 1F 25 7D 22 4F 3E 02 4E 62 3C 69 0B 6E 23 0B 2F 59
18 2F 04 1F 4D 66 21 23 62 48 3D 5A 31 5A 2C 74 74 3A 13 1E 3B 28 55
10 5A 39 75 64 5D 4F 40 1D 16 58 0E 67 0E 74 56 72 20 50 5D 64 5A 15
60 32 29 77 0A 16 1A 41 75 38 4B 1A 50 58 27 30 38 0D 10 11 55 1B 46
2B 38 41 34 0D 07 39 42 38 34 31 40 04 42 60 4D 11 37 1D 22 32 31 32
45 3C 44 79 53 53 4F 52 06 2D 2F 21

4. The ASCII code characters represented by the hexadecimal numbers above are as follows:

+1RSRxEsT_$ygDLEi,GCS1m41uESCETXu0sYlDCA4,HTW+SYN+ENN'2c7ENQZSTX+ETB
^L[yBSSbVTa{aNj.EOTLF[L+RFF&NAK~!b#CAN^BEL9#LFVTtACKBELQSTXDC2Z3h5BE
LCANFQpETB(BELFSYNBS@(FFYDC2vS$&SYN&XGHUS::y*`UuDC4

5. The data in paragraph 1 above was enciphered a second time using the same matrix configuration. However, the keys, IIKEY1 and IIKEY2, were 256983771 and 406835231. The command "111" was used for the control module. The enciphered data follows:

76 76 42 51 38 60 05 60 75 02 00 30 12 19 7D 60 51 4C 34 2B 66 36 39
38 3E 62 35 29 70 0E 77 2D 3A 71 7B 09 6E 08 70 07 74 35 09 03 16 43
6C 47 75 2B 3B 2B 69 06 7B 7F 2A 73 11 29 02 56 24 2D 07 52 26 69 50
1E 76 78 0D 74 73 3C 23 2E 3C 03 2C 34 34 2B 79 77 66 11 6C 09 0B 01
7E 3E 6C 1D 2A 6C 7C 6C 02 1F 39 21 18 0B 24 3A 65 79 29 40 62 25 2E
3B 0C 61 07 51 27 07 19 5B 17 51 5C 4C

A SOFTWARE CIPHER SYSTEM FOR PROVIDING SECURITY FOR COMPUTER DATA

by

JOHN CLEVE WALKER

B.S., University of Detroit, 1971

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1977

The purpose of this paper is to demonstrate that a software cipher system can be designed to provide adequate security for computer data.

This security system will accept plaintext source data and convert it into random character strings. Design of the system will be centered around the multisubstitution cipher principle, i.e., a letter "C" will be enciphered as a "B" the first time and as a "Z" the second time. Additionally, the security provided by this system will be enhanced by use of a random cipher key which sets the pattern for multisubstitution ciphering. Overall, the security provided by this system will be to such a degree, that the cost of trying to break the cipher system would deter any potential intruder. Modules of the security system will consist of a control module, an encipher module, a decipher module, a change module and a random number generator.

The control module will determine if the user desires to encipher data or decipher data. Upon determining which mode is desired, the appropriate cipher module and the random number generator module will assume control.

The random number generator will be set by the user by use of the random cipher key. Determination of the cipher sequence will depend on the random cipher key and the random numbers produced by the generator.

In conjunction with the random numbers and the plaintext characters, the encipher module will select a cipher character from the cipher matrix. These cipher characters are then stored in an array which will be read out upon completion of the enciphering.

To decipher, the decipher module will require use of the enciphered

text plus the same cipher key used to encipher the original data. The decipher module will use the cipher character, cipher matrix and the random number to produce the plaintext character. Upon completion of deciphering, an array of deciphered data will be read out to the user.

Included in this paper will be a prototype of the system plus documentation covering the testing of the security of the cipher system.