THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
Printed in U.S.A.

Analog Scramblers for Speech Based on Sequential Permutations in Time and Frequency

By N. S. JAYANT, R. V. COX, B. J. McDERMOTT, and A. M. QUINN

(Manuscript received July 20, 1982)

Permutation of speech segments is frequently utilized in scramblers for analog speech privacy. This paper discusses a "sequential" *permutation procedure that has better segment-separation properties than the well-known procedure of* "block" *permutation, where contiguous segments are arranged in blocks of appropriate size, and permuted within such blocks. It further proposes the application of the sequential procedure to a novel technique for simultaneous permutations in time and frequency. The paper also presents results of a subjective experiment where we measured residual speech intelligibility at the output of scramblers using permutations in time [time segment permutation (TSP)], or permutations in time* and *frequency [time-frequency segment permutation (TFSP)]. The experiment included examples of block TSP, sequential TSP, and sequential TFSP. We measured spoken-digit-intelligibility as a function of the communication delay introduced by the scrambling operation. We found that even with a delay of 512 ms, the residual intelligibility in a TSP scrambler is no lower than about 50 percent; however, a sequential TFSP scrambler can realize an average digit intelligibility in the order of 20 percent with a delay of 256 ms. A companion paper discusses the implementation of the sequential TFSP scrambler, and the quality of descrambled speech in the context of real-channel operation.*

I. INTRODUCTION

Permutation of speech segments that are about 10 to 30 ms long is a bandwidth-preserving operation[1] that is frequently utilized in the design of scramblers for analog speech privacy.[2,3] The procedure,

known as Time Segment Permutation (TSP), lends itself to fairly robust real-channel operation[2] and efficient microprocessor implementation.[4] However, the reduction in speech intelligibility by this method is very small if the communication delay introduced by the scrambler and descrambler is constrained to be no longer than, say, 256 to 512 ms. This rather well-known deficiency was quantitatively demonstrated in a recent article[1] that discussed a segment scrambler, which will be referred to as "block" TSP in this paper (see Section 2.2). In this scrambler, contiguous speech segments are arranged in blocks of appropriate size, and permuted *within* such blocks. The entire block with permuted segments is then transmitted as scrambled speech, before proceeding to the next block. In this paper, we discuss another segment permutation procedure to be called "sequential" TSP (see Section 2.3). In this case, permutations are not constrained to be within blocks, and transmissions of scrambled speech are not constrained to be on a block-by-block basis. We will compare the two procedures in terms of how well they separate segments that are initially adjacent in the unscrambled speech. We will show that the sequential TSP has segment-separation properties that are much better than those of block TSP, but that, unfortunately, this is not accompanied by substantial gains in the residual intelligibility in scrambled speech, except for large values of communication delay (see Section 3.3).

To realize substantial reductions of intelligibility, it is imperative to use so-called two-dimensional approaches to scrambling.[1,3] One example of two-dimensional scrambling is the combination of block TSP and frequency inversion.[1] However, frequency inversion is a very straightforward, simple and time-invariant operation with only one possible input-output mapping, or "key." It therefore has no cryptanalytical strength. An important purpose of this paper is to propose and evaluate a two-dimensional procedure that offers a residual intelligibility very similar to that of block TSP plus frequency inversion, and a cryptanalytical strength that is much higher than in that method. This new procedure (Section IV) will be called Time-Frequency Segment Permutation (TFSP). In particular, we will be discussing a sequential version of this procedure, "sequential" TFSP.

A companion paper[5] discusses the implementation of the sequential TFSP scrambler, and the quality of descrambled speech in the context of a real-channel operation.

II. ONE-DIMENSIONAL SCRAMBLERS: BLOCK TSP AND SEQUENTIAL TSP

This section describes a sequential approach to segment-permutation and shows that it has much better segment separation properties

than a non-sequential, or block, procedure such as that discussed in Ref. 1. The block and sequential approaches to be discussed below are sometimes referred to as "*hopping window*" and "*sliding window*" approaches.[6]

2.1 Temporal distance d

The purpose of permutation scrambling is to reduce intelligibility by altering the normal time order of speech segments. The greater the separation in scrambled speech between normally adjacent segments, the lower the intelligibility is expected to be. Conversely, adjacent samples in the scrambled speech should be well separated in normal speech. An important result of this paper is that average segment separation and intelligibility are generally, if not always, monotonically related. It is useful, therefore, to define the following objective measure of effectiveness for permutation scramblers as the *temporal distance between a pair of segments in normal speech that appear as adjacent segments in scrambled speech.* For simplicity, we will henceforth refer to this non-zero, positive parameter merely as the "temporal distance d." By definition,

$$d = 1 \quad \text{for adjacent segments in unpermuted speech}$$

$$d \geq 1 \quad \text{for adjacent segments in permuted speech.} \qquad (1)$$

Illustrations of temporal distance appear in Fig. 1. The scramblers used in this figure will be defined in the next two sections.

Briefly, block TSP is a procedure where an entire block of segments in scrambler memory is transmitted before proceeding to the next block. Segment selection involves a number of candidates that decreases from the initial number of segments in the block to 1, as a given block is processed. In sequential TSP, each stage of segment

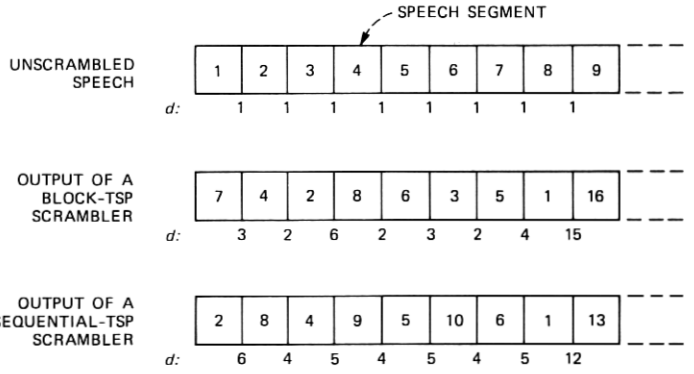

Fig. 1—Temporal distance d.

selection involves a constant number of candidates equal to the maximum number of segments that can be stored in scrambler memory.

2.2 Block TSP

Figure 2(a) defines the block approach to segment scrambling. In this approach, the scrambler memory consists of a block of b' contiguous segments; one can identify in the scrambler output a corresponding block whose member segments are the same as the segments that comprise the input block. A succeeding block (block 2 in Fig. 2a), also comprising b' segments, enters the scrambler memory after all the b' segments of the preceding block (block 1 in Fig. 2a) have been processed and transmitted. Table Ia depicts a realization of the random processes in block TSP for the example of $b' = 8$ (Fig. 1). Shown are the successive contents of scrambler memory, the sequence of transmitted segments, and the temporal distance d between adjacent segments in the scrambler output.

Note that transmitted segment s and temporal distance d are both random variables. In a practical implementation, the variable s will be pseudorandom so that the intended receiver can invert the scrambler operation. What is significant is that the maximum value of d in the table is 15. This is indeed a global maximum for a block TSP scrambler with memory $b' = 8$. This maximum separation is attained when the random permutation is such that the *last* segment of block n [segment 16 from block $n = 2$ in Table Ia immediately follows the *first* segment of block $n - 1$ (segment 1 from block $n = 1$ in the table)]; and in general,

$$\max(d) = 2b' - 1 \text{ (in segments)}. \tag{2}$$

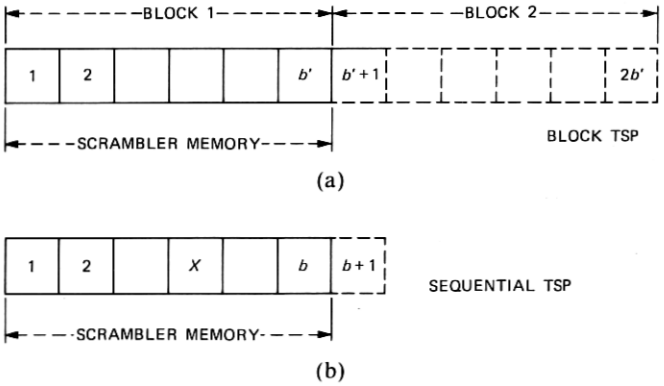

Fig. 2—Schematic representations of TSP scramblers using: (a) block and (b) sequential modes. In (a), an entire block of segments is transmitted before proceeding to the next block. Segment selection involves a number of candidates that decreases from b' to 1, as a given block is processed. In the sequential case (b), each stage of segment selection involves a constant number of candidates equal to the scrambler memory b.

Table I—Illustration of two TSP algorithms where **m** is the contents of scrambler memory, s is the transmitted segment, and d is the temporal distance to previous transmitted segment

(a) Block TSP ($b' = 8$)										(b) Sequential TSP ($b = 8, t = 16$)									
m								s	d	**m**								s	d
1	2	3	4	5	6	7	8	7	—	1	2	3	4	5	6	7	8	2	—
1	2	3	4	5	6	8		4	3	1	3	4	5	6	7	8	9	8	6
1	2	3	5	6	8			2	2	1	3	4	5	6	7	9	10	4	4
1	3	5	6	8				8	6	1	3	5	6	7	9	10	11	9	5
1	3	5	6					6	2	1	3	5	6	7	10	11	12	5	4
1	3	5						3	3	1	3	6	7	10	11	12	13	10	5
1	5							5	2	1	3	6	7	11	12	13	14	6	4
1								1	4	1	3	7	11	12	13	14	15	1	5
9	10	11	12	13	14	15	16	16	15	3	7	11	12	13	14	15	16	13	12
9	10	11	12	13	14	15		12	4	3	7	11	12	14	15	16	17	3	10
9	10	11	13	14	15			10	2	7	11	12	14	15	16	17	18	18	15
9	11	13	14	15				9	1	7	11	12	14	15	16	17	19	14	4
.	.	.						.	.	.	.	.	.	.	.	.	.	.	.
.	.	.						.	.	.	.	.	.	.	.	.	.	.	.

The communication delay C in block TSP scrambling is the sum of two delays: (i) a delay of $b' - 1$ at the transmitter (the additional time for completion of the block subsequent to the arrival of segment 1), followed by (ii) an additional delay of b' at the receiver (the maximum time for which the descrambler may have to wait before it has access to the permuted segment 1). As a result,

$$C = 2b' - 1 \text{ (in segments)}. \tag{3}$$

2.3 Sequential TSP

Figure 2(b) defines a sequential approach to scrambling. In this approach, processing proceeds one segment at a time rather than one block at a time. The permutation process is constrained by two parameters: the scrambler memory b, and the maximum time t (in multiples of segment duration) that a segment is allowed to stay in scrambler memory. When the first segment [such as x in Fig. 2(b)] is pseudorandomly selected for transmission and released, the contents of the scrambler memory to the right of x are shifted to the left by one unit, and the last position is *immediately* occupied by segment $b' + 1$. The next transmission is based on another random selection that, unlike in the block approach of Fig. 2(a), can involve the newest segment $b' + 1$, which was not a member of the original block. The above process continues on a segment-by-segment basis, with one constraint mentioned earlier: if a segment is retained in scrambler memory for $(t - 1)$ units of time, it is unconditionally released for transmission at time t, even if this means that the dictates of the random segment selector algorithm should be overridden. An important property of the sequential scrambler is that every segment has an

equal probability of spending the maximum allowed time in scrambler memory. To permit a meaningful comparison with block scrambling, we shall concentrate on the special case of

$$t = 2b \tag{4}$$

in most of the ensuing discussion. With the above specific design for t, the total encoding delay will be the same for both block and sequential scramblers for a given scrambler memory. This will be clear from subsequent discussion [see eq. (6)]. The design in (4) is also known to maximize the total number of unique permutations for a given delay and block length.[6] Results with a sequential TFSP scrambler indicate that (4) is also an optimal design from the viewpoint of residual intelligibility (see Section V).

Table Ib depicts a realization of the random process in sequential TSP for the example of $b = 8$ and $t = 16$ (Fig. 1). Shown once again are the successive contents of scrambler memory, the sequence of transmitted segments, and the temporal distance d between adjacent segments in the scrambler output. Note in this example that segment 1 indeed stays in scrambler memory for the maximum of 16 time units. It stays in the extreme left-hand slot of scrambler memory for $t - b = 8$ time units; subsequent segments that succeed in reaching the extreme left slot tend to have maximum allowed staying times less than $t - b$ in that slot. Note also that, as in Table Ia, the temporal distance d is a random variable with a maximum value of 15, equal to the maximum in block TSP. In fact this is a global maximum for the sequential design $b = t/2 = 8$; and in general, as in (2),

$$\max(d) = t - 1 = 2b - 1 \text{ (in segments)}. \tag{5}$$

In block TSP, the maximum separation (2) can be realized only in output segment pairs that involve the first and last segments of adjacent blocks (adjacent output segments 1 and 16 in Table Ia). In sequential TSP the maximum separation (4) can be realized in more general instances (for example, with adjacent output segments 3 and 18 in Table Ib).

The communication delay in sequential TSP is given by

$$C = t - 1 = 2b - 1 \text{ (in segments)}. \tag{6}$$

This is a delay inherent in the scrambling parameter t. There is no additional delay at the descrambler because of the absence of a block operation. After an initial waiting time of $t - 1$ segments, the descrambler has a guaranteed access to every consecutive segment needed to reconstitute the original input signal.

Table II provides a summary comparison of block and sequential approaches. The parameter $\min(d)$ in the last row will be explained

Table II—Summary comparison of block and sequential TSP

	Block TSP	Sequential TSP (special case, $t = 2b$)	Sequential TSP (general case, $t > b$)
Scrambler memory	b'	b	b
Total communication delay $C_t = C + 1$	$2b'$	$2b$	t
Maximum temporal distance $max(d)$ between adjacent output segments	$2b' - 1$	$2b - 1$	$t - 1$
Minimum temporal distance $\min(d)$ that can be specified between adjacent output segments	1 if $b' < 8$ 2 if $b' \geq 8$	$[(b + 1)/2]$ $[x]$: greatest integer $< x$	

presently. The total communication delay C_t includes a delay of 1 unit that is inherent in buffer read-in and read-out, and hence equals $C + 1$ segments. If the duration of a time segment is B ms, the delay in ms equals C_tB. In all of this paper, $B = 16$ ms.

2.4 min(d) and $\bar{d}$

The maximum temporal distance between adjacent output segments has been discussed above and shown to be a function of scrambler memory size. This section will provide further characterization of the random variable d, in particular, the minimum value of d that can be specified a priori, the probability density function of d, and its average value $\bar{d}$.

An important property of sequential TSP—one that is not shared by block TSP—is that the selection of a segment for transmission always involves a constant number of candidate segments; this number is equal to the scrambler memory parameter b. A consequence of this property is that it is possible to specify in general a minimum distance $\min(d) > 1$ in the output of the sequential scrambler. The random number generator that dictates output segment selection is simply resampled repeatedly, if needed, until the output has a distance of at least $\min(d)$ from its predecessor in the output sequence. The intended descrambler is assumed to know the "key," or the random number sequence used by the scrambler. The descrambler does not need to know the value of $\min(d)$ implied by that key.

The maximum value of $\min(d)$ that will ensure a legitimate scrambler output depends both on scrambler memory b and the contents thereof. A globally safe value (one that will not cause algorithm "hanging," as explained below) is

$$\min(d) = [(b + 1)/2], \tag{7}$$

where $[x]$ denotes the greatest integer less than x. The only occasion when (7) will have to be violated in scrambler operation is when a

segment has spent its full life span t in scrambler memory, and it has to be released unconditionally without regard to its temporal distance from the most recent output.

Table III illustrates how the scrambling algorithm "hangs" when a $\min(d)$ greater than the value in (7) is specified a priori. In this example, $b = 8$, $t = 16$, $\min(d) = 6$, a value that exceeds the limit of 4 suggested by the formula (7).

As indicated in the last row of Table II, the $\min(d)$ values that can be specified a priori are much smaller in block TSP. For example, if $b' = 8$, after a possible output sequence of [3 5 7 4 6 8 2], segment 1 can never be transmitted (immediately after segment 2) unless $\min(d) = 1$. Even if the $\min(d)$ requirement is overridden in the case of the last transmitted segment of a block, the greatest value of $\min(d)$ that will not result in a "hanging" of the scrambler operation is a very slowly increasing function b. For example with $b = 8$, the $\min(d)$ value that can be specified a priori is no greater than 2. This should be compared with the value of 4 for sequential TSP with $b = 8$.

In both block and sequential TSP schemes, a priori insistence on a $\min(d)$ value greater than unity has attendant penalties in the cryptanalytical strength of the scrambler.[6] This is due of course to the fact that with $\min(d) > 1$, fewer random segment permutations are legal, in comparison with the totally random situation that obtains with $\min(d) = 1$.

An interesting property of a sequential scrambler with $\min(d) = 1$ and the constraint $t = 2b$ is that the fraction of segments that spends the maximum allowable time in scrambler memory is very nearly 20 percent for values of $b > 4$. An analytical demonstration of this property appears in the appendix.

Figure 3 shows histograms of the random variable d in block TSP with $b' = 8$ and sequential TSP with $b = t/2 = 8$, and two values of $\min(d)$ in each case. The results are from a simulation involving a total of 132 segments (about 4 seconds of speech, with 32 ms segments). Note that, in general, the probability of d-values *less than* $\min(d)$ is very small. The probability is non-zero, however, because of occasional situations where a segment has spent the maximum lifespan of $t = 2b$

Table III—Illustration of an unrealizable $min(d) = 6$ in sequential TSP with $b = t/2 = 8$

Contents of Scrambler Memory m								Transmitted Segment s	Temporal Distance d
1	2	3	4	5	6	7	8	2	—
1	3	4	5	6	7	8	9	8	6
1	3	4	5	6	7	9	10	1	7
3	4	5	6	7	9	10	11	7	6
3	4	5	6	9	10	11	12		≥6

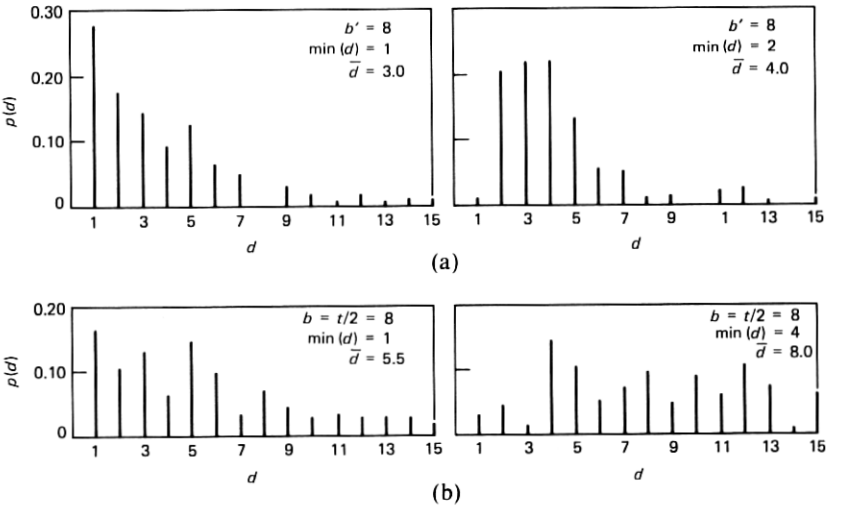

Fig. 3—Histograms of temporal distance d in (a) block TSP ($b' = 8$) and (b) sequential TSP ($b = 8$, $t = 16$). The non-zero probability of $d < \min(d)$ is related to violations of $\min(d)$ because of unconditional releases of segments that have spent a full life-span = 16 in scrambler memory.

times in scrambler memory, and it becomes necessary to release it unconditionally without regard to the resulting value of d. Figure 3 also notes respective values of average separation $\bar{d}$. Note that this depends on scrambler type as well as on min d. It increases with $\min(d)$ for both types of scramblers, and it is greater for sequential TSP than for block TSP, for the case of $\min(d) = 1$.

Figure 4 compares $\bar{d}$ values for block and sequential TSP scramblers as a function of scrambler memory. Results for the sequential system correspond to the special case of $t = 2b$. Note that $\bar{d} \sim b$ in this case, one-half of $\max(d) = 2b$. The faster increase of $\bar{d}$ in sequential TSP is a result of the greater values of $\min(d)$ that can be specified in this system. Recall from Tables I and II that $\max(d)$ is the same for block and sequential systems for a given value of scrambler memory.

III. RESIDUAL INTELLIGIBILITY IN BLOCK AND SEQUENTIAL TSP SCRAMBLERS

The residual digit intelligibility in a block TSP system has been the subject of a recent comprehensive article.[1] The emphasis in this section is on the digit intelligibility performance of sequential TSP, as evaluated in formal listening tests.

3.1 Test conditions

The duration of speech segments was $B = 16$ ms for all schemes. This is a design that provides a useful compromise between the

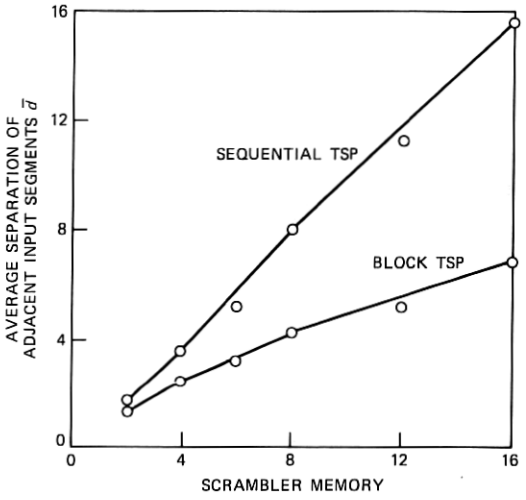

Fig. 4—Average temporal distance $\bar{d}$ as a function of scrambler memory in (*i*) block TSP and (*ii*) sequential TSP with $t = 2b$. Minimum specified distances are the min(d) values in Table II.

conflicting requirements of encoding delay and bandwidth expansion.[1] Total communication delay (C_t) and minimal separation min(d) were variables in the test. The three delay conditions that were tested were 256, 384, and 512 ms. The min(d) values that were combined with each delay condition were the smallest and largest values that could be successfully assigned for each type of scrambler. Of course, the smallest value of min(d) in all cases is 1, which would, at most, allow adjacent segments to be in reverse order when transmitted. However, the maximum value that min(d) can assume depends upon the type of scrambler and the length of the memory buffer, as shown in Table II. The three delay conditions at each of two min(d) values generated six test conditions for each type of scrambler, block and sequential.

One other condition was also included with the sequential scrambler using a delay of 512 ms. This included a μ-law logarithmic compression at the scrambler output, with $\mu = 100$.

Before making the test tapes, we listened to recordings of four-digit numbers (as described in Section 3.2 below) as processed by each of the test conditions. Two sets of recordings were employed, one from a male talker and one from a female talker. The recorded speech bandwidth was 200 to 3200 Hz in each case. The consensus of opinion in this pilot listening session was that the intelligibility was higher for the recordings of the female talker, presumably because her speaking rate was slower. The average duration of the four-digit utterances was 2.34s for the female talker, as compared to 1.99s for the male talker. Recordings from both talkers were included for the final test.

The speech samples were 50 four-digit numbers such as 3860, spoken as: *three-eight-six-zero*. Each talker recorded a different list of fifty numbers. The list of numbers was balanced so that within every set of ten numbers, each of the ten digits occurred four times, and each occurrence of a given digit was at a different position in a number. Although the same list of 50 numbers was used for all the tests with a given talker, each subset of ten numbers was presented in a different random order in each test.

The use of digits rather than continuous speech as test inputs follows the procedure in previous tests.[1] Conversational speech has redundancies that make the task of an analog scrambler more difficult; sentence intelligibility, for example, would be higher than word intelligibility as a result of these redundancies. There are no such redundancies in the digit strings in our tests, these strings being sequences of randomly chosen digits. But still, our experience with analog scramblers indicates that digit intelligibility, measured as discussed, is a fairly critical test of scrambler performance. The fact that there is a limited stimulus vocabulary (of ten) makes the task of the analog scrambler quite difficult, perhaps more so than in the case of a continuous speech input, which has a much larger, albeit redundant vocabulary. In the context of scrambled speech,[3] as well as in the context of speech corrupted by additive noise,[7] there is clear evidence that digit intelligibility scores tend to be much higher than word intelligibility scores.

3.2 Test procedure

The subjects were employees of Bell Laboratories at Murray Hill, New Jersey. They were not formally trained listeners of scrambled speech. Each scrambler scheme was judged by 18 subjects, although each subject judged only six schemes, chosen to represent the range of expected intelligibility. The subjects listened to the recordings through earphones while seated in a sound-treated booth. They were told to listen to each number and write the four digits they heard on their answer sheets. They were also told that some of the numbers would be difficult to understand and, if they were uncertain, they were to write their best guess rather than to leave blanks.

3.3 Results

For each scrambler type, the mean and standard deviation of the percent-correct identification of the digits was computed for each digit position of the four digit numbers. These values confirmed two observations that we had made in the pilot test mentioned earlier.

The mean intelligibility scores confirmed our observation about the effect of the slower speaking rate (by the female talker) on TSP performance. The effect was most apparent in the scores for the two

voices with the sequential scrambler and $C_t = 512$ ms. While the mean intelligibility scores for the two middle digits of this condition were only 66 and 61 percent with the male talker, the same scores were 84 and 91 percent for the female talker.

In the preliminary listening session, we had also noticed that the first and last digits of the four-digit numbers seemed to be easier to identify than the two in the middle. To see whether this was also true in the test data, the percent-correct identification was computed for each digit position. The scores illustrated quite clearly that the digit position affects the residual intelligibility. At each time delay, the percent correct for the third-digit position has the lowest value and the fourth digit position the highest value. Digit positions one and four have less context than positions two and three in the sense that digit position one has no left neighbor and digit position four has no right neighbor. The consistently lower scores of the third digit position are very likely due to the strong influence of context. Indeed, the scores at this digit position are probably a better indication of the level of intelligibility that could be expected in continuous speech where pauses are less frequent.

Intelligibility scores showed that there is no general advantage in using $\min(d)$ values greater than one in sequential TSP. The result is surprising in view of the effect of $\min(d)$ on the average temporal separation $\bar{d}$ (see Fig. 3). One possible explanation of the result is the presence of some kind of a threshold effect in the perception of temporally scrambled speech.

The upper right-hand corner of Fig. 5 shows residual intelligibility in sequential TSP scrambling as a function of communication delay, for $\min(d) = 1$, and for input conditions most favorable to the scrambler—digit position three and male talker. The lower edge of the cross-hatched region refers to the same most favorable case, while the upper edge refers to the average, over both male and female talkers, and over all four digit positions.

Intelligibility scores for block TSP were significantly different from those of sequential TSP only in the case of a communication delay equal to 512 ms. The dashed-line block TSP characteristic in Fig. 5 refers to the most favorable case of male speaker and digit position three, and to the (only available) example of $\min(d) = 2$.

The results of Fig. 5 indicate that there is no intelligibility advantage in sequential TSP as compared with block TSP at low values of delay C_t. When C_t is increased to 512 ms, the sequential approach produces a significant reduction of about 10 percent over the block approach. The condition involving μ-law compression of speech reduces the residual intelligibility even more, as shown by the point marked $\mu = 100$. The refinement of μ-law compression is simple to implement,

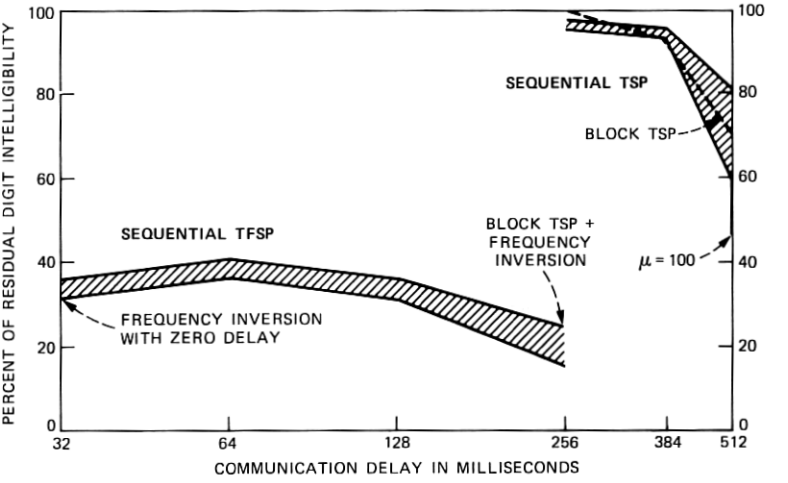

Fig. 5—Mean intelligibility of digits as a function of delay for sequential TSP sequential TFSP scramblers. Upper curves refer to averages over all four digit positions and over both male and female speakers. Lower curves refer to scores for the third digit position for the male speaker. The dashed lines refer to block TSP, digit position three, and male speaker.

although a penalty paid for reduced intelligibility in this case is increased bandwidth expansion and real-channel sensitivity, as compared with the conventional case of $\mu = 0$.

3.4 Discussion

It is interesting that sequential TSP, in spite of its better segment separation properties, provides an insignificant reduction of intelligibility as compared with block TSP, at low values of delay C_t. However, this result can be reconciled with the objective results in Fig. 4, which show that the increase of average speech segment separation owing to sequential scrambling is an increasing function of scrambler memory. In the "one-dimensional" procedure of time segment permutation, the only way of providing greater scrambling memory is by increasing delay and, apparently, the objective separation gain becomes perceptually significant only at C_t values in the order of 512 ms.

In the task of identifying one of ten possible digits, the lowest meaningful intelligibility score is 10 percent, corresponding to purely random guessing. This lower bound is particularly meaningful if listeners do not use complex cues and indeed perform decision tasks with ten alternatives. Clearly, none of the TSP scramblers in this study approaches a score in the order of 10 percent. This reinforces our earlier stand[1] that time permutation is best used in conjunction with frequency manipulations such as frequency inversion or frequency band permutations to provide practical and useful values of residual

intelligibility. The much lower residual intelligibility of TSP with frequency inversion is illustrated by the 25-percent result in Fig. 5, for the case of block TSP and $C_t = 256$ ms, a condition tested in earlier work.[1] In that study, frequency inversion alone provided a residual intelligibility of 30 percent. Unfortunately, however, frequency-inverted speech has identifiable characteristics that can be learned,[8] and frequency inversion is also very easy to undo. The next section describes another "two-dimensional" procedure for analog scrambling; it employs sequential permutations of a time-frequency speech matrix using sub-band partitions of 16 ms time segments. This two-dimensional procedure also realizes a residual intelligibility on the order of 15 to 25 percent. In addition, it has better cryptanalytical properties than TSP with frequency inversion.

IV. PERMUTATIONS WITH A TIME-FREQUENCY MATRIX: TIME-FREQUENCY SEGMENT PERMUTATION (TFSP)

In this section, we propose a scrambler that provides a simultaneous and fully two-dimensional manipulation of both time and frequency information in speech. The permutations of time-frequency segments are based on the sequential permutation approach described in Section 2.3.

The basic principle of the proposed scrambler can be explained with reference to Fig. 6. The $(f \times b)$ matrix depicts a total of fb time-frequency segments. These belong to b contiguous time segments of speech, each of which is split into f contiguous frequency sub-bands or segments. The scrambler memory is considered to be equal to the product fb. For subsequent discussions, the contents of this memory can be considered to be a one-dimensional array of fb time-frequency segments. A random number algorithm picks one of these fb segments for transmission. When this p-th segment is transmitted, the contents of all q-th cells $(q > p)$ are promoted by one position in the memory, and the fb-th cell is filled by an incoming $(fb + 1)$-th time-frequency

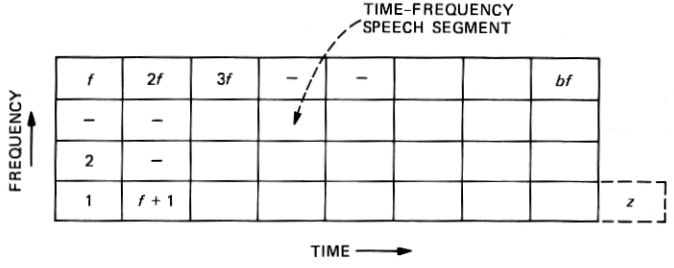

Fig. 6—Sequential permutations of a time-frequency speech matrix: sequential TFSP. Time-frequency segment z enters scrambler memory as soon as p-th segment in memory $(1 < p \leq bf)$ is randomly selected and transmitted.

segment. This random scrambling procedure is repeated with two constraints previously described in the discussion of sequential TSP (see Section II):

(*i*) No time-frequency segment stays in the scrambler memory matrix for a number of stages greater than $t_f = 2bf$. This corresponds to the special case $t = 2b$ in the one-dimensional case [Section 2.3; eq. (4) and Table II]. It implies a total communication delay of $C_t = 2fb$ segments. However, since f successive frequency segments are combined to produce one time segment, the total communication delay is independent of f, and given as in the one-dimensional case, by

$$C_t \text{ (in ms)} = 2bB,$$

where B is the duration (in ms) of a time segment. As in Sections II and III, $B = 16$ ms.

(*ii*) Contiguous time-frequency segments in scrambler output can be arranged to have a separation which, with a high probability, exceeds $\min(d_f)$; $1 < min(d_f) \leq fb/2$. The parameter $\min(d_f)$ provides a trade-off between average segment separation in scrambler output, which is an increasing function of $\min(d_f)$, and total number of possible permutations, which is a decreasing function of $\min(d_f)$.

Every set of f successive outputs from the scrambler matrix is reconstituted into a pseudo-speech time segment for transmission. One-dimensional sequential TSP is the special case of $f = 1$ in Fig. 6.

V. RESIDUAL INTELLIGIBILITY IN A SEQUENTIAL TFSP SCRAMBLER

Residual intelligibility tests were conducted following the same general procedures used for the block and sequential TSP tests (see Sections 3.1 and 3.2). The same original recordings of four-digit numbers spoken by a male and female talker were used as speech inputs. Eighteen subjects listened to the recorded digits as processed by each test scheme.

5.1 Test conditions

A total of seven conditions were tested, with the simple design of $\min(d_f) = 1$ in each case. The first four conditions correspond to the $t_f = 2bf$ design, with $f = 4$ and $b = 1, 2, 4$, and 8. Corresponding memory sizes are 4, 8, 16, and 32 time-frequency segments. Corresponding communication delays are $2bB = 32b = 32$, 64, 128, and 256 ms. The other three conditions tested used $b = 8$ also, but values of $t_f \neq 2bf$; specifically, $t_f = 1.5bf$, $3bf$, and $4bf$. In other words, the memory size is fixed at 32 segments for these three cases, but the maximum age in memory varies from 1.5 to 4 times the memory length. The communication delays corresponding to these three conditions are 192, 384, and 512 ms.

5.2 Results

The mean and standard deviation of the percent-correct identification was computed for each digit position of each test condition. In general, the subject variability was not large. The average standard deviation was about 6 percent for the male voice and slightly higher, 9 percent, for the female voice. However, there was one subject whose scores were consistently very high, 1.5 to 2.25 standard deviations above the mean. (On one test he correctly identified 72 percent of the digits in the third position while the average for the remaining subjects was 37 percent.) When questioned, he claimed that he did not have any special strategy. However, a closer examination of his data showed a pattern that was evident to some extent in the scores of other better-than-average listeners. The data suggest that these listeners would focus their attention on one of the four digit positions, even when the pauses between digits were difficult to detect.

Since the four digit numbers were balanced and presented in groups of ten, their scores at each digit position for each group could be compared. For instance, for one type of scrambler, the unusual subject mentioned above had scores of 0.20, 0.40, 0.10, and 0.60 for the four digit positions of one group of ten numbers. For the next group of ten numbers, his scores were 0.60, 0.50, 0.30, and 0.10, suggesting that he had shifted his attention to the first two digit positions while listening to the second group of ten numbers. Because of the difference owing to the speaking rate of the two talkers, the mean intelligibility (across subjects and digits) was compared for the two talkers. Even though the differences in the average scores were small—on the order of 5 to 8 percent—the scores for the female voice were consistently higher and, in all cases but one, the difference was statistically significant.

In general, the scores for different digit positions did not display the context effect mentioned for one-dimensional TSP. On the whole, about 90 to 95 percent of the subjects' scores were not significantly different for different digit positions. The remaining 5 to 10 percent of the subjects, who had significantly different scores owing to digit position, were generally the listeners whose overall scores were higher than the average. Although the effects of talker and digit position were not as strong as in the TSP experiment, to be consistent with those results, the scores of the third digit position with the male voice were again evaluated separately, and considered as the closest approximation to scores with continuous speech. These values are indicated by the lower edge of the cross-hatched TFSP region in the lower part of Fig. 5. The upper edge of this region refers, once again, to averages over both male and female talkers, and over all four digit positions.

Four observations are worth noting in the TFSP characteristic of Fig. 5: (*i*) the intelligibility with the smallest communication delay (32

ms) is close to that in frequency inversion (which has a zero communication delay); (*ii*) the intelligibility with a delay of 256 ms is close to that in block TSP (with the same delay) plus frequency inversion; (*iii*) there is a significant drop in intelligibility when the delay exceeds 128 ms (corresponding to a 4 × 4 time-frequency matrix in Fig. 6), and, finally, (*iv*) the intelligibility with a 256-ms delay is significantly higher than the expected lower bound of 10 percent; the characteristic, however, shows a tendency to drop further at delays greater than 256 ms, and with the suggested design of $t_f = 2bf$.

The differences among the scores with the design $t_f = 2bf$ and delays less than 256 ms are not statistically significant, but these scores are all significantly different from the score of 15 percent at a delay of 256 ms. For the case of $b = 8$, the scores at the other three values of t_f ($1.5bf$, $3bf$, and $4bf$) were also not significantly different from each other, but they were all significantly higher than the score when $t_f = 2bf$ (the 256-ms point in Fig. 5). This result suggests that the maximum staying time of $2bf$ may represent an optimal design that minimizes identification. This result is very interesting because the last two of the three t_f conditions above involve communication delays that are 50 and 100 percent greater than the delay in the $t_f = 2bf$ design.

A significant property of all the analog scramblers in this paper is that intelligibility is digit-dependent. This is shown by the illustrative confusion matrices of Table IV. As seen from the diagonal terms in these matrices, digits six and five are the most difficult to scramble. The very high residual intelligibilities for these digits expose what may be an inherent limitation of analog scramblers, at least those based on permutations, as opposed to digital scramblers that transform any given input to an output that sounds like white noise. The fact that the output of the analog scramblers discussed is not the white-noise type is well illustrated by the spectrograms of Fig. 7. Because of the residual structure in these spectrograms, they can be used as the starting point for non-real-time descrambling by a trained eavesdropper. This is especially the case with the TSP spectrogram of Fig. 7b.

The matrices in Table IV also show that the male speaker was easier to scramble than the female speaker. As stated earlier, we feel that this is due to the slower speaking of the female speaker.

5.3 *Interpretation of digit-intelligibility scores*

An important consideration in interpreting the results of the tests described in this paper is the use of spoken digits as speech input. The lower bound of 10 percent is particularly meaningful if the information available to the listener is limited only to the possibility of the ten digits. Actually, the subjects are trying to recognize phonemes and the phonemes of the spoken words for each digit are not a balanced sample

Table IV—Digit confusion matrices in a TFSP scrambler with communication delay = 256 ms (results are averages over 18 listeners)

		LISTENERS' RESPONSE										
		0	1	2	3	4	5	6	7	8	9	
	0	*0.30*	0.08	0.05	0.05	0.10	0.09	0.05	0.11	0.05	0.11	
S	1	0.09	*0.17*	0.05	0.05	0.06	0.10	0.03	0.10	0.07	0.27	
C	2	0.14	0.05	*0.18*	0.19	0.02	0.07	0.18	0.07	0.05	0.05	
R I	3	0.14	0.08	0.15	*0.14*	0.11	0.07	0.15	0.07	0.05	0.03	
A N	4	0.08	0.12	0.04	0.05	*0.23*	0.18	0.10	0.06	0.08	0.05	FEMALE
M P	5	0.03	0.06	0.02	0.04	0.07	*0.57*	0.03	0.06	0.07	0.07	SPEAKER
B U	6	0.03	0.05	0.03	0.04	0.08	0.09	*0.53*	0.09	0.04	0.03	
L T	7	0.06	0.11	0.03	0.07	0.07	0.15	0.10	*0.28*	0.07	0.07	
E	8	0.06	0.13	0.07	0.06	0.06	0.14	0.15	0.07	*0.23*	0.03	
R	9	0.08	0.10	0.04	0.06	0.04	0.22	0.02	0.06	0.06	*0.32*	

		LISTENERS' RESPONSE										
		0	1	2	3	4	5	6	7	8	9	
	0	*0.24*	0.12	0.08	0.05	0.08	0.11	0.09	0.08	0.09	0.06	
S	1	0.08	*0.17*	0.06	0.08	0.09	0.19	0.06	0.09	0.09	0.09	
C	2	0.22	0.10	*0.13*	0.08	0.10	0.05	0.18	0.08	0.04	0.03	
R I	3	0.14	0.11	0.09	*0.20*	0.10	0.06	0.10	0.10	0.04	0.06	
A N	4	0.08	0.13	0.07	0.07	*0.18*	0.15	0.11	0.10	0.03	0.07	MALE
M P	5	0.09	0.11	0.02	0.03	0.06	*0.40*	0.04	0.11	0.08	0.08	SPEAKER
B U	6	0.11	0.07	0.09	0.09	0.07	0.05	*0.35*	0.08	0.06	0.04	
L T	7	0.15	0.12	0.06	0.06	0.08	0.07	0.16	*0.16*	0.09	0.05	
E	8	0.09	0.13	0.09	0.07	0.09	0.09	0.12	0.12	*0.13*	0.08	
R	9	0.18	0.11	0.04	0.09	0.09	0.12	0.08	0.10	0.10	*0.08*	

of the English language. For instance, eight is the only spoken digit with an initial vowel, seven and zero are the only words with two syllables, and five of the ten spoken digits have unvoiced fricatives as the initial phoneme (three, four, five, six, seven). Thus, if an astute listener recognized the first phoneme as an unvoiced fricative, then the probability of being correct by guessing is 20 percent rather than 10 percent. If, in addition, the word were recognized as having only one syllable (eliminating seven), the probability would be 25 percent.

In the first experiment of this series (Section III), the unusually high scores for the spoken digit "six" were observed, but a more detailed analysis was not done. In the analysis of the TFSP data, the scores for each of the spoken digits were computed and compared as shown in Fig. 8. The mean intelligibility of each spoken digit (with digit position disregarded) is indicated for both the male (M) and female talker (F), for scramblers with $t_f = 2fb$ and $b = 1$, 2, 4, and 8. Labels 1, 2, 4, and 8 refer to these values of b; they also indicate respective communication delays of 32, 64, 128, and 256 ms. The generally higher scores for the female talker are apparent. More important, these plots indicate that some of the linguistic cues were affecting the listeners' judgments. The scores for the two-syllable words, zero and seven, are elevated and the scores for five and six are extremely high. The range of scores indicate

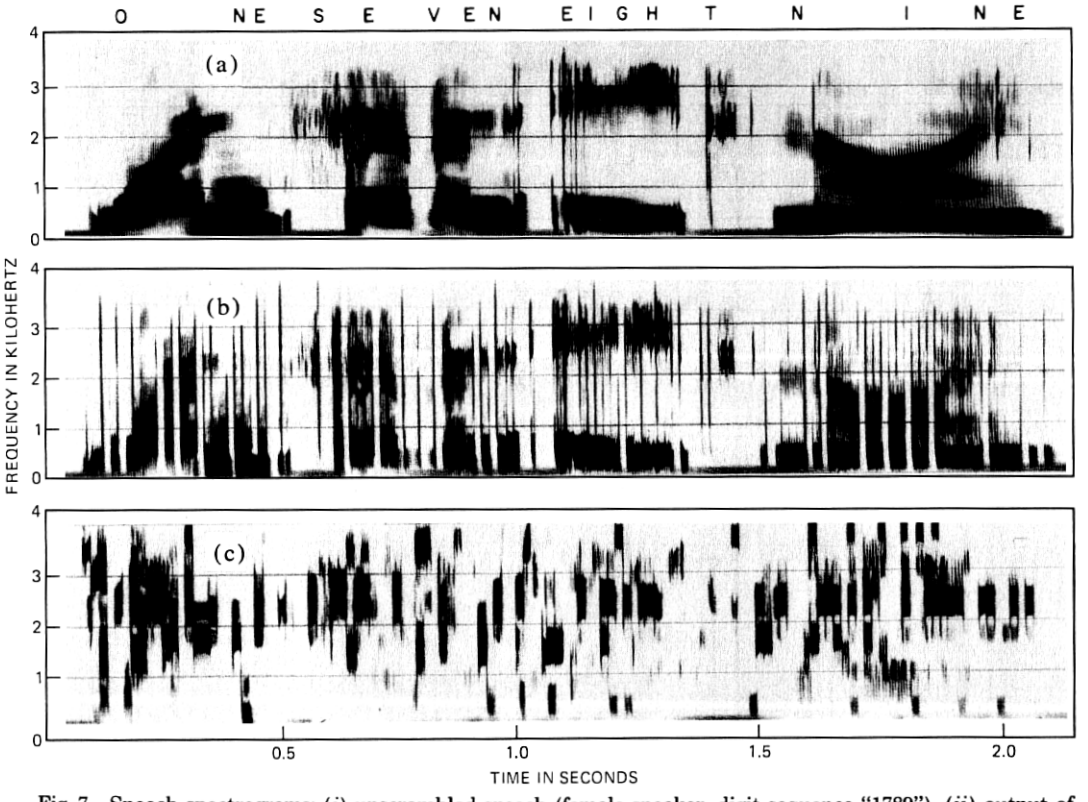

Fig. 7—Speech spectrograms: (*i*) unscrambled speech (female speaker, digit sequence "1789"), (*ii*) output of sequential TSP scrambler with communication delay of 128 ms, and (*iii*) output of sequential TFSP scrambler with the same communication delay of 128 ms.

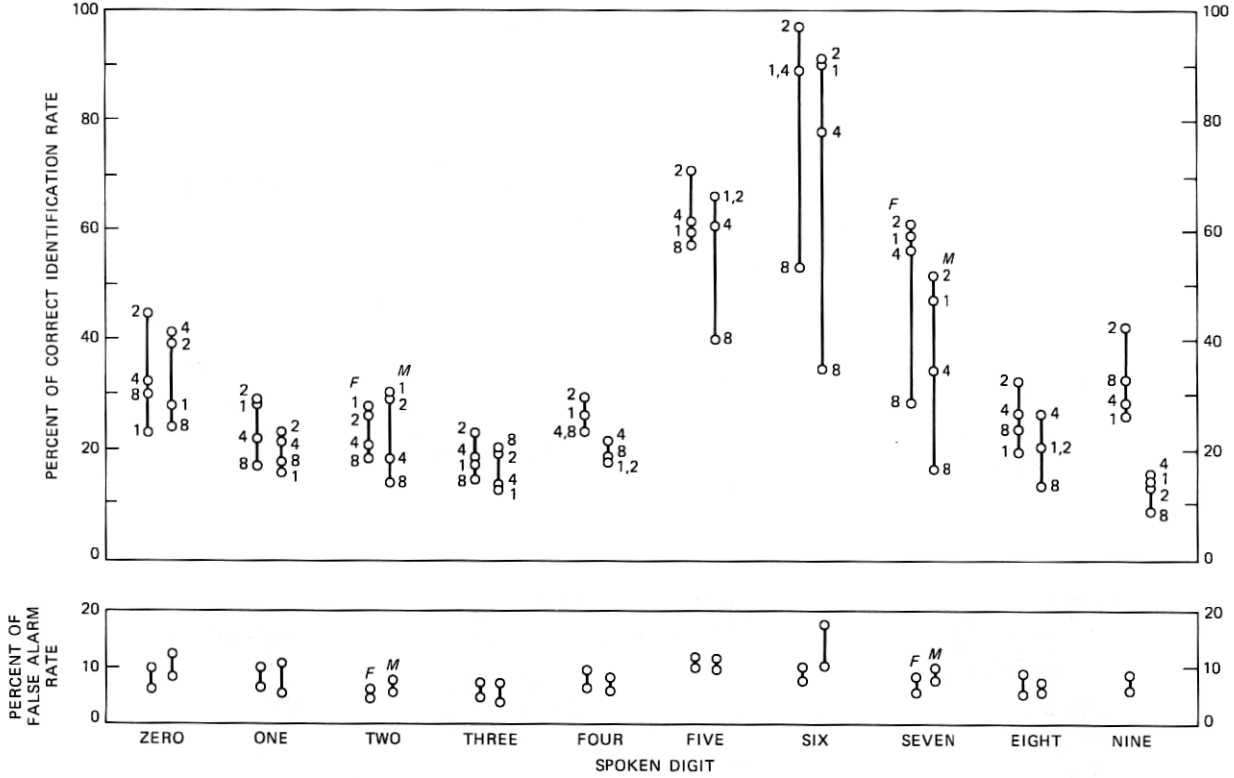

Fig. 8—Correct identification and false alarm rate scores as a function of spoken digit. For each digit, there are two sets of scores, one for female speaker (left vertical bar) and one for male speaker (right vertical bar). These are distinguished by labels *F* and *M* for the digit examples TWO and SEVEN. Each set of scores has four values, corresponding to four values of total communication delay. These four values are identified by numbers that represent multiples of 32 ms. Numbers 1, 2, 4, and 8 therefore indicate delays of 32, 64, 128, and 256 ms.

that the reduced intelligibility with a delay of 256 ms is largely the result of disproportionately lowering the identification of "five," "six," and "seven."

The lower plot in Fig. 8 shows the false alarm rate, i.e., the percent of time that a digit was incorrectly identified as the one labelled in the figure. For simplicity, these plots show only the maximum and minimum values of false alarm rates, as a function of b, rather than values for all four values of (b), as in the upper plot of the figure. If subjects were really guessing among the ten possible digits when they were uncertain, these values should be about 10 percent. The somewhat higher false alarm rate for the digit six indicates that it was probably being used as a default response by some of the subjects.

The observations of this section reinforce our earlier stand that the intelligibility of connected speech where the words are less predictable may, indeed, be much lower than the scores shown for the spoken digits in Fig. 5.

In Fig. 8, the condition of 256-ms delay (labeled 8 in the figure) represents the least intelligibility in only 14 out of 20 conditions shown, and the condition of 32-ms delay (labeled 1 in the figure) represents the greatest intelligibility in only 3 out of 20 conditions. A good example of monotonic behavior is the digit "two." Good examples of non-monotonic behavior are the female utterances of "zero" and "nine." In the latter two examples, the intelligibility difference between delays of 32 and 64 ms (points labeled 1 and 2) were in fact tested to be statistically significant. We do not have an adequate explanation of why an increase of delay from 32 to 64 ms (and in some cases, to 128 and 256 ms) should actually *increase* residual intelligibility, a feature that is counter to the general trends of the average scores in Fig. 5. (Recall that in these averages, the differences between scores at 32, 64, and 128 ms were stated to be statistically insignificant). The phenomenon of significant intelligibility increases owing to delay seems however to be peculiar to digits with sustained sounds such as the *o* in zero and the *n* in nine. In these cases, certain increases of communication delay, or equivalently, certain increases of staying time in memory, may increase the probability that at least one of many fragments of the long sustained sound gets outputted in an "interference-free" context such as an interdigit silence, causing an increase of intelligibility. On the average, however, separation of intra-digit fragments *increases* as a function of delay, and this causes a *decrease* of intelligibility. This was indeed noted in the average scores of Fig. 5.

REFERENCES

1. N. S. Jayant, B. J. McDermott, S. W. Christensen, and A. M. S. Quinn, "A Comparison of Four Methods for Analog Speech Privacy," IEEE Trans. Commun., *COM-29* (January 1981), pp. 18–23.

2. R. C. French, "Speech scrambling and synchronization," Philips Res. Rep., No. 9 (1973), pp. 1–115.
3. E. R. Brunner, "Efficient scrambling techniques for speech signals," Proc. Int. Conf. Commun., Seattle, WA, June 1980, pp. 16.1.1–6.
4. S. Udalov, "Microprocessor-based techniques for narrow-band scrambling," in Proc. Int. Conf. Commun., Seattle, WA, June 1980, pp. 16.4.1–5.
5. R. V. Cox and J. M. Tribolet, "Analog Voice Privacy Systems Using TFSP Scrambling: Full Duplex and Half Duplex," B.S.T.J., this issue.
6. S. T. Hong and W. Kuebler, "An Analysis of Time Segment Permutation Methods in Analog Voice Privacy Systems", Proc. 1981 Carnahan Conference on Crime Countermeasures, Univ. of Kentucky, Lexington, KY, May 1981.
7. G. A. Miller, G. A. Heise, and W. Lichten, "The intelligibility of speech as a function of the context of test materials," J. Exp. Psychol., *41*, (1951), pp. 329–35.
8. B. Blesser, "Speech perception under conditions of spectral transformation: I. Phonetic characteristics," J. Speech and Hearing, *15*, (1972), pp. 5–41.

APPENDIX

Let the probability of "forced-exit" (i.e., the probability that a segment spends the maximum allowable time in scrambler memory) be P. The probability that a segment exits as a result of random selection is therefore $1 - P$. With a buffer size b, and a uniform pdf for random segment selections in buffer positions 1 to b, the probability of unforced exit from any given buffer stage is therefore $(1 - P)/b$, and the corresponding probability of non-exit is $P_n = [1 - (1 - P)/b]$. With the $t = 2b$ design, the probability of non-exit for all possible ages s, $s = 1, 2, \cdots, 2b$ of a given segment is $(P_n)^{2b}$. By definition, this should equal the forced-exit probability P. Therefore,

$$P = P_n^{2b} = \left(1 - \frac{1 - P}{b}\right)^{2b}. \tag{8}$$

Taking logarithms and using $\ln(1 - x) \sim -x$ for $x \ll 1$,

$$\ln P = 2(P - 1); \qquad P \sim 0.203$$

for large b. Numerical solution of (8) shows that P is extremely close to the above asymptotic value of 20 percent for values of $b > 4$. The value at $b = 4$ is about 0.210.