Secure Audio Teleconferencing: A Practical Solution

Rafi Heiman
Bellcore

Abstract

Secure audio teleconferencing is a multi-point communication service which uses encryption to prevent eavesdroppers from listening to the speech signals. Its greatest vulnerability is the audio bridge — that component which combines the conferees' speech signals and returns the result to them.

A new secure teleconferencing system is proposed here. It fits the public telephone network by eliminating the need for the conferees to share their secrets with the bridge. It combines a simplified ('instantaneous') bridging technique with secure bridging ideas previously suggested in the literature, overcoming their main practical disadvantages. In particular, it is not restricting the audio signals to be coded by linear PCM, a technique which is wasteful in terms of bit-rate. Rather, it enables the use of conventional μ-law and A-law PCM, as well as vector quantized PCM, thus can be used with a conventional 64kb/s digital channel.

1 Introduction

1.1 The Problem

Consider three or more parties that talk together over the telephone. This is called an audio teleconference and it is established by using special conferencing equipment called bridge. Conferees transmit their speech signals to the bridge. The bridge in turn, detects which signals are active (e.g., contain speech), and returns to each conferee the sum of some of the active signals. This creates the illusion that each conferee hears all others simultaneously, just as if they were all talking in the same room.

A problem arises when the conferees wish to use encryption to guarantee the privacy of their conference. In all commercial systems for secure teleconferencing, either conferees 'trust' the bridge or only a 'simplex' mode is allowed. Trusting the bridge means that conferees let it know the secure keys so that it can decrypt the signals, combine the clear signals, and encrypt the combined signals before returning

R.A. Rueppel (Ed.): Advances in Cryptology - EUROCRYPT '92, LNCS 658, pp. 437-448, 1993.

them to the conferces. This results in the need to own and guard the bridge. In a simplex link all the bridge does is switching: only single and uncombined signal is returned to each conferee. This causes the inability to hear more than one conferee even when two or more conferees simultaneously speak.

To avoid these limitations, a non standard encryption scheme is needed. It is desirable that the bridge would not be able to listen to the conference contents but still perform its combining function. Therefore a special method is needed — a method that reveals enough information to enable bridging, but does not reveal information about what is being said, even if eavesdroppers have access to the bridge.

In Section 1.3 we mention previously suggested solutions and explain their practical disadvantages. In order to better understand these disadvantages, and see how our new solution overcomes them, a word on digital audio processing is due.

1.2 Digital Audio and Bridging

In order to better understand the problem, we briefly review some standard ways of encoding to digital and bridging audio signals. Further details may be found in the textbooks [RS79] and [JN84] and in bridging articles, [PC83], [PC85] and [MDS91].

Digital encoding of analog signals (such as voice) involves two parts. First, the analog (continuous-time and continuous-amplitude) signal is sampled, yielding a discrete-time and continuous-amplitude signal. Second, quantization converts the continuous-amplitude into a discrete-amplitude.

In Pulse Code Modulation (PCM) a so called 2^R-level quantizer is used and each sample is encoded independently to an R-bit number. In Linear PCM the range of possible amplitude levels is divided into segments of equal size, thus the mapping between amplitude levels and their codes is linear. Linear PCM introduces high distortion power relative to the signal power in low input signal levels. Logarithmic PCM techniques (μ-law and A-law) provide a more constant signal-to-noise ratio due to the code being logarithmically dependent on the sample amplitude. It is generally assumed that to achieve the adequate (so called 'toll') quality of logarithmic PCM with 7- or 8-bits per sample, linear PCM would require 11-12 bits per sample (see [JN84] section 5.3.2 and [RS79] section 5.3.2). Consequently, the logarithmic PCM techniques fit into 56kbit/sec or 64kbit/sec rates, became standards (μ-255 in North America, A-87.56 in Europe [CCI72]), and are widely used in practice.

A less known PCM technique, but one which further compresses the audio, is vector quantized PCM [MRG85]. In vector quantization a few consecutive samples, constituting a vector, are quantized jointly. The quantization process results in a sequence of numbers (or codes) over a previously selected set of possible quantization levels. When quantizing, each sample vector is replaced by that vector from a pre-established code-book of vectors which best matches it. The index of the code-book vector is then transmitted. At the receiver side each such index (or number) is decoded to its corresponding code-book vector of amplitude levels.

Bridging of audio signals typically consists of three stages: energy detection, selection, and addition. In the first stage some 'long term' statistics of each incoming signal are computed in order to determine whether this signal is active or silent. In the second stage a limited number of the active signals is selected. If there are only very few active signals (say, one or two) they are all selected. If there are too many active signals the bridge selects only few of them. The selection algorithm vary from bridge to bridge and it may depend, for example, on predetermined priorities between the conferees and on the exact time at which each conferee became active. In the third stage, the selected active signals are added, and the bridge returns the sum of the signals to each conferee. If a receiving conferee is active and selected, its signal is excluded from the sum that this conferee receives, avoiding echo effects. Among other computations, these stages involve converting the received streams of codes to linear PCM, summing, and converting back to the specific source coding used.

1.3 Previously Suggested Solutions

A few solutions were previously suggested for the problem of secure audio teleconferencing using an untrusted bridge. The first, by Brickell, Lee and Yacobi [BLY87], suggests a few ways to compute addition in the encrypted domain. That is, given the encryptions of a few numbers, to compute the encryption of the numbers' sum (without being able to compute the clear numbers or their clear sum). This provides a solution to our problem with the following assumptions: (1) Conferees provide the bridge with their activity levels. (2) Incoming signals are synchronized. (3) The operation the bridge has to perform on the speech samples is indeed addition. We view assumption 1 and 2 as not too demanding. Computing activity levels can be added to the encryption/decryption 'black box' conferees have anyway, and the activity information does not carry enough intelligibility to understand what is being said. A way to synchronize the incoming signals is described in Section 4. The third assumption, however, means that speech must be encoded by linear PCM, and this is probably the main barrier for the Brickel et al solution to become practical, as can be understood from the discussion above. A solution that uses e.g. logarithmic PCM, and fit into a 64kb/s bit-rate is thus desired.

A simple solution is to let one of the (trusted) conferees own the bridge. All conferees send their encrypted signals to that conferee, who then combines the decrypted signals and returns the results encrypted. This trivially solves the problem of trusting the bridge. However, owning a private bridge is not always economically attractive. A more efficient use of the bridge would be if it is offered by the public telephone network, thus be a shared resource. Even in large private networks, where there are many users to share the private bridge, the need to guard it carries a significant cost and would rather be avoided.

Two other solutions, in which all or part of the bridging process is done by the conferees were suggested by Steer, Strawczynski, Diffie and Weiner [SSDW88]. In the first, conferees are connected in a chain (as opposed to a star whose center is the bridge). Each conferee adds its speech (if active) to the prefix-sums received from its two neighbors and forwards these extended prefixes. The main barrier for this solution to become practical is probably the doubling effect: In this solution each conferee has two incoming and two outgoing voice signals thus bit-rate is doubled. This solution also introduces delay and noise problems.

The second solution proposed in [SSDW88] partially maintains the bridge as part of the communication network. Conferees send their encrypted signals as well as clear activity indications to the bridge. The bridge returns to each of them two selected signals without adding them. Conferees then decrypt them and add them by themselves. This solution too suffers from the doubling effect: one single link (bridge-to-conferee) is carrying two signals, thus, a special channel is needed. The new ISDN standard does allow two audio channels in a single link, but still, it is desirable to have the secure teleconference using the same bit-rate (and single channel) that the non-secured teleconference is using.

A simplification of the last solution that avoids the doubling effect is the so called 'simplex mode' in which the bridge returns only one signal to each conferee. All conferees get the loudest signal, presumed to be the active speaker's signal, except for the loudest speaker who gets the second loudest conferee's signal. In a typical conference most of the time only one conferee is speaking. This may be even more typical of conferences with a high demand for privacy: In business conferences, as opposed to informal private talks, conferees do not tend to interrupt or talk simultaneously with each other that much, and in many cases a chairman controls the conference. In any case interruption usually consists of a single word in order to get attention, and only when the active speaker becomes silent does the interrupting conferee begin his 'real' talk. However, a solution that does allow third parties to hear and be aware of interruptions is preferred. Human factor experts say that third parties prefer to hear interruptions, even at the cost of slightly decreasing the speech quality during these interruptions, a compromise we will adopt here.

1.4 The New Solution

Our new solution enables conferees to hear more than one speaking conferee at a time, it avoids the doubling effect mentioned above, it keeps the increase of bit-rate very low — one bit per sample, and it minimizes the extent to which conferees are involved in bridging. It can use all source coding techniques which are monotone in a sense defined below, among them are the conventional μ-law and A-law PCM and vector quantized PCM [JN84], [MRG85]. For a 64 kbit/sec channel, 7-bit μ-law or A-law can be used, maintaining adequate ('toll') quality using simple inexpensive terminals.

The new solution is very simple. In essence, it combines the main ideas of [BLY87] with a certain bridging technique, called **max-bridging** or **instantaneous bridging** [PR71]. These tools are described in Section 2. Their proposed combination and the system's security are discussed in Section 3. A possible concrete design of the system is described in Section 4.

2 Tools

2.1 Max-Bridging

A typical bridge, called sum-bridge was described above. A max-bridge differs from a sum-bridge only in the third stage, the addition. Once the max-bridge selects the (limited number of) active signals to be combined, it combines them by computing the maximum rather than addition. Namely, if m active signals were selected to be combined, and their amplitude levels at some time instance are $\{a_i\}_{i=1}^{m}$, then their combination is defined to be that a_i with maximal energy, or equivalently, with maximal absolute value, $|a_i|$. This is in contrast to the ordinary sum-bridge whose output is $b = \sum_{i=1}^{m} a_i$. We call it **sample-by-sample max-bridging**.

If only one conferee is speaking, that is $m = 1$, there is clearly no difference between the outputs of the two bridges. Only at those times when two or more conferees speak simultaneously do the max-bridge and the sum-bridge have different outputs. The interesting (and perhaps surprising) fact is that this difference is hardly noticeable when listening to these two outputs.

Pushing this phenomenon one step further, the bridge may operate on vectors of samples at a time rather on a sample-by-sample basis. It views the input signals as sequences of vectors, where each vector consists of a few consecutive samples. Each vector it outputs is the maximal one of the corresponding m input vectors. The definition of maximal vector can be relative to the energy of the whole vectors, yielding a bridge we call **vector-max-bridge**, or the energy of the central samples of the vectors, yielding a bridge we call **center-max-bridge**.

Preliminary simulations we made for evaluating these three types of max-bridging showed that the difference between signals obtained by max-bridging and signals obtained by sum-bridging is small (relative to the signals themselves). A few colleagues who listened to the different bridges' output found it hard to distinguish between them. The reason is perhaps that the effect of hearing two persons talking simultaneously dominates the distortion caused by max-bridging.

In order to quantize this we introduce a measure **signal-to-difference ratio** of a signal s relative to another signal s', defined as the power of s divided by the power of the difference $s - s'$ of the two signals. The signal-to-difference ratios of the output of sum-bridge relative to the outputs of various max-bridges are given in Figure 1. In the worst case studied, center-max-bridging of 5-length vectors, the signal-to-difference ratio was 9.2 db. All numbers were calculated for a few seconds of bridging two active signals (containing simultaneous speech of two persons).

The advantage of the max-bridge is clear. Given various kinds of digital PCM signals all it has to do is compute the maximum. Consider for example log-PCM. While

type of max-bridge	vector length	signal-to-difference
sample-by-sample		11.6 db
vector-max-bridge	3	11.4 db
	5	10.9 db
center-max-bridge	3	10.3 db
	5	9.2 db

Figure 1: Signal-to-difference ratio for max-bridges

a sum-bridge has to translate the logarithmic codes to linear, sum them, and then translate back to the logarithmic scale, a max-bridge can directly compare the codes (excluding their sign bits). This is so because the coding of an amplitude level in log-PCM consists of a sign bit and of a few more bits whose value (interpreted as a binary number) encodes the absolute value of the given amplitude in a monotone manner. Thus, any source coding technique that associates codes to sampled amplitude levels in a monotone manner requires only max computation in max-bridging.

The max-bridge is also suitable for vector quantized PCM. One only has to make sure that the code-book used for the vector quantization is sorted by increasing energy. This way vectors with higher energy would get higher indices in the code-book, i.e., get higher codes.

2.2 Secure-Sum and Secure-Max Calculations

The basic idea of Brickell, Lee and Yacobi [BLY87] for summing numbers given their encryptions is to use an encryption scheme of an additive nature: To encrypt two n-bit numbers a_1 and a_2, a randomly and uniformly chosen $(n+1)$-bit number r is added to them modulo N where $N = 2^{n+1}$. This yields the ciphers

$$\bar{a}_i = a_i + r \pmod{N}, \quad i = 1, 2.$$

Now, $\bar{a} = \bar{a}_1 + \bar{a}_2 \pmod{N}$ can be computed by an untrusted authority (the bridge), from which $a_1 + a_2 = \bar{a} - 2r \pmod{N}$ can be deciphered (by conferees).

In the context of a sequence of sums that have to be carried, like that of summing sequences of audio samples, pseudo random number generator is used to produce the r-s. All conferees have identical generators (and identical seeds). The generators must be synchronized in the transmitting terminals so that the bridge can sum samples encrypted by the same value of r, and the conferees in turn, can subtract the appropriate value.

The secure sum calculation can be easily modified to calculate the maximum of two numbers securely. Given the two encryptions, $\bar{a}_1$ and $\bar{a}_2$ of a_1 and a_2, their difference,

$$\bar{d} = \bar{a}_1 - \bar{a}_2 = a_1 - a_2 \pmod{N}$$

can be computed by the untrusted authority (the bridge). This allows the maximum to be computed: Since $a_i < 2^n$ for $i = 1, 2$, clearly $a_1 \geq a_2$ if $\bar{d} < 2^n$, and $a_1 < a_2$ if $2^n \leq \bar{d}$. Therefore, the bridge can determine which of the a_i-s is maximal and return the corresponding $\bar{a}_i$. To decrypt,

$$a_i = \bar{a}_i - r \pmod{2^n}$$

is computed. Note that the most significant bit of the returned $\bar{a}_i$ is not needed. Note also that the difference between the two numbers is revealed, not just the knowledge of which is larger. This security weakness, which is inherited by our solution, is considered below.

3 The Proposed System and Its Security

Having these two tools in mind, the solution is rather simple. Conferees encode their speech in any monotone PCM technique. They encrypt their code-streams for secure-max computation using synchronized pseudo random number generators. A way to synchronize the generators is described in the next section. A max-bridge is used to combine the signals by computing the max on the given encrypted sequences.

The long term energy detection, the first stage of bridging, is done by the conferees. They provide the bridge with a clear and low-bit-rate signal that contains this information.

Self-detection of activity is done also for the sake of improving security. Clearly, a non active signal, say a signal with a constant PCM code that encodes zero amplitude, would reveal the pseudo random sequence and thus the active speaker's content. Therefore, a conferee that is not active transmits an idle signal to the bridge in clear (without adding the pseudo random numbers). This way, the pseudo random generator will be used in a one-time-pad manner except when at least two conferees are simultaneously speaking. The leakage of information will be limited to short and infrequent periods of time. A signal that the bridge is able to compute when, say, two conferees are simultaneously speaking, is the difference between the two encrypted signals. Each of its sample-codes (in case of sample-by-sample max-bridge) is the difference of the absolute values of the two incoming signals' sample-codes. Listening to this difference signal in an experiment we made, gave us no idea about what is being said. Speech signals contain much redundancy, though. A more sophisticated attack might exist. However, such attack, if exists, is applicable during simultaneous speech only.

Nevertheless, to reduce the amount of information that is revealed, vector quantized PCM can be used. This is more secure since vector quantized PCM contains less redundancy and the difference between the code-book indices of two vectors seems to be harder to interpret than the difference between absolute values of samples. When standard log-PCM is used center-max-bridge would be more secure. It would allow encrypting in the max-calculation manner only one sample in each vector. Thus the bit-rate of the 'parasitic' signal available to the bridge would be reduced by a factor equal to the vector length. Another way to reduce this bit-rate is by letting the bridge compare only some of the most significant bits of the numbers to be compared. This way fewer bits are encrypted in the max-calculation manner. Increasing the vector length, likewise decreasing the number of bits per sample-code that are involved in the max-calculation improve the security of the system but degrade the audio quality of the signal the bridge returns during simultaneous speech.

All samples in a vector other than the central one are encrypted by other and more secure means due to the fact that the bridge only has to forward them. Keeping in mind that conferees may be switched from vector to vector, conventional encryption methods, e.g. DES [DES77] or RSA [RSA78] can not be used here in a straight forward way without significantly expanding the bit-rates. The reason is that their block-lengths are too big. They can however be used for generating a pseudo random bit (or number) stream to be exclusive-ored (or to scramble in some other way). By the considerations above, different conferees will use different pseudo random bits to encrypt their non-central samples. This however implies that in addition to each vector, the bridge communicates some information to identify the conferee that has encrypted this vector.

On top of this, a system implementing this technique should have a user option for simplex bridging. In this mode each conferee can hear only one other conferee at a time, but no information other than activity indication leaks, provided a 'good' pseudorandom generator is used. Another possible option is to add a second layer of encryption for which the conferees do share the key with the bridge. This will cause the minor leakage of information during interruptions to be revealed to the bridge only but not to others who may eavesdrop to the conferees-to-bridge links.

4 A Concrete Design

In this section we suggest a detailed design for implementing the secure audio teleconference system. This suggestion is made for the sake of concreteness, and some decisions made here might be changed elsewhere, depending on the specific implementation. We assume that audio signals consist of 7-bit PCM at sampling rate of 8000 samples per second and describe a system that fits a 64kbit/sec channel.

An octet is eight consecutive bits, consist of a 7-bit PCM word and an additional bit, used for the side information required by our scheme. A **vector** consists of five consecutive octets. The term vector also refers here to the five PCM words only, excluding the five overhead bits. The five overhead bits are used for framing (**f-bit**), pseudo random bit generator indexing (**i-bit**), extra bit for the secure-max scheme (**BLY-bit**), conferee activity reporting (**a-bit**), and conferee identification (**id-bit**). A **frame** consists of 16 vectors, which are 80 octets, and corresponds to 10 milliseconds.

The 16 f-bits per frame are used for frame synchronization and other channel signaling. The method suggested in [CCI90] for its FAS and BAS signals might be used here.

The bridge switches vector by vector based on comparing the absolute values of the third PCM-word received from each active conferee (center-max-bridge). The third PCM word in each vector is encrypted as in Section 2.2, using eight pseudo random bits. The extra eighth bit introduced is the above called BLY-bit. Prior to encrypting the third PCM word, this word is left-cyclic-shifted so that its sign bit becomes the least significant bit. This is done because the bridge has to return that conferee's sample of maximum absolute value. The other four PCM words are encrypted each by using seven pseudo random bits in a bit-by-bit exclusive-or (Xor) manner.[1]

The eight bits used to encrypt the third (central) PCM word are the same for all conferees. The seven bits used to Xor other PCM words vary from conferee to conferee. DES [DES77] can be used to generate all these pseudo random bits. A single computation of DES generates 64 pseudo-random bits. These are used either to encrypt the third PCM word of eight consecutive vectors or to encrypt other four PCM words in two consecutive vectors (using only 56 pseudo-random bits). Thus, to encrypt a frame, each conferee performs ten DES computations, two of them for the secure-max scheme and eight for the Xor-s. Before the conference takes place, conferees agree upon a secret DES key, k, to be used during the conference.[2]

Since non central samples are encrypted in a way that is conferee dependent, the signal returned from the bridge to conferees must identify the source of each vector. For this reason the number of conferees is bounded. In fact the number of listeners to the conference can be arbitrary, but the number of conferees that are allowed to speak is bounded. We first describe the system assuming only 7 conferees are allowed to speak. The pseudo random bits to encrypt frame number i of conferee j, $1 \leq j \leq 7$ are the output of DES when using the key k to encrypt the numbers $64i$ and $64i + 32$ (for the secure-max scheme) and $64i + j$, $64i + 8 + j$, ... , $64i + 56 + j$ (for the Xor's).

The 16 i-bits of a frame carry the frame number i (modulo 2^{16}). This repeats itself every $2^{16} \cdot 10$ milliseconds, which is over 10 minutes. The bridge needs this information to synchronize incoming signals and conferees need this for decryption. When a new conferee joins the conference, it performs a hand-shake protocol with the bridge, so that they can measure their transmission delays and the conferee can synchronize its pseudo random generator with those of the previously joined conferees. This synchronization should only be very rough, say with uncertainty of fifty milliseconds. To further synchronize the signals incoming to the bridge, and in order to compensate over transmission delays which may vary from conferee to conferee, the bridge buffers the incoming signals. It then can compare central samples encrypted with same

[1] In fact it would be more secure to encrypt this way only the seven bit absolute value of the third PCM word, and to encrypt the sign bit in a one time pad manner, similar to the other four PCM words. This is because the secure sum/max scheme reveals during interruptions the parity of the least significant bits of the two clear numbers, a_1 and a_2, as pointed out by J. Massey [Mas92]. Our concrete design does not follow this line for the sake of the system's simplicity.

[2] This key exchange is done using some other private channel or a public key system.

pseudo random bits. The i-bits in the returned signal allow conferees to synchronize the pseudo random bits for the decryption.

Every four of the 16 activity bits in a frame contain a 4-bit activity level of a conferee. Activity level is computed every four vectors (2.5 milliseconds) by averaging the signal's energy during that time. If the activity level represents a too low energy average, this conferee does not transmit its PCM samples but rather, an idle signal along with valid overhead bits. This allows the bridge to maintain synchronization with the conferee's signal, but prevents gaining information about the pseudo random bits used to encrypt the central samples, as discussed in Section 3.

When the bridge returns a vector, say generated by conferee number j, $1 \leq j \leq 7$, it also communicates the value j to identify the specific DES bits that should be used to decrypt the non-central samples of the vector. If no conferee is active the bridge returns idle vector (or white noise) and the value j=0. There are three bits per vector that are used to carry the value j. These are the id-bit, the a-bit, recalling that the a-bit is used only in the conferee-to-bridge link, and the BLY-bit, recalling that the most significant bit is not needed for the secure-max decryption. For this reason we assumed that at most seven conferees are allowed to speak in the conference.

A possible way to increase this number is by letting additional conferees use only 6 bits per sample, 'robbing' their seventh bit of each PCM sample for the additional conferee id-numbers, and by reassigning the pseudo random numbers generated by the DES machine. A better possibility is to dynamically assign the seven conferees that are allowed to speak and to maintain a list of the identities of these seven conferees. A conferee which is not in the list but starts talking and being bridged replaces one of the seven conferees that occupy the list at that moment. The bridge does it by sending the corresponding update of the list to all conferees. 'Robbing' a few bits, e.g., 16 consecutive i-bits for such update is enough.

5 Open Problems

- A system for secure bridging is proposed. Its major advantages are that the bridge need not be trusted, that it allows conventional PCM source coding and standard channels, and still it is a duplex system — more than one conferee can be heard at a time. During simultaneous speech a negligible degradation of the combined signal quality occurs as well as a minor leakage of information. Is there any attack on this system that we are not aware of? Known attacks on speech ciphers seem to be irrelevant here (cf. [CM86], [CR87] and [GDS91]).

- The [BLY87] solution which we adopt here enables the bridge to compare two numbers given their encryptions, while revealing their difference and requiring an extra bit. Is there a method for comparing encrypted numbers without one or both of these disadvantages?

- A technical need we have in our system is that for encryption scheme of small block size, say 8–32 bits, which is secure under known message attack of a small number, say 7, of messages. Commonly used encryption schemes that encrypt multiple messages by same key have block size of 64 bits or more. One-time-pad is secure for any block size, even of one bit. Can there be anything in between?

- A problem along the line of this paper that comes to mind is to develop a method to securely bridge signals encoded by more sophisticated source coding techniques. Examples are ADPCM, a method for which standards for 3 kHz audio over 32kbit/sec channel and for 7kHz audio over 64kbit/sec channel exist, LPC, and vector-quantized LPC.

Acknowledgements

I thank Milton Anderson, Alex Gelman, Sharad Singhal and Yacov Yacobi for many discussions on this subject, all of which were crucial for me.

References

[BLY87] E.F. Brickell, P.J. Lee, and Y. Yacobi. *Secure audio teleconference*, volume 293 of *Lecture notes in computer science: Advances in cryptology - CRYPTO '87*, pages 418-426. Springer-Verlag, 1987.

[CCI72] International Telegraph and Telephone Consultative Committee. *Pulse Code Modulation (PCM) of voice frequencies.* CCITT recommendation G.711, Geneva, 1972.

[CCI90] International Telegraph and Telephone Consultative Committee. *Frame structure for a 64 to 1920 kbit/s channel in audiovisual systems*, 1990. CCITT recommendation H.221.

[CM86] J.M. Carrol and S. Martin. The automated cryptanalysis of substitution ciphers. *Cryptologia*, 10:193-209, 1986.

[CR87] J.M. Carrol and L.E. Robbins. The automated cryptanalysis of polyalphabetic ciphers. *Cryptologia*, 11:193-205, 1987.

[DES77] The National Bureau of Standards. *Data Encryption Standard*, January 1977. U.S. department of Commerce, FIPS pub. 46.

[GDS91] B. Goldburg, E. Dawson, and S. Sridharan. The automated cryptanalysis of analog speech scramblers. In *Advances in Cryptology, Eurocrypt*, pages 422-430, 1991.

[JN84] N.S. Jayant and P. Noll. *Digital coding of waveforms: principles and applications to speech and video.* Prentice-Hall Inc., Englewood Cliffs, N.J., 1984.

[Mas92] J.L. Massey. *private communication.*

[MDS91] Electrospace Systems Inc. *MDS-1 video conferencing bridge*, manuscript. Mail stop 4000, p.o.box 831359, Richardson, Texas 75083-1359, 1991.

[MRG85] J. Makhoul, S. Roucos, and H. Gish. Vector quantization in speech coding. *Proc. of the IEEE*, 73(11), November 1985.

[PC83] M. V. Pitke and T. Chandrasekaran. An improved conference circuit. *Proc. of IEEE*, 71(12):1460–1461, 1983.

[PC85] M. V. Pitke and T. Chandrasekaran. A digital conferencing technique. *Proc. of IEEE*, 73(11):1687–1688, 1985.

[PR71] S. G. Pitroda and B. J. Rekiere. A digital conference circuit for an instant speaker algorithm. *IEEE transactions on communication technology*, com-19(6):1069–1076, December 1971.

[RS79] L.R. Rabiner and R.W. Schafer. *Digital processing of speech signals.* Prentice-Hall, 1979.

[RSA78] R. Rivest, A. Shamir, and L. Adelman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[SM87] A. Shimizu and S. Miyaguchi. Fast data encryption algorithm (FEAL). In *Advances in Cryptology, Eurocrypt*, pages 267–267, 1987.

[SSDW88] D.G. Steer, L. Strawczynski, W. Diffie, and M. Wiener. *A secure audio teleconference system*, volume 403 of *Lecture notes in computer science: Advances in cryptology - CRYPTO '88*, pages 520–528. Springer-Verlag, 1988.