AUTOMATED CRYPTANALYSIS OF XOR PLAINTEXT STRINGS

E.Dawson and L.Nielsen

ADDRESS Information Security Research Centre, Faculty of Information Technology, Queensland University of Technology, GPO Box 2434, Brisbane, Queensland 4001, AUSTRALIA.

ABSTRACT: In this paper a fully-automated attack on two XORed plaintext strings is implemented using only knowledge of the statistical properties of the plaintext language.

KEYWORDS: Stream cipher, keystream, frequency distribution, automated cryptanalysis, statistics, Vernam cipher.

1 INTRODUCTION

Stream ciphers based on XORing binary plaintext with a binary string (keystream) formed by a pseudorandom number generator can provide a highly secure means of communication, where XOR represents the bitwise exclusive-or operation. This is especially the case when the keystream is formed by a random process. This is the well-known Vernam cipher [2]. The Vernam cipher is an example of a class of encryption algorithms commonly known as a one-time pad. As was proven by Shannon in the 1940's a one-time pad is unbreakable [10]. However, the Vernam cipher suffers from the problem that both the sender and receiver need to be provided with the keystream over a secure channel. In the past this presented problems, especially in the case where one of the participants does not have a secure channel available, for example a foreign embassy located in an unfriendly country. One method to overcome this problem is to re-use the same keystream for more than one message. One of the most infamous examples of a practical application of the use of a repeated one-time pad is that published in Peter Wright's autobiography [11], the VENONA codebreak. Wright states that *'The VENONA codebreak became possible because during the early years of the war the Russians ran short of cipher material. Such was the pressure on their communications system that they made duplicate sets of their one-time pads and issued them to different embassies in the West.'* As a direct result of the Russians' duplication of one-time pads, their system was eventually broken into, manually, by cross-matching of ciphertext passages encrypted with the same keystream.

It has long been known that re-using the same keystream may be subject to attack. A theoretical attack on two cryptograms formed by the same keystream was proposed by Frank Rubin in 1978 [8]. In this paper we present results on such attacks. The attacks are fully automated and based on knowledge of the plaintext statistics only. A description of the software that was designed is given along with some experimental results. The results of such attacks have not been recorded previously in the open literature.

On the other hand, automated methods (based on ciphertext alone) for attacking simple ciphers such as substitution and transposition ciphers have been recorded extensively. This includes attacks on cryptograms from written text [3] [4] [9] [1] [7] as well as attacks on analog speech scramblers [5] [6]. To a large extent these attacks used both the weakness of the cipher and the redundancy of the plaintext. The success of the attacks was based on the ability of being able to design efficient algorithms to use the redundancy of the plaintext. Similar methods are employed in this paper.

2 OVERVIEW OF THE AUTOMATED COMPUTER PROGRAM

The program itself consists of a number of routines executed in succession. The two files $ptxt_0$ and $ptxt_1$ are the program's attempts at reproducing the original P_0 and P_1 plaintext files from the combined file $P_0 \oplus P_1$. At the end of each routine the $ptxt_0$ and $ptxt_1$ files are amended.

Initially, a 600K section (600 000 characters) of The Bible was prepared as follows: all alphabetic characters are converted to uppercase; all other characters (excepting the <space> character) are deleted; no two consecutive <space> characters are allowed. This gave a plaintext message space of A,B,...,Z,□ where □ represents the <space> character.

From this 600K section of The Bible, the frequency distribution of the twenty-seven characters was prepared as were the distributions of the one hundred most frequent digraphs and trigraphs. Distributions of the one hundred most frequent 2-,3-,...,11-character words were prepared, where a word is defined as being a set of consecutive non-<space> characters between two <space> characters. Other distributions prepared were listings of *impossible* 2-tuples and 3-tuples, where *impossible* means that the particular tuple never occurred.

In the case where we suspect that two separate sections of the plaintext message space were encrypted with the same (repeated) keystream, we need to confirm that this has occurred. It is simply a matter of looking for a particular distinctive pattern. If the two resulting ciphertext streams are XORed together (thus eliminating the identical keystreams)

$$(P_0 \oplus K) \oplus (P_1 \oplus K) = P_0 \oplus P_1$$

where $\oplus$ represents the binary exclusive-or operation, a definite pattern appears due to the characteristics of the valid plaintext characters. If the keystreams are not identical then this pattern will not appear when the ciphertext streams are XORed together. In this case the pattern resulting from the plaintext is that the first bit in every eight-bit block (that is, the first bit of each character) is a zero. This is due to the fact that the only valid plaintext characters are A,B,...,Z,□, whose hexadecimal values are $40_x..5A_x$ and 20_x. The first bit in the hexadecimal values 4, 5 and 2 is always a zero, therefore any XOR combination of these values will always give a zero in the first bit position. Identification of the position of the repeated keystream section is complete.

From the 600K section of The Bible, two 12K subsections were randomly selected and XORed, giving

$$C(i) = P_0(i) \oplus P_1(i) \qquad i = 0, 1, ..., 11999$$

where $P_0(i)$ represents the ith bit of P_0 (the first 12K plaintext block), $P_1(i)$ represents the ith bit of P_1 (the second 12K plaintext block) and $C(i)$ represents the ith bit of the ciphertext.

3 OVERVIEW OF ROUTINES

The deciphering routines work on blocks of characters, referred to as tuples. The number of characters per tuple varies depending on the routine employed. During each routine tuples are selected and tested for a particular format, once again depending upon the routine employed. If the tuple is accepted, the routine attempts to decipher the tuple for both plaintexts. The routines will only decipher words which are included in the lists of the most frequent words. When a routine successfully deciphers a tuple to both plaintexts, it then writes the deciphered tuple to the $ptxt_0$ and $ptxt_1$ files. One challenge in designing the program was to determine into which file a decrypted character or tuple was to be placed. For example if a ciphertext character was decrypted to, say, a G in one plaintext and an E in the other plaintext, how did one determine whether the G went to $ptxt_0$ or $ptxt_1$? The answer was that initially there is no way of telling.

In the case of tuples, however, it is noted that consecutive tuples overlap by two characters. Therefore, at the completion of each decryption routine the tuple is *matched* to either $ptxt_0$ or $ptxt_1$, depending upon the matching of the initial/final two characters in the tuples with their associated two characters in the $ptxt_0$ and $ptxt_1$ files. As tuples are integrated into the $ptxt_0$ and $ptxt_1$ files, a record is kept of the positioning of these groups of consecutive characters (tuples). These groups must remain unbroken as subsequent decryptions are performed. In this manner, individual sequences grow in length and concatenate as the decryption process continues. The file in which this information is recorded is referred to as the *sequence* file.

Routine 1

This routine effectively decrypts all characters which result from XORing a □ character (20_x) and an A..Z character ($40_x..5A_x$). Since the □ character is the most frequently occurring character (approximately 20% of all characters in this character set are □), 40% of the characters in the combined file represent a □ in one plaintext file and an associated character (A..Z or □) in the same position in the other plaintext file. Routine 1, therefore, decrypts characters in the combined file in the range

$\{60_x, ..., 7A_x\}$ to □ in one plaintext file and its unique associated character in the same position in the other plaintext file. The □ is written to $ptxt_0$ (default) and the character is written to $ptxt_1$. Consecutive □ characters are illegal (due to the defined character set), so consecutive decrypts such as [□ □ □],[A B C] are written to $ptxt_0$ as [□ B □] and $ptxt_1$ as [A □ C]. After Routine 1, 32.7% of the 24000 characters in the combined file are decrypted correctly. Since the A..Z characters are decrypted uniquely, no errors are evident at this stage. All □ characters are now placed except where two □ characters concur in the two plaintext files.

By noting that a [□ c] tuple in $ptxt_0$ and its associated [d □] tuple in $ptxt_1$ (where c,d ϵ {A,...,Z}), indicate both the beginning of a word in each of the plaintexts and the ending of the previous word, the placements of a majority of the words are now identified. Extract 1 shows simultaneous extracts from the $ptxt0$ and $ptxt1$ files. The boxed characters are those that identify the placement of concurrent words. For example, the first tuple pair of [□B???□M][L□???U□] indicates either a four-letter word beginning with B coupled with a four-letter word ending with U, or a five-letter word beginning with B and ending with U and a three-letter word. Other tuple pairs identified by Routine 1 are [□F??□D][F□??E□], [□W???□D][D□???O□],...

Using the frequency distributions of the most common words, Routines 2 to 31 were designed to fill in the gaps between the spaces.

Routines 2..9

Tuple pairs beginning with [□ c] and ending with [d □] are selected. An example is

$$\begin{array}{cccccc} \square & W & ? & ? & \square & W \\ D & \square & ? & ? & E & \square \end{array}$$

where $? \,\epsilon\, \{A, ..., Z\}$.

By searching the most frequent words files, this set of routines finds words which satisfy the required criteria; that is, in this example, a three-letter word beginning with W and a three-letter word ending with E, where the XOR combination of the associated ? characters from each tuple in the pair has a value equivalent to the character in the same position in the combined file. In this example

$$\begin{array}{cccccc} \square & W & ? & ? & \square & W \\ D & \square & ? & ? & E & \square \end{array} \quad \text{becomes} \quad \begin{array}{cccccc} \square & W & \mathbf{A} & \mathbf{S} & \square & W \\ D & \square & \mathbf{T} & \mathbf{H} & E & \square \end{array}$$

Note here that if the words WAS and THE are not included in the list of the most frequent three-letter words, then the six-tuple would not have been decrypted. These routines, incorporating words of length two to eleven letters, account for a further 5.7% correctly decrypted characters, taking the total characters correctly decrypted to 38.4% of the 24000 characters, with two of the 24000 characters being incorrectly decrypted. This error margin is introduced because the routines *guess* a character that occurs concurrently (i.e. identical charactersd) in the two plaintext files; that is, their XOR sum in the combined file is 00_x.

Extract 2 shows simultaneous extracts from the *ptxt*0 and *ptxt*1 files. The boxed words are those that were decrypted using Routines 2..9. For example, in the *ptxt*0 extract the first word BOTH was decrypted simultaneously with the first word THOU from the *ptxt*1 extract. Other decrypted pairs are FOR/THE, WERE/UNTO, ...

Routines 10..21

This set of routines extends on the ideas in Routines 2..9 by decrypting tuples to three words instead of two words. Tuple pairs beginning with [□ c] and ending with [d □] are selected, with the added constraint that one of the central characters must be a □ with its associated A..Z character. An example is

$$\begin{array}{cccccccccc} \square & H & ? & ? & ? & R & ? & ? & \square & S \\ F & \square & ? & ? & ? & \square & ? & ? & R & \square \end{array}$$

where $? \epsilon \{A, ..., Z\}$. Using similar techniques to the previous routines

$$\begin{array}{cccccccccc} \square & H & ? & ? & ? & R & ? & ? & \square & S \\ F & \square & ? & ? & ? & \square & ? & ? & R & \square \end{array} \text{ becomes } \begin{array}{cccccccccc} \square & H & \mathbf{U} & \mathbf{N} & \mathbf{D} & R & \mathbf{E} & \mathbf{D} & \square & S \\ F & \square & \mathbf{T} & \mathbf{H} & \mathbf{E} & \square & \mathbf{A} & \mathbf{I} & R & \square \end{array}$$

Only words which are found in the lists of the most frequent words are decrypted. At this stage 40.7% of the 24000 characters in the combined file are now correctly decrypted, with no further errors.

Extract 3 shows simultaneous extracts from the *ptxt*0 and *ptxt*1 files. The boxed words are those that were decrypted using Routines 10..21. For example, in the *ptxt*0 extract the first words THEY-THAT were decrypted simultaneously with the first word BECAUSE from the *ptxt*1 extract. Other decrypted pairs are HIM-IN/MOTHER, OF-THE/PASS, and THE-ARK/BROUGHT.

Routines 22..31

This set of routines is similar to Routines 10..21 in that selected tuples are decrypted to four words instead of three words. Tuple pairs beginning with [□ c] and ending with [d □] are selected, with the added constraint that one of the central characters must be a • character, where • denotes identical characters in both plaintexts. This set of routines decrypts the central • to a □ character in both plaintext files. This, then, gives four words for the routine to decrypt using similar techniques to previous routines. In the example

$$\begin{array}{ccccccccc} \square & A & ? & ? & \bullet & ? & ? & \square & E \\ D & \square & ? & ? & \bullet & ? & ? & D & \square \end{array} \text{ becomes } \begin{array}{ccccccccc} \square & A & \mathbf{N} & \mathbf{D} & \square & \mathbf{O} & \mathbf{F} & \square & E \\ D & \square & \mathbf{H} & \mathbf{E} & \square & \mathbf{D} & \mathbf{I} & D & \square \end{array}$$

At this stage 41.4% of the 24000 characters in the combined file are now correctly decrypted, with the number of incorrectly deciphered characters rising to three.

Routines 32..34

After Routines 10..21 most of the spaces and a large number of common words are identified along with isolated letters in other words. It is observed from the plaintext statistics that words of shorter character length (four or less) occur much more often than words of longer character length (seven or greater). This set of routines exploits this observation by decrypting seven-tuples of the form [???•???] to [???□???], where • represents identical (unknown) characters in both plaintext files, and six-tuples of the forms [???•??] and [??•???] to [???□??] and [??□???] respectively. This set of routines is very effective, given the simplicity of the observation. A further 2.2% of the 24000 characters are correctly decrypted characters, taking the total to 43.6%, with 0.1% of the 24000 characters being incorrectly decrypted. The number of word positions has now increased, with passages of text being easily read.

Routines 35..45

The decryption process could continue in this manner (that is, by selecting tuples of a specific format and decrypting them using known plaintext words); however, a different approach is now taken. A word from the most frequent words list is selected. The program attempts to fit the word into the partially-decrypted text. For example, □THE□ could be $fitted$ into a five-tuple of the format [□T??□], [□?H?□],or [□??E□]. The words selected are in order of frequency of occurrence in the original plaintext. By taking the first eight words of each length (three to eleven) a further 4.2% of the 24000 characters are correctly decrypted, giving a total of 47.8%, with 0.8% incorrect decryptions of characters.

By varying this technique a wide range of results occurrs. The results previously stated occur when selection of the tuple insists on the first and last characters being □ characters and at least one of the central characters matching the appropriate letter in the word. For example, [□T??□] is decrypted to [□THE□]. If the criteria for the first and last characters is allowed to incorporate concurrent (unknown, but guessing to be □) characters, with no insistence that any of the central characters match the letters of the word (for example [•???•] is decrypted to [□THE□]) then the results increase to 62.7% correctly decrypted characters with 17.8% incorrect decryptions. Either result is acceptable, depending on what is to follow in the decryption process; that is, one may wish to minimise the error or one may choose to ignore the error factor. These results use the eight most common words of each length for decryption. This number of words used can also be varied, once again depending on how critical the error factor is to further decryption techniques. Extract 4 shows simultaneous extracts from the $ptxt0$ and $ptxt1$ files. The boxed characters are probable words which may, at this stage, belong in either file.

4 A CHANGE IN DIRECTION

The results presented in Section 3 were based on the assumption that the same keystream was used to form two different cryptograms. If an attacker discovers that the same keystream was used to form three or more intercepted cryptograms, there should be additional leakage of a substantial amount of information.

In order to investigate this hypothesis, the techniques described in Section 3 were applied to three plaintext strings P_0, P_1 and P_2 which were encrypted with the same keystream. By XORing the cryptograms we are provided with three files to separate, namely $P_0 \oplus P_1$, $P_1 \oplus P_2$ and $P_0 \oplus P_2$. The program in Section 3 is now used to decrypt these files. Table 1 gives the results of these three decryptions.

Combined Files	*Correct*	*Percentage*	*Incorrect*	*Percentage*
$P_0 \oplus P_1$	7522	62.7%	2137	17.8%
$P_1 \oplus P_2$	7377	61.5%	2107	17.6%
$P_2 \oplus P_0$	7515	62.6%	2152	17.9%

Table 1

The *Correct* column gives the number of correctly-decrypted pairs of characters (out of 12000 pairs) belonging to either of the plaintext files named in the *Combined Files* column. The *Incorrect* column gives the number of incorrectly-decrypted pairs of characters.

By combining the results of these three decryptions, correctly decrypted characters are sorted and placed in an attempt to explicitly reproduce the three original plaintext files. This second program is implemented in two stages. The first stage uses the resulting *ptxt* files from the first program to determine the placement of decrypted characters in their respective plaintext files. This is achieved by sorting known characters. For example, if the characters at position i $(0 \leq i < 12000)$ for the decryption of $P_0 \oplus P_1$ are A and B, for $P_1 \oplus P_2$ are B and C, and for $P_0 \oplus P_2$ are C and A, then the only possibility for character i in the *ptxt*0 file is A, for *ptxt*1 is B, and for *ptxt*2 is C. This technique explicitly places 62% of the characters in their correct plaintext files, with 5% incorrect decryptions.

The second stage selects tuples beginning and ending with a □ character and finds words from the lists of most frequent words to fill inbetween the □ characters. Table 2 gives the results after this second program is implemented. The *Correct* column gives the number of characters which are correctly decrypted and explicitly placed in their file of origin. The *Incorrect* column gives the number of characters incorrectly-decrypted.

Explicit Files	*Correct*	*Percentage*	*Incorrect*	*Percentage*
P_0	9027	75.2%	1471	12.3%
P_1	9157	76.3%	1372	11.4%
P_2	9053	75.4%	1415	11.8%

Table 2

5 SUMMARY

An automated program has been implemented which has decrypted 62% of the total characters in the two plaintext files. It was shown how this can be extended to produce a more accurate result of 76% of the total characters being correctly decrypted.

With three-quarters of the total number of characters now being correctly decrypted and placed in their original plaintext files, the intent of the plaintext message is easily understood. Extract 5 shows simultaneous passages from the three decrypted plaintext messages. Extract 6 is the original plaintext.

The ideas used in this automated attack are based solely on the statistics of the plaintext message space. The same ideas can be utilised in an attack on plaintext from any message space. When the decryption is based on the plaintext statistics alone a highly-structured plaintext message space will obviously give a better result than a uniform plaintext message space. The degree of accuracy of the plaintext statistics will also affect the degree of accuracy of the resulting decrypted files.

These results emphasise the importance of using a different keystream for a stream cipher for each cryptogram. As well these results demonstrate that, for a deterministic keystream generator, the period of the generator should be much larger than the length of any cryptogram.

6 AUTOMATED ATTACK VERSUS MANUAL ATTACK

The attack described in this paper gives a very satisfactory result in a minimal time frame. The automated processes which were coded were based on simple observations of the plaintext statistics. The analyst then exploits the ability of the computer to quickly process a large number of repetitive tasks which is, of course, an intrinsic feature of statistical data.

The automated program, then, provides a good foundation for further decryption techniques and refinements. However, a computer is primarily a high capacity processor and is much more suited to executing say, one task on one thousand different data items than executing one thousand different tasks on one data item. Consequently, as the analyst refines the program (by allowing for more contingencies) he is faced with larger quantities of more complex code, longer execution times and diminishing returns. A computer running artificial intelligence software may be worthwhile investigating, but its application to this task is outside the scope of this paper.

As an alternative to the completely automated attack, the analyst may consider a completely manual attack. The problems here are obvious, including the time factor, the enormity of the task and the logistics of the attack. Consider, though, that these concerns are basically capacity problems, and capacity problems are perfectly suited to solution by a computing machine.

In contrast to the computing processor, the manual attacker is much more suited to the refinements required than the capacity work. For example, a human will recognise the word PARTY as an acceptable word immediately, whereas a computer will have to consult its list of most frequent words and, if the word is not listed, execute other routines in order to determine if this is an acceptable word. Another important example is that a manual attacker can make an intuitive guess at a decryption before proceeding with a manual technique, whereas the computer is restricted to the selection of certain routines according to the occurrence of certain conditions. This intuitive guessing is intrinsic to the nature of the information system that gives rise to perception. The manual attacker then, has the decided advantage of perception over its computerised counterpart. The human self-organising information system contrasts with the externally organised information system of the computer.

Although the completely automated attack is sufficient in this particular instance, the attack which would give the optimum results (ie maximum correct decryptions with minimum incorrect decryptions within a reasonable time frame) would be one which involves the computer for the bulk of the statistical processing and the human for the refinements and perceptions. The optimum decryption would be determined by a finely-balanced combination of automation and manual interface.

REFERENCES

1. Anderson, Roland. Improving the Machine Recognition of Vowels in Simple Substitution Ciphers *Cryptologia*, Vol.10, No.1, 1986, pp 10-22.
2. Beker, H. and Piper, F. Cipher Systems: The Protection of Communications *Northwood Books*, London, 1982.
3. Carroll, J.M. and Martin, S. The Automated Cryptanalysis of Substitution Ciphers *Cryptologia*, Vol.10, 1986, pp 193-209.
4. Carroll, J.M. and Robbins, L.E. The Automated Cryptanalysis of Polyalphabetic Ciphers *Cryptologia*, Vol.11, 1987, pp 193-205.
5. Goldburg, B., Dawson, E. and Sridharan, S. Automated Cryptanalysis of Analog Speech Scramblers *Advances in Cryptology: Proceedings of EUROCRYPT 91*, edited by D.Davies, Springer-Verlag, 1991, pp 422-430.
6. Goldburg, B., Sridharan, S. and Dawson, E. Design and Cryptanalysis of Transform Based Analog Speech Scramblers, *IEEE Journal of Selected Areas of Communication*, Volume 11, Number 5, 1993, pp 735-744.
7. Matthews, Robert. An Empirical Method for Finding the Keylength of Periodic Ciphers *Cryptologia*, Vol.12, No.4, 1988, pp 220-224.
8. Rubin, Frank. Computer Methods for Decrypting Random Stream Ciphers *Cryptologia*, Vol.2, No.3, July 1978.
9. Schatz, Bruce R. Automated Analysis of Cryptograms *Cryptologia*, Vol.1, No.2, April 1977.
10. Shannon, C.E. Communication Theory of Secrecy Systems *Bell System Technical Journal*, Vol.28, October 1949, pp 656-715.
11. Wright, Peter. Spy Catcher *William Heinemann*, Australia, 1987.

Biographical Sketch

Ed Dawson obtained a BSc in 1969 from University of Washington, an MA in 1974 from University of Sydney, an MLitStu in 1981 and MSc in 1985 both from University of Queensland, and a PhD in 1991 from Queensland University of Technology. He is a member of the IEEE and the International Association for Cryptologic Research, and a fellow of the Insititute of Combinatorics and its Applications. Currently, he is an Associate Professor and Acting Director of the Information Security Research Centre (ISRC) at the Queensland University of Technology (QUT). His research and teaching interests lie in the areas of cryptology and error control coding.

Lauren Nielsen obtained a BSc in 1990 from the Queensland University of Technology. Currently, she is employed as a full-time computer systems officer in the ISRC at QUT and is undertaking part-time MSc studies in cryptology within the ISRC.

```
? · ? ? ? ? · ? ?  T ? ?  ? ?  ? ? ?  B ? ? ?  M ? ?  ? ? ? · ? ? · ?  ?
 ?  ?  ? ? ?  C ? ? ?  ? ? ? · ? ?  ? ?  W · ? ? · · · ?  F ? ? ?  ?
F ? ?  D ? ? ?  ? ?  ?  ?  ·  ? ?  W ? ? ?  D ? ? ? ?  ? ? ?  ?  ? ?
 ?  ?  ? ? · ?  A ? ?  A ? ? ?  C ? ? · ?  R ? ? ?  ? ? ?  ?  ? ? ?
A ? ?  T ? · ?  ? ? ?  S ? ·  ?  ?  · ·  H ? ?  ? ?  T ? ? · · ? ? · ?
? ?  ? ?  U ? ? ?  ? ?  ?  ? ? ? ?  ? ?  U ? ·  H ? ?  W ? ? ?  ?  ?
 T · ?  ? ? ? · · ? ? · · ? ? ? ?  C ? · ? · · ? ?  S ? ? ?  R ? ? ?  ? ?
? ? ?  ?  ? ·  ?  ?  ?  ? ? ? · · ? ? ?  ·  ?  ? ? ?  A ? ?  ? · ? ·
· · · · ? · ?  ? ? · ? ? ? ?  ?  ?  W ? ? ? · ? ?  U ?  T ?  ?  ? ?  G
? ? · · ? ? · · ? ? ? ? · ·  ?  · ?  T ?  ? ?  ?  O ? ? ? · ? ? ?  ?  ?
? ? · ? ? ?  T ? ?  W ? ? ? ?  A ? · ?  ? ?  ?  ?  ?  ? ?  ? ?  ? ? ?
 A ? ? ? · ? ?  T ? ?  D ? ? ?  A ? ?  T ? ?  ?  ? ? ? ? ?  · ?  H ? ?
?  ?  ? ?  ?  ?  ? ? ? ?  H ? ?  F ? ? ?  R ? ? ?  F ? ? ? · ? ? ? ? ·
· · ? ?  E ? · ?  ? ?  ? ? ?  A ? ?  T ? ?  ? ? ?  ? ?  ?  ? ? ? · ? ?
· ? ?  ?  ?  ?  ? ? ? · ? ?  ? ? · ? ? ? ? ?  ? ? ?  ?  ?  ?  ?  ? ?
? · ? ?  P ? ? ?  ?  A ·  ?  ?  ? ? ? ?  O ?  ?  ?  ? ? ?  ? ?  ?
T ? ?  ?  ? ? ? ?  ? ? ? ? · · ? ?  ?  F ? ? ?  ? ?  S ? ? ?  R ? ? ? ?
? · ? ?  ? ? ? · ? ? ? ? ?  ?  ?  ? ?  K ? ?  T ? ?  S ? ? ?  ? ?  ? ?
? · ? ? ?  O ?  T ? ?  ?  ? ? ? · ? ? ? ? · ? ? ·  M ? ? ?  ? · ?  W ? ·
 A ? ? ?  ?  ?  ?  ? ? ·  ? ? ?  ? ?  D ? ? · ·  ? ? ?  C ? · ?  ? ?
? ?  ? · ? ? ? ?  G ? ? ?  ?  ?  ?  ?  ? ? ?  I ?  ?  ?  ? ?  ? ?
M ? ? ?  A ? ?  ?  ?  ? ?  ? ?  D ? ?  ? ?  ?  ?  ? ?  ? ?  W ? ? ·
 T ? ?  T ?  ?  ? ?  ?  ?  M ? ? ? ? ?  ? ?  S ? ? ?  A ? ?  I ? · ?
? ? ?  ?  O ?  ? ?  A ?  ? ? ?  E ? ?  O ?  F ? · ? ?  ? ?  ?  T ? ?
?  N ? ?  C ? ? ·  ? ? · ? ? ?  W ? ? ? ?  C ? ?  T ? ?  ? ? ?  W ? ?
```

```
? · ? ? ? ? · ? ? T  ? ? E ? ? O ? ? ? L  ? ? ? U  ? ? T ? ? ? · ? ? · ? L ?
H ? N ? R ? ? ? D  ? ? ? P ? ? ? · ? ? I ? ? S  · ? ? · · · ? U  ? ? ? E ? F
 ? ? E  ? ? ? V ? ? O ? N ? R · H ? ? D  ? ? ? O  ? ? ? ? O ? ? ? L ? R ? ?
O ? H ? E ? ? · ? H  ? ? D  ? ? ? H  ? ? · ? D  ? ? ? I ? ? ? E ? L ? ? ? E
 ? ? E  ? · ? A ? ? ? T  ? · R ? A ? I · · E  ? ? T ? ? R  ? ? · · ? ? · ?
? ? I ? ? E  ? ? ? E ? ? A ? R ? ? ? ? L ? ? D  ? · N  ? ? E  ? ? ? T ? I ?
N  · ? N ? ? ? · · ? ? · · ? ? ? ? Y  ? · ? · · ? ? D  ? ? ? S  ? ? ? M ? ?
? ? ? E ? O ? · E ? N ? N ? V ? ? ? · · ? ? ? N · O ? H ? ? ? D  ? ? H ? · ? ·
· · · · ? · ? T ? ? · ? ? ? ? M ? A ? S  ? ? ? · ? ? M  ? N  ? H ? N ? ? K
? ? · · ? ? · · ? ? ? ? · · N ? I · ? W  ? E ? ? S ? E  ? ? ? · ? ? ? T ? A ?
? ? · ? ? ? D  ? ? D  ? ? ? ? S  ? · ? W ? ? E ? H ? H ? R ? ? U ? ? A ? ? ?
E  ? ? ? · ? ? T  ? ? D  ? ? ? E  ? ? R  ? ? E ? I ? ? ? ? ? ? F · ? E  ? ?
? E ? O ? ? R ? O ? T ? ? ? ? ? D  ? ? D  ? ? ? H  ? ? ? M  ? ? ? · ? ? ? ? ·
· · ? ? S  ? · ? T ? ? I ? ? ? ? L  ? ? E  ? ? O ? ? ? E ? ? O ? E ? ? ? · ? ?
· ? ? O ? A ? F ? A ? ? ? · ? ? R ? ? · ? ? ? ? ? ? N ? ? ? L ? H ? N ? A ? F ? ?
? · ? ? E  ? ? ? E ? F  · H ? H ? U ? ? ? ? ? D  ? N ? H ? I ? ? ? D ? ? Y ? F
 ? ? E ? A ? ? ? ? U ? ? ? ? · · ? ? T ? D  ? ? ? I ? ? E  ? ? ? D  ? ? ? ?
? · ? ? R ? ? ? · ? ? ? ? ? ? T ? A ? O ? ? H  ? ? P  ? ? E  ? ? ? N ? ? E ? ?
? · ? ? ? E  ? F  ? ? F ? O ? ? ? · ? ? ? ? · ? ? · W  ? ? ? T ? · ? S  ? ·
E  ? ? ? A ? H ? N ? O ? ? · I ? ? ? E ? ? D  ? ? · · A ? ? ? N  ? · ? I ? ?
? ? L ? · ? ? ? ? L  ? ? ? E ? E ? T ? A ? O ? ? ? M  ? H ? H ? O ? ? N ? ? D
 ? ? ? H  ? ? I ? H ? A ? ? R ? ? S  ? ? O ? ? E ? H ? B ? ? N ? ? D  ? ? ·
L  ? ? S  ? P ? E ? ? E ? H ? F  ? ? ? ? ? ? I ? ? T  ? ? ? N  ? ? S  ? · ?
? ? ? E ? O  ? A ? ? E  ? R ? ? ? D  ? ? D  ? N  ? · ? ? E ? ? Y ? F  ? ?
? E  ? ? H  ? ? · N ? ? · ? ? ? S  ? ? ? ? W  ? ? N  ? ? O ? ? ? T  ? ? C
```

Extract 1

```
? · ? ? ? ? · ? ?  T ? ?  ? ?  ? ? ?  BOTH  M ? ?  ? ? ? · ? ? · ?  ?
 ?  ?  ? ? ?  C ? ? ?  ? ? ? · ? ?  ? ?  W · ? ? · · · ?  F ? ? ?  ?
FOR  D ? ? ?  ? ?  ?  ?  ·  ? ?  WERE  D ? ? ? ?  ? ? ?  ?  ? ?
 ?  ?  ? ? · ?  AND  A ? ? ?  CALLED  ? ? ?  ? ? ?  ?  ? ? ?
A ? ?  T ? · ?  ? ? ?  S ? ·  ?  ?  · ·  H ? ?  ? ?  T ? ? · · ? ? · ?
? ?  ? ?  U ? ? ?  ? ?  ?  ? ? ? ?  ? ?  UPON  ? ?  W ? ? ?  ?  ?
 T · ?  ? ? ? · · ? ? · · ? ? ? ?  C ? · ? · · ? ?  S ? ? ?  R ? ? ?  ? ?
? ? ?  ?  ? ·  ?  ?  ?  ? ? ? · · ? ? ?  ·  ?  ? ? ?  A ? ?  ? · ? ·
· · · · ? · ?  ? ? · ? ? ? ?  ?  ?  W ? ? ? · ? ?  US  T ?  ?  ? ?  G
? ? · · ? ? · · ? ? ? ? · ·  ?  · ?  T ?  ? ?  ?  O ? ? ? · ? ? ?  ?  ?
? ? · ? ? ?  THE  WATERS  ? · ?  ? ?  ?  ?  ?  ? ?  ? ?  ? ? ?
 A ? ? ? · ? ?  THE  D ? ? ?  AND  T ? ?  ?  ? ? ? ? ?  · ?  H ? ?
?  ?  ? ?  ?  ?  ? ? ? ?  HIM  FORTH  ? ? ?  F ? ? ? · ? ? ? ? ·
· · ? ?  E ? · ?  ? ?  ? ? ?  AND  T ? ?  ? ? ?  ? ?  ?  ? ? ? · ? ?
· ? ?  ?  ?  ?  ? ? ? · ? ?  ? ? · ? ? ? ? ?  ? ? ?  ?  ?  ?  ?  ? ?
? · ? ?  P ? ? ?  ?  A ·  ?  ?  ? ? ? ?  O ?  ?  ?  ? ? ?  ? ?  ?
T ? ?  ?  ? ? ? ?  ? ? ? ? · · ? ?  ?  F ? ? ?  ? ?  S ? ? ?  R ? ? ? ?
? · ? ?  ? ? ? · ? ? ? ? ?  ?  ?  ? ?  KEEP  ? ?  S ? ? ?  ? ?  ? ?
? · ? ? ?  OF  T ? ?  ?  ? ? ? · ? ? ? ? · ? ? ·  M ? ? ?  ? · ?  WIF
E  ? ? ?  ?  ?  ?  ? ? ·  ? ? ?  ? ?  D ? ? · ·  ? ? ?  C ? · ?  ? ?
? ?  ? · ? ? ? ?  G ? ? ?  ?  ?  ?  ?  ? ? ?  I ?  ?  ?  ? ?  ? ?
MONTH  ? ?  ?  ?  ? ?  ? ?  D ? ?  ? ?  ?  ?  ? ?  ? ?  W ? ? ·
 THE  T ?  ?  ? ?  ?  ?  M ? ? ? ? ?  ? ?  S ? ? ?  AND  I ? · ?
? ? ?  ?  O ?  ? ?  A ?  ? ? ?  END  OF  F ? · ? ?  ? ?  ?  THA
T  N ? ?  C ? ? ·  ? ? · ? ? ?  W ? ? ? ?  C ? ?  T ? ?  ? ? ?  W ? ?
```

```
? · ? ? ? ? · ? ? T  ? ? E ? ? O ? ? ? L  THOU  ? ? T ? ? ? · ? ? · ? L ?
H ? N ? R ? ? ? D  ? ? ? P ? ? ? · ? ? I ? ? S  · ? ? · · · ? U  ? ? ? E ? F
 THE  ? ? ? V ? ? O ? N ? R · H ? ? D  UNTO  ? ? ? ? O ? ? ? L ? R ? ?
O ? H ? E ? ? · ? H  AND  ? ? ? H  ONLY  R ? ? ? I ? ? ? E ? L ? ? ? E
 ? ? E  ? · ? A ? ? ? T  ? · R ? A ? I · · E  ? ? T ? ? R  ? ? · · ? ? · ?
? ? I ? ? E  ? ? ? E ? ? A ? R ? ? ? ? L ? ? D  TO  H ? ? E  ? ? ? T ? I ?
N  · ? N ? ? ? · · ? ? · · ? ? ? ? Y  ? · ? · · ? ? D  ? ? ? S  ? ? ? M ? ?
? ? ? E ? O ? · E ? N ? N ? V ? ? ? · · ? ? ? N · O ? H ? ? ? D  ? ? H ? · ? ·
· · · · ? · ? T ? ? · ? ? ? ? M ? A ? S  ? ? ? · ? ? M  IN  ? H ? N ? ? K
? ? · · ? ? · · ? ? ? ? · · N ? I · ? W  ? E ? ? S ? E  ? ? ? · ? ? ? T ? A ?
? ? · ? ? ? D  AND  TAKE  A ? · ? W ? ? E ? H ? H ? R ? ? U ? ? A ? ? ?
E  ? ? ? · ? ? T  AND  ? ? ? E  FOR  ? ? E ? I ? ? ? ? ? ? F · ? E  ? ?
? E ? O ? ? R ? O ? T ? ? ? ? D  AND  THE  R ? ? ? M  ? ? ? · ? ? ? ? ·
· · ? ? S  ? · ? T ? ? I ? ? ? L  THE  ? ? O ? ? ? E ? ? O ? E ? ? ? · ? ?
· ? ? O ? A ? F ? A ? ? ? · ? ? R ? ? · ? ? ? ? ? N ? ? ? L ? H ? N ? A ? F ? ?
? · ? ? E  ? ? ? E ? F  · H ? H ? U ? ? ? ? D  ? N ? H ? I ? ? ? D ? ? Y ? F
 ? ? E ? A ? ? ? ? U ? ? ? ? · · ? ? T ? D  ? ? ? I ? ? E  ? ? ? D  ? ? ? ?
? · ? ? R ? ? ? · ? ? ? ? ? ? T ? A ? O ? ? H  ON  T ? ? E  ? ? ? N ? ? E ? ?
? · ? ? ? E  OF  ? ? F ? O ? ? ? · ? ? ? ? ? · ? ? · W  ? ? ? T ? · ? S  OF
 A ? ? ? A ? H ? N ? O ? ? · I ? ? ? E ? ? D  ? ? · · A ? ? ? N  ? · ? I ? ?
? ? L ? · ? ? ? ? L  ? ? ? E ? E ? T ? A ? O ? ? ? M  ? H ? H ? O ? ? N ? ? D
 SHE  A ? ? I ? H ? A ? ? R ? ? S  ? ? O ? ? E ? H ? B ? ? N ? ? D  ? ? ·
L  WAS  ? P ? E ? ? E ? H ? F  ? ? ? ? ? ? I ? ? T  ? ? ? N  WAS  ? · ?
? ? ? E ? O  ? A ? ? E  ? R ? ? ? D  AND  IN  ? · ? ? E ? ? Y ? F  TI
ME  ? ? H  ? ? · N ? ? · ? ? ? S  ? ? ? ? W  ? ? N  ? ? O ? ? ? T  ? ? C
```

Extract 2

```
?·????·?? T?? ?? ??? BOTH M?? ???·??·? ?
 ? ? ??? C??? ???·?? ?? W·??···? F??? ?
FOR D??? ?? ? ? · ?? WERE D???? ??? ? ??
 ? ? ??·? AND A??? CALLED ??? ??? ? ???
A?? THEY THAT ?· ? ? ·· HIM IN T??··??·?
?? ?? U??? ?? ? ???? ?? UPON ?? W??? ? ?
 T·? ???··??··???? C?·?··?? S??? R??? ??
??? ? ?· ? ? ? ???··??? · ? ??? A?? ?·?·
····?·? ??·???? ? ? W???·?? US T? ? ?? G
??··??··????·· ? ·? T? ?? ? O???·??? ? ?
??·??? THE WATERS ?·? ?? ? ? ? ?? ?? ???
 A???·?? THE D??? AND T?? ? ????? ·? H??
? ? ?? ? ? ???? HIM FORTH ??? F???·????·
··?? E?·? ?? ??? AND T?? ??? ?? ? ???·??
·?? ? ? ? ???·?? ??·????? ??? ? ? ? ? ??
?·?? P??? ? A· ? ? ???? O? ? ? ??? ?? ?
T?? ? ???? ????··?? ? F??? ?? S??? R????
?·?? ???·????? ? ? ?? KEEP ?? S??? ?? ??
?·??? OF T?? ? ???·????·??· M??? ?·? WIF
E ??? ? ? ? ??· ??? ?? D??·· ??? C?·? ??
?? ?·???? G??? ? ? ? ? ??? I? ? ? ?? ??
MONTH ?? ? ? ?? ?? D?? ?? ? ? ?? ?? W??·
 THE T? ? ?? ? ? M????? ?? S??? AND I?·?
??? ? OF THE ? ??? END OF F?·?? ?? ? THA
T N?? C??· ??·??? W???? C?? THE ARK W??
```

```
?·????·??T ??E??O???L THOU ??T???·??·?L?
H?N?R???D ???P???·??I??S ·??···?U ???E?F
 THE ???V??O?N?R·H??D UNTO ????O???L?R??
O?H?E??·?H AND ???H ONLY R???I???E?L???E
 ??E BECAUSE S?·R?A?I··E MOTHER ??··??·?
??I??E ???E??A?R????L??D TO H??E ???T?I?
N ·?N???··??··????Y ?·?··??D ???S ???M??
???E?O?·E?N?N?V???··???N·O?H???D ??H?·?·
····?·?T??·????M?A?S ???·??M IN ?H?N??K
??··??··????··N?I·?W ?E??S?E ???·???T?A?
??·???D AND TAKE A?·?W??E?H?H?R??U??A???
E ???·??T AND ???E FOR ??E?I??????F·?E ??
?E?O??R?O?T?????D AND THE R???M ???·????·
··??S ?·?T??I???L THE ??O???E??O?E???·??
·??O?A?F?A???·??R??·??????N???L?H?N?A?F??
?·??E ???E?F ·H?H?U?????D ?N?H?I???D??Y?F
 ??E?A????U????··??T?D ???I??E ???D ????
?·??R???·??????T?A?O??H ON T??E ???N??E??
?·???E OF ??F?O???·????·??·W ???T?·?S OF
 A???A?H?N?O??·I???E??D ??··A???N ?·?I??
??L?·????L ???E?E?T?A?O???M ?H?H?O??N??D
 SHE A??I?H?A??R??S ??O???E?H?B??N??D ??·
L WAS ?P?E??E?H?F ??????I??T ???N WAS ?·?
???E?O PASS A?R???D AND IN ?·??E???Y?F TI
ME ??H ??·N??·???S ????W ??N BROUGHT ??C
```

Extract 3

? ???? ?? THE ?? ??? BOTH M?? ??? ??·? ?
? ? ??? C??? ??? OF IT W·??···? F??? ?
FOR D??? ?? ? ? · ?? WERE D???? ??? ? ??
? ? ??·? AND A??? CALLED ??? ??? ? ???
AND THEY THAT ?· ? ? ·· HIM IN T??··?? ?
?? ?? UNTO ?? ? ???? ?? UPON ?? W??? ? ?
THE ???··??··???? C?·?··?? SAID R??? ??
??? ? ?· ? ? ? ???··??? · ? ??? AND ?·?·
····?·? ?? ???? ? ? WITH OF US T? ? ?? G
??··??··????·· ? ·? T? ?? ? O??? ??? ? ?
?? ZNC THE WATERS ?·? ?? ? ? ? ?? ?? ???
ALSO ER THE D??? AND THE ? ????? ·? H??
? ? ?? ? ? ???? HIM FORTH ??? F??? ????·
··?? E?·? ?? ??? AND THE ??? ?? ? ??? ??
·?? ? ? ? ??? SO HE ????? ??? ? ? ? ? ??
? HE P??? ? A· ? ? ???? O? ? ? ??? ?? ?
THE ? ???? ????··?? ? F??? ?? S??? R????
? ?? ??? ????? ? ? ?? KEEP ?? S??? ?? ??
? ??? OF THE ? ??? ???? ??· M??? ?·? WIF
E ??? ? ? ? ??· ??? ?? D??·· ??? C?·? ??
?? ?·???? G??? ? ? ? ? ??? I? ? ? ?? ??
MONTH ?? ? ? ?? ?? D?? ?? ? ? ?? ?? W??·
THE T? ? ?? ? ? M????? ?? S??? AND I?·?
??? ? OF THE ? ??? END OF F?·?? ?? ? THA
T NOT C??· ?? ??? W???? C?? THE ARK W??

? ???? ??T BRE??O???L THOU ??T??? ??·?L?
H?N?R???D ???P??? THINGS ·??···?U ???E?F
THE ???V??O?N?R·H??D UNTO ????O???L?R??
O?H?E??·?H AND ???H ONLY R???I???E?L???E
EVE BECAUSE S?·R?A?I··E MOTHER ??··?? ?
??I??E WATE??A?R????L??D TO H??E ???T?I?
N HUN???··??··????Y ?·?··??D MONS ???M??
???E?O?·E?N?N?V???··???N·O?H???D BEH?·?·
····?·?T?? ????M?A?S ONE HIM IN ?H?N??K
??··??··????··N?I·?W ?E??S?E ??? ???T?A?
?? SAID AND TAKE A?·?W??E?H?H?R??U??A???
E AND OUT AND ???E FOR EVE?I?????F·?E ??
?E?O??R?O?T????D AND THE R???M ??? ????·
··??S ?·?T??I???L THE GRO???E??O?E??? ??
·??O?A?F?A??? EARTH ?????N???L?H?N?A?F??
? THE ???E?F ·H?H?U????D ?N?H?I???D??Y?F
EDE?A????U????··??T?D ???I??E ???D ????
? ??R??? ?????T?A?O??H ON T??E ???N??E??
? ???E OF LIF?O??? ???? ??·W ???T?·?S OF
A???A?H?N?O??·I???E??D ??··A???N ?·?I??
??L?·????L ???E?E?T?A?O???M ?H?H?O??N??D
SHE A??I?H?A??R??S ??O??E?H?B??N??D ??·
L WAS ?P?E??E?H?F ?????I??T ???N WAS ?·?
???E?O PASS A?R???D AND IN ?·??E??Y?F TI
ME IAH ??·N?? ???S ????W ??N BROUGHT ??C

Extract 4

T THEE EAT BREAD WILL THOU R?TU?D OF ???
?? GROUND FOR ??? OF IN WISE TOOK ?YKEF
FOR D?S? THOU ARE AND UNTO THOU SHAL?R??
O? FATHER AND AE?M CALLED HIS ??FE? N??E
??VE BECAUSE SK? WAS TO ?M?THER OF IN ?
?VINN UNTO THAT ALSO PND TO HIS WHEN DID
THE IOR? GATHERED C??T? ?? SDINS ??? BL
ESSED THEM AND THE LORD FOR SAID ?EHO??F
??? ??? IS BECOME AS ONE OF US TO ?N?C G
I? A????V????ND NOW LEFT HE PUS FORTH N
UR SAID AND ?ALE ALSO OF THE TR?? OF LIF
E AND EAT AND ?I?E??OR OVER THEREFORE ??
? LORD FOR ???? HIM FORTH FROM ??? GARDE
N OF EVEN TO TI?L ?? GROUND FROM PN??C?
HE ??? TAKEN OH HE DROV? HOT THE WAS AN
D OF PHACEE AS THE ALSO OF THE GARDEN OF

E ?A?U OF THE GROUND BOTH MAN A?N AI SER
VANT THE CREEPING THINGS AND ?OE?F?OL GF
THE HEAVEN AND THEY WERE D???SOYED ? ??
WHEREWITH AND O?AH ONLY RECEIVED ?LI??
AND THEY THAT T?R? ?ITOE?I? IN THE WI A
ND TAE WATERS ?R??A?LTD UPON THE DEATH I
N ?UNARE? A???????Y DAYS AND HOD REMEMBE
RED LORD AND EVERY LIVING THINE AND AND
??? CATTLE THAT WAS WITH HIM IN THE ?FK
GO GENERATION WITH TO PASS OVEU THE ?AT
HI AND THE W?TBRS ???WAGE? THE MOUNTAINS
ALSO OF THE DIED AND THE WIND?WS OF HEA
VEN WERE ?T????D AND THE RAI? FROM HEAVE
N WAS RESTRAINED A??ETHE WATERS ?ESS??E?
FRO???F? THE ?FRT? CONT?NR?LL? AND AFTE
R ?KE AND NF THE HUNDRED AND FIFTY DAYS

HEM ?BO? UR OF THE ?HALDEES TO HE OF ???
?? LAND OF CANAAN AND THEM LAMB KNOW HIR
AN AND DWELT THERE AND THE YOUR OF TO ??
WERE ??? HUNDRD? AND HAVE SHALL AND EAR
TH DIED IN ?ARB? FOR OF ?L?RD HAD S LF U
NTO HBR?? GOD THAT ?UE OF THY ?OUO??Y AN
D ?ROH THY ??????? AND FROM TGY FATHERS
THESE UNTO ? LAND THAT I?W?L? THOU THAT
??? ???I?L MAKE OF THEE A GREAT NA?I?Z A
H?DI ???? YEARS THEE AND MAKE TOY NAME A
NDA? AND THOU THAL???E A BLESSING AND ?
WILL BLESS THEM THAT BLESS THEM AND CU??
? HIS THAT KNOWETH THEM AND IN THEE SHAL
L ALL FAMILIES OF ?? EARTH BE B?ETU?? T
O ABRAM ?EP?RT?C AS THE YOUR HAD SPOKEN
?N?L HMM AOD FOR WERE WITH HIS AND ABRAM

Extract 5

T THOU EAT BREAD TILL THOU RETURN UNTO T
HE GROUND FOR OUT OF IT WAST THOU TAKEN
FOR DUST THOU ART AND UNTO DUST SHALT TH
OU RETURN AND ADAM CALLED HIS WIFES NAME
EVE BECAUSE SHE WAS THE MOTHER OF ALL L
IVING UNTO ADAM ALSO AND TO HIS WIFE DID
THE LORD GOD MAKE COATS OF SKINS AND CL
OTHED THEM AND THE LORD GOD SAID BEHOLD
THE MAN IS BECOME AS ONE OF US TO KNOW G
OOD AND EVIL AND NOW LEST HE PUT FORTH H
IS HAND AND TAKE ALSO OF THE TREE OF LIF
E AND EAT AND LIVE FOR EVER THEREFORE TH
E LORD GOD SENT HIM FORTH FROM THE GARDE
N OF EDEN TO TILL THE GROUND FROM WHENCE
HE WAS TAKEN SO HE DROVE OUT THE MAN AN
D HE PLACED AT THE EAST OF THE GARDEN OF

E FACE OF THE GROUND BOTH MAN AND CATTLE
AND THE CREEPING THINGS AND THE FOWL OF
THE HEAVEN AND THEY WERE DESTROYED FROM
THE EARTH AND NOAH ONLY REMAINED ALIVE
AND THEY THAT WERE WITH HIM IN THE ARK A
ND THE WATERS PREVAILED UPON THE EARTH A
N HUNDRED AND FIFTY DAYS AND GOD REMEMBE
RED NOAH AND EVERY LIVING THING AND ALL
THE CATTLE THAT WAS WITH HIM IN THE ARK
AND GOD MADE A WIND TO PASS OVER THE EAR
TH AND THE WATERS ASSWAGED THE FOUNTAINS
ALSO OF THE DEEP AND THE WINDOWS OF HEA
VEN WERE STOPPED AND THE RAIN FROM HEAVE
N WAS RESTRAINED AND THE WATERS RETURNED
FROM OFF THE EARTH CONTINUALLY AND AFTE
R THE END OF THE HUNDRED AND FIFTY DAYS

HEM FROM UR OF THE CHALDEES TO GO INTO T
HE LAND OF CANAAN AND THEY CAME UNTO HAR
AN AND DWELT THERE AND THE DAYS OF TERAH
WERE TWO HUNDRED AND FIVE YEARS AND TER
AH DIED IN HARAN NOW THE LORD HAD SAID U
NTO ABRAM GET THEE OUT OF THY COUNTRY AN
D FROM THY KINDRED AND FROM THY FATHERS
HOUSE UNTO A LAND THAT I WILL SHOW THEE
AND I WILL MAKE OF THEE A GREAT NATION A
ND I WILL BLESS THEE AND MAKE THY NAME G
REAT AND THOU SHALT BE A BLESSING AND I
WILL BLESS THEM THAT BLESS THEE AND CURS
E HIM THAT CURSETH THEE AND IN THEE SHAL
L ALL FAMILIES OF THE EARTH BE BLESSED S
O ABRAM DEPARTED AS THE LORD HAD SPOKEN
UNTO HIM AND LOT WENT WITH HIM AND ABRAM

Extract 6

References

[1] Roland Anderson.*Improving the Machine Recognition of Vowels in Simple Substitution Ciphers* Cryptologia, Vol.10, No.1, 1986, pp 10-22.

[2] H.Beker and F.Piper.*Cipher Systems: The Protection of Communications* Northwood Books, London, 1982.

[3] J.M.Carroll and S.Martin.*The Automated Cryptanalysis of Substitution Ciphers* Cryptologia, Vol.10, 1986, pp 193-209.

[4] J.M.Carroll and L.E.Robbins.*The Automated Cryptanalysis of Polyalphabetic Ciphers* Cryptologia, Vol.11, 1987, pp 193-205.

[5] B.Goldburg,E.Dawson and S.Sridharan.*Automated Cryptanalysis of Analog Speech Scramblers* Advances in Cryptology: Proceedings of EUROCRYPT 91, edited by D.Davies, Springer-Verlag, 1991, pp 422-430.

[6] B.Goldburg,S.Sridharan and E.Dawson.*Design and Cryptanalysis of Transform Based Analog Speech Scramblers* IEEE Journal of Selected Areas of Communication, Volume 11, Number 5, 1993, pp 735-744.

[7] Robert Matthews.*An Empirical Method for Finding the Keylength of Periodic Ciphers* Cryptologia, Vol.12, No.4, 1988, pp 220-224.

[8] Frank Rubin.*Computer Methods for Decrypting Random Stream Ciphers* Cryptologia, Vol.2, No.3, July 1978.

[9] Bruce R.Schatz.*Automated Analysis of Cryptograms* Cryptologia, Vol.1, No.2, April 1977.

[10] C.E.Shannon.*Communication Theory of Secrecy Systems* Bell System Technical Journal, Vol.28, October 1949, pp 656-715.

[11] Peter Wright.*Spy Catcher*William Heinemann, Australia, 1987.