ers',
(ETH),

SPEECH SECURITY AND PERMANENTS OF (0,1) MATRICES

C. Mitchell
(Hewlett-Packard Laboratories, Bristol)

1. INTRODUCTION

Voice communications still rely to a considerable extent on narrow band channels, such as telephone or radio links. The ease with which conversations over such links can be monitored has led to a continuing wide requirement for means to protect the secrecy of such conversations.

Given that the channel has a narrow bandwidth, the measures that can be taken to protect the secrecy of the voice signal are somewhat limited. This is because, even with expensive and sophisticated modems, bit rates over such channels are normally limited to 2000-3000 bits/second. Thus, if the voice signal is to be encrypted digitally, special low bit rate voice coders must be used. These coders are not only relatively sophisticated and hence relatively costly, but are also highly sensitive to errors.

So if the requirement is for a security system of modest cost and high reliability over poor channels then analogue "scrambling" techniques must be used. This requirement remains an extremely common one, and so analogue voice scramblers remain an important part of the market for security and privacy equipment.

Most analogue speech security systems operate by scrambling the voice signal in either the time domain or the frequency domain, and some operate in both domains. A variety of techniques can be employed to scramble the signal, but we are concerned here with one particular technique for scrambling in the time domain. This technique, known as time element scrambling (TES), is an extremely popular method and can be readily combined with scrambling effects in the frequency domain.

The time element scrambling technique is one of the most effective and easily implemented speech scrambling techniques; speech is divided into segments in the time domain, and then these segments are re-ordered prior to transmission; see, for example, [2]. Properly used, this technique can render the transmitted signal extremely difficult to decipher, and, as a result, this type of technique is widely used in commercially available equipment. However, this technique necessarily introduces a delay into the communications path, which must be minimised for the sake of user convenience.

Thus strategies must be devised for re-ordering the speech segments which minimise the time delay, whilst at the same time maximising the diversity of patterns available in order to maximise the security level. In order to achieve this, a number of different strategies have been produced, and a description and comparison of some of the various different rearrangement methods can be found in [5].

For some of these strategies, the problem of assessing the diversity of rearrangement patterns available reduces to a permutation enumeration problem. For many such systems the enumeration of the possible usable permutations remains an intractable problem.

The purpose of this paper is to consider two such strategies for which the permutation enumeration problem is equivalent to evaluating the permanent of certain (0,1) matrices. Computing the permanent of a (0,1) matrix is known to be a hard problem in the general case, [3], [4]. This paper will describe new work of the author, which, in conjunction with other recent joint work of Beker and the author, [1], means that it is now possible to compute the permanent for a larger number of the relevant matrices than was previously possible.

The new results on permanent evaluation take the form of proving that the permanent of certain (0,1) matrices of dimension n by n is equal to the sum of certain entries in the nth power of another matrix of size independent of n. These results will form the central part of this paper.

Although time element scramblers are still of considerable commercial importance, they do represent just one type of speech security device. For a general introduction to speech security techniques, the interested reader is referred to the recent book of Beker and Piper, [2], which appears to be the only book dedicated to this subject.

2. TWO STRATEGIES FOR TIME ELEMENT SCRAMBLING

2.1 The two techniques

The two techniques described here both represent compromises between pattern diversity and minimal time delay, but have different targetted security levels. The first strategy we consider is that described in [5] as Overlapping Frame Sliding Window Scrambling, which has a somewhat limited security level. The second strategy however, called Disjoint Frame Sliding Window Scrambling in [5], has a level of security approaching the maximum possible from a time element scrambling system.

2.2 Overlapping Frame Sliding Window Scramblers

The first technique involves dividing the clear speech signal into frames of n segments for some pre-selected n, where each segment represents T seconds of speech. For this technique it is necessary to choose a second integer $k<n$ which determines the system delay (in conjunction with the value T chosen for the segment length). In fact the total system delay will equal $(k+1)T$ seconds, and this is a relevant factor in the choice of k. Thus if $k=16$ and $T=3\times10^{-2}$, then the system delay will be 0.51 seconds. It is important to note that n does not affect this delay. Having chosen k we then select permutations from S_n for use in rearranging the speech segments.

We select and use these permutations so that each segment is transmitted within kT seconds of entering the scrambler. This is achieved by limiting the scrambling patterns to those permutations $\pi \in S_n$ satisfying:

$$[\pi(i)]_n \in \{[i-1]_n, [i-2]_n, \ldots, [i-k]_n\} \text{ for every } i \ (1 \leq i \leq n),$$

where $[i]_n$ represents the residue class of i modulo n.

We now describe both formally and by means of an example how these permutations are used. Formally, if we assume that time t=0 occurs at a frame boundary, then the segment transmitted between $t=(s-1)T$ and $t=sT$ is the segment input to the scrambler between $t=(r-1)T$ and $t=rT$, where $1 \leq s-r \leq k$, $r'=\pi(s')$, $[r]_n=[r']_n$ and $[s]_n=[s']_n$.

The way in which these patterns are used is illustrated in Figure 1, where we show a system for which $n=8$, $k=3$ and π satisfies $\pi(1)=6$, $\pi(2)=1$, $\pi(3)=8$, $\pi(4)=3$, $\pi(5)=2$, $\pi(6)=5$, $\pi(7)=4$ and $\pi(8)=7$. In the figure we have used different letters to distinguish between different frames. So, for instance, A1 is the first segement of the first frame while B1 is the first

segment of the second. Note that the values of n and k used in this example are not realistic in that they are much too small to offer any real security.

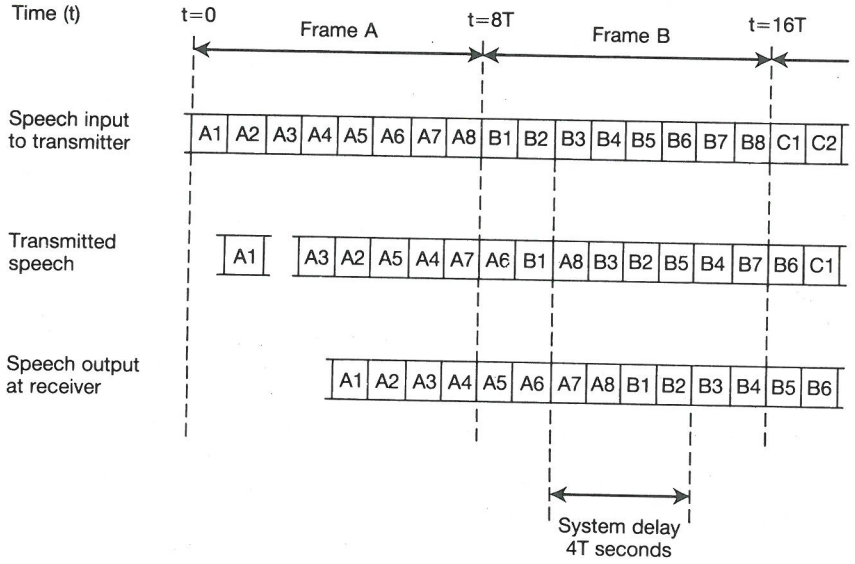

Fig. 1 Overlapping Frame Sliding Window TES

One advantage of this system is that it is not necessary to complete the transmission of the segments from one frame before commencing the transmission of segments from the next frame, thus increasing the diversity of patterns available. However this leads to practical implementation restrictions (discussed in more detail in [2] and [5]); in particular it is normally necessary to force the system to re-use the same segment permutation for a period of time. This in turn limits the security of this scheme.

2.3 *Disjoint Frame Sliding Window Scramblers*

In the second technique, we again choose a segment length T, and then also select a positive integer h, both of which values affect the system delay. We then select a second positive integer $n \geq h$, which determines the "size" of the permutations used. We then use as our scambling patterns those permutations $\pi \in S_n$ with the property that $|i-\pi(i)|<h$ for all i.

To do the scrambling we first divide the clear speech into frames of n consecutive segments, and a separate permutation is then used to determine how to rearrange the segments within each frame. Suppose that a frame of speech begins at time t=0 and ends at t=nT, and that permutation $\pi \in S_n$ is to be used to

re-order the segments within this frame. As before we label the segments 1,2,...,n so that segment i was originally spoken between t=(i-1)T and t=iT. The segments of the frame are then transmitted between t=hT and t=(n+h)T in such a way that, for any i between 1 and n, the segment transmitted between t=(h+i-1)T and t=(h+i)T is $\pi(i)$. The total system delay is then 2hT seconds.

As an illustration of this type of system consider Figure 2, where we give an example having n=8 and h=2. Suppose that π is used to permute the segments of the first frame, and τ is used for the second, where π,τ satisfy $\pi(1)=2$, $\pi(2)=1$, $\pi(3)=3$, $\pi(4)=5$, $\pi(5)=4$, $\pi(6)=7$, $\pi(7)=6$, $\pi(8)=8$, $\tau(1)=1$, $\tau(2)=3$, $\tau(3)=2$, $\tau(4)=5$, $\tau(5)=4$, $\tau(6)=6$, $\tau(7)=8$ and $\tau(8)=7$. As before, we label the segments of the first frame A1, A2, ..., A8 and the segments of the second frame B1, B2, ..., B8.

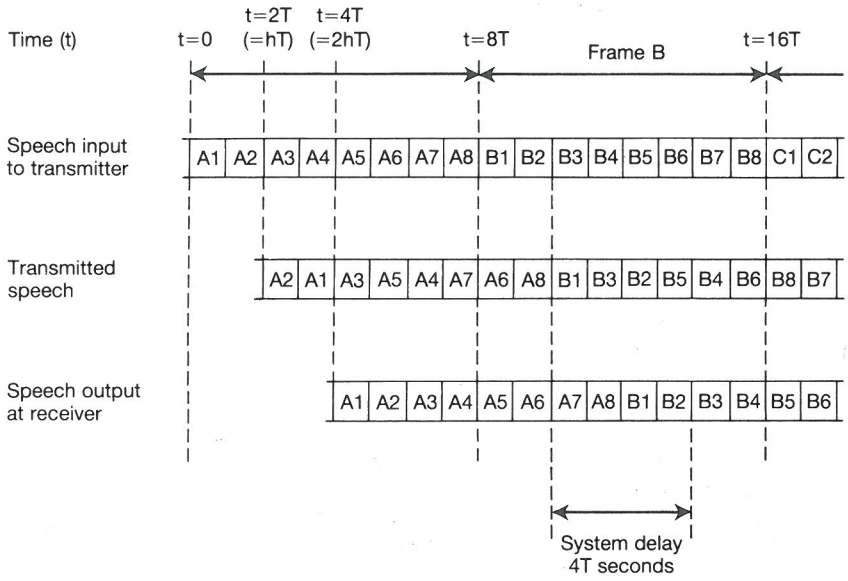

Fig. 2 Disjoint Frame Sliding Window TES

An important advantage of this system over the Overlapping Frame technique, is that it is now straightforward to arrange for a different permutation to be used for each frame of speech.

2.4 Permutation selection and enumeration

One problem which is common to both types of scrambler discussed here is the choice of usable permutations. This is of great importance because the variety of available scrambling

patterns to a great extent determines the security level of the system. For overlapping frame sliding window scramblers we are restricted to those permutations $\pi \in S_n$ which satisfy:

$$[\pi(i)]_n \in \{ [i-1]_n, [i-2]_n, \ldots, [i-k]_n \} \text{ for every } i.$$

We denote this set of permutations by $A^*(n,k)$, and if we denote the cardinality of this set by $a(n,k)$, i.e. $a(n,k)=|A^*(n,k)|$, then in order to assess the security of such a system we need to evaluate $a(n,k)$. In fact it can be shown that $a(n,k)$ is equal to the permanent of the cyclic (0,1) n by n matrix having (11...100...0) as its first row, where the number of ones in this row is k. Note that the permanent of an n by n matrix (a_{ij}) is

$$\sum_{\pi \in S_n} a_{1\pi(1)} \cdot a_{2\pi(2)} \cdots a_{n\pi(n)}$$

i.e. the definition is exactly the same as the definition of determinant except for the omission of the sign(π) term.

For disjoint frame sliding window scramblers the permutations we are interested in are those $\pi \in S_n$ which satisfy:

$$|\pi(i)-i| < h \text{ for every } i.$$

We call the set of all such permutations $C(n,h)$, and as before we let $c(n,h)=|C(n,h)|$. This enumeration problem also turns out to be equivalent to the evaluation of the permanent of an n by n (0,1) matrix, this time one with a 1 in position (i,j) iff $|i-j|<h$.

We now proceed to give results which enable these permanents to be readily computed for values which might be typical in genuine applications. Certainly, evaluating these permanents for such practical values of n, k and h would not be straightforward without some such result.

3. COUNTING SCRAMBLING PATTERNS AND PERMANENTS OF (0,1) MATRICES

Just as in [1], for the purposes of the theory that follows it is convenient to consider the set

$$A(n,k) = \{\pi \in S_n : [\pi(i)]_n \in \{[i]_n, [i+1]_n, \ldots, [i+k-1]_n\} \text{ for all } i\} \tag{3.1}$$

and it is clear that $|A(n,k)| = a(n,k) = |A^*(n,k)|$.

Before proceeding to specific results we need some notation. Firstly, if $\pi \in S_n$ and $i \in \{1,2,\ldots,n\}$ then define

$$X_k(\pi,i) = \{\pi(j) : [j]_n \in \{[i]_n,[i+1]_n,\ldots,[i+k-2]_n\} \text{ and}$$

$$[\pi(j)]_n \in \{[j]_n,[j+1]_n,\ldots,[i+k-2]_n\}\}. \qquad (3.2)$$

We can now state the following key lemma.

Result 3.1 (Lemma 4.1 of [1]) If $\pi \in A(n,k)$ then there exists an integer $r \in \{0,1,\ldots,k-1\}$ such that:

$$|X_k(\pi,i)| = r \text{ for every } i \in \{1,2,\ldots,n\}. \qquad (3.3)$$

Because of this result, for every $r \in \{0,1,\ldots,k-1\}$ we define:

$$A(n,k,r) = \{\pi \in A(n,k) : |X_k(\pi,i)| = r \text{ for every } i\}. \qquad (3.4)$$

In addition we set $a(n,k,r) = |A(n,k,r)|$. Using this notation, the following result is trivial.

Result 3.2 (Lemma 4.2 of [1])

$$a(n,k) = \sum_{r=0}^{k-1} a(n,k,r). \qquad (3.5)$$

To complete this introductory material now suppose that k and r are integers satisfying $0 \leq r \leq k-1$, and let $t = {}_{k-1}C_r$. Label the t distinct r-subsets of $\{0,-1,\ldots,-k+2\}$: $R_1,R_2,\ldots,R_t$, and then for every $i \in \{1,2,\ldots,t\}$ let $S_i = \{ j+1 : j \in R_i - \{0\} \}$. We now define the t by t (0,1) matrix $H(k,r) = (h_{ij})$ by:

$$h_{ij} = 1 \quad \text{iff} \quad S_i \subset R_j.$$

We can now state the following result which gives a direct means of computing a(n,k,r) and hence a(n,k).

Result 3.3 (Thereom 4.4 of [1])

$$a(n,k,r) = \text{trace}(H(k,r)^n). \qquad (3.6)$$

We now show how a similar result may be achieved for c(n,h) for the situation where $n \geq 2h-1 \geq 3$. This restriction on the values of n and h is necessary for the following theory, and is therefore implicitly assumed for the remainder of this section. In most speech scrambling applications the values of n and h

used will satisfy the above inequalities, a typical pair of values being n=16, h=8.

We first need some further definitions. Let $C^*(n,h)$ be the set: $\{\pi^* \in S_n$: there is $\pi \in C(n,h)$ with $[\pi^*(i)]_n = [\pi(i)+h-1]_n$ for all i$\}$ and clearly $c(n,h) = |C^*(n,h)|$. Then we have:

Lemma 3.4 $|C^*(n,h)| = |D(n,h)|$, where

$$D(n,h) = \{ \tau \in A(n+2h-2, 2h-1) : \tau(n+i) = n+i+h-1 \ (1 \leqslant i \leqslant h-1) \text{ and}$$

$$\tau(n+i) = i-h+1 \ (h \leqslant i \leqslant 2h-2) \ \}. \qquad (3.7)$$

Proof We establish the lemma by exhibiting a 1-1 correspondence between the elements of $C^*(n,h)$ and $D(n,h)$.

First suppose that $\pi^* \in C^*(n,h)$. Then let $\tau = \Phi(\pi^*)$ be the following element of S_{n+2h-2}:

$$\begin{aligned} [\tau(i)]_n &= [\pi^*(i)]_n, && h \leqslant \tau(i) \leqslant n+h-1, \quad 1 \leqslant i \leqslant n \\ \tau(i) &= i+h-1, && n+1 \leqslant i \leqslant n+h-1 \\ \tau(i) &= i-n-h+1, && n+h \leqslant i \leqslant n+2h-2. \end{aligned} \qquad (3.8)$$

Then Φ is a 1-1 correspondence between $C^*(n,h)$ and $D(n,h)$ and the result follows. □

We now need some further notation. Let $\underline{E}$ be the class of all (2h-2)-subsets E of $(-2h+3, -2h+4, \ldots, 2h-2)$ satisfying the property that E contains precisely h-1 elements of $\{-2h+3, -2h+4, \ldots, 0\}$. Further, if $E \in \underline{E}$, then let U(E) be the set of all (2h-2)-tuples $\underline{c} = (c_1, c_2, \ldots, c_{2h-2})$, where $\{c_1, c_2, \ldots, c_{2h-2}\} = E$ and $c_i \in \{ i-2h+2, i-2h+3, \ldots, i \}$ for every $i \in \{1,2,\ldots,2h-2\}$.

Then if $E \in \underline{E}$, and if $\underline{c} \in U(E)$, we let $v_n(E) = |P(\underline{c})|$, where $P(\underline{c})$ is defined to be the set of permutations $\pi \in A(n,2h-1,h-1)$ satisfying:

$$\pi(j) = \begin{cases} c_i & \text{if } \pi(j) < j \\ c_i + n & \text{if } \pi(j) \geqslant j \end{cases} \quad \text{where } j = n-2h+i+2, \quad 1 \leqslant i \leqslant 2h-2. \qquad (3.9)$$

The fact that $v_n(E)$ is well defined follows immediately from Lemma 5.3 of [1]. We can now state the following result whose proof is implicit in the proof of Theorem 4.4 of [1].

Result 3.5 As before let $R_1, R_2, \ldots, R_t$ be a labelling of the t distinct (h-1)-subsets of { 0, -1, ..., -2h+3 } (where $t = {}_{2h-2}C_{h-1}$). In addition let

$$C_i = \{ j+2h-2 : j \in \{0,-1,\ldots,-2h+3\} - R_i \}, \quad 1 \leq i \leq t,$$

and let the t by t matrix $W(n) = (w_{ij})$ be defined by

$$w_{ij} = v_n(R_i \cup C_j).$$

Then we have:

$$W(n) = H(2h-1,h-1)^{n-2h+2}. \tag{3.10}$$

Using this result in conjunction with Lemma 3.4 we now have

Theorem 3.6 If the labelling $R_1, R_2, \ldots, R_t$ is chosen so that $R_m = (0, -1, \ldots, -h+2)$, then

$$c(n,h) = \text{the}(m,m) \text{ entry in } H(2h-1,h-1)^n. \tag{3.11}$$

Proof First note that, by Lemma 3.4: $c(n,h) = |D(n,h)|$, where $D(n,h)$ is the subset of $A(n+2h-2,2h-1)$ defined in the statement of the Lemma. Then, using the notation following Lemma 3.4, it is straightforward to see that $|D(n,h)| = v_{n+2h-2}(E)$, where $E = \{-h+2,-h+3,\ldots,0\} \cup \{1,2,\ldots,h-1\} = R_m \cup C_m$. Note that to make this latter observation we need to establish that $D(n,h) \subset A(n+2h-2,2h-1,h-1)$. But this follows by noting that, by definition, $X_{2h-1}(\pi,n+1) = \{n+h,n+h+1,\ldots,n+2h-2\}$ for every $\pi \in D(n,h)$.

Hence $c(n,h) = v_{n+2h-2}(R_m \cup C_m)$. The theorem then follows immediately from Result 3.5. □

REFERENCES

[1] Beker, H.J. and Mitchell, C.J., (1987) Permutations with restricted displacement, *SIAM Journal on Algebraic and Discrete Methods*, **8**, *338-363*.

[2] Beker, H.J. and Piper, F.C., (1985) "Secure speech communications", Academic Press, London.

[3] Garey, M.R. and Johnson, D.S., (1979) "Computers and intractability: A guide to the theory of NP completeness", Freeman.

[4] Minc, H., (1984) "Permanents", Cambridge University Press.

[5] Mitchell, C.J. and Piper, F.C., (1985) A classification of time element speech scramblers, *J. Inst. Electronic and Radio Engineers*, **55**, 391-396.

Cryptography and coding

Based on the proceedings of a conference organized by The Institute of Mathematics and its Applications on Cryptography and Coding, held at the Royal Agricultural College, Cirencester on 15th–17th December 1986.

Edited by

HENRY J. BEKER
Zergo Consultants Ltd.

and

F. C. PIPER
Royal Holloway and Bedford New College

CLARENDON PRESS · OXFORD · 1989