END-TO-END CHANNEL ASSURANCE FOR COMMUNICATION OVER OPEN VOICE CHANNELS

David J. Coumou[*], *Gaurav Sharma*[*†]
[*] Electrical and Computer Engineering Department, University of Rochester, Rochester, NY 14627
[†]Dept. of Biostatistics and Computational Biology, University of Rochester Medical Center, Rochester, NY 14642
Email: DavidCoumou@ieee.org, gaurav.sharma@rochester.edu

ABSTRACT

Voice channels are commonly utilized in a wide variety of communication and control scenarios that span the gamut of applications ranging from military and commercial aviation to emergency response applications. Often the existing equipment for these applications utilizes openly accessible voice communication systems that offer no assurance of data security or integrity. It is possible to secure the systems with encryption but this would require significant new investments and introduce interoperability problems with existing equipment. As an alternative, we propose an in-band methodology for end-to-end channel assurance in these scenarios based on speech watermarking. Our speech watermark is resilient to low-bit rate coding standards commonly used in these voice communication applications and in mobile telephony and digital voice-over-IP (VoIP).

1. INTRODUCTION

To highlight a scenario of an open voice communication system we present air traffic control (ATC) air-ground voice communications. In this particular application, the voice channels are a "party line" between the air traffic controller and the aircraft in the respective flight sector, illustrated in Figure 1 with two aircraft and a single air traffic controller. Current systems do not explicitly address authentication of the conversing parties. Though a functional protocol, the system is prone to security breaches and potential safety mishaps due to the absence of source and data authentication. It is paramount for safety that the addresser and addressee realize their stake in the process of communicating voice data and information. The security problem can be illustrated with a spurious party intercepting communication and emulating reception of aircraft or the ATC controller, ultimately compromising security and safety. The safety concern can be illustrated further with call-sign confusion. In call-sign confusion, the controller mis-identifies the call-sign of the pilot; a possible result of the controller incorrectly deciphering the call-sign when the pilot initially addresses the ATC controller. The commands that ensue from the controller may be mistakenly directed to an alternate aircraft.

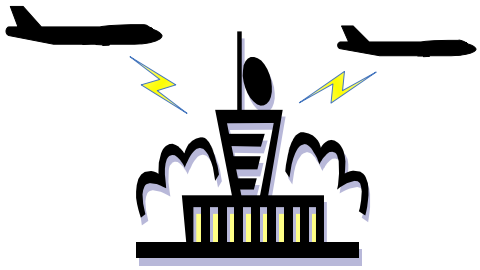

Figure 1: Air Traffic Control Communication Scenario

This commercial ATC example is analogous to military ATC. Military aviation employs a network-centric scheme to support landing ashore and all phases of flight in the shipboard environment. The military aviation scheme provides facilities for covert and secure operation with broad interoperability for services, allies, and civil airspaces. The secure portion includes waveforms and encryption schemes that can provide security and integrity, though at the cost of a closed architecture which often entails higher cost.

Incorporating a layer of authentication can alleviate these types of safety and security issues in open systems. We propose a speech watermarking based method for providing end-to-end channel assurance for voice communications. By embedding information in a innocuous and perceptually undetectable manner in the speech signal, watermarking provides a means of authentication of the speech signal source. Since the information embedded into the speech signal is imperceptible, covert message passing may also be achieved with digital watermarking. Since low-bit rate compression methods are commonly used for voice communication systems, either currently or as a part of a planned enhancement, it is also desirable that such systems be resilient to such coding.

Low bit rate speech coding methods [1] utilize a source representation for the speech signal, decomposing it into a quasi-periodic excitation signal (pitch) and vocal tract filter (formant). In order to ensure survivability in these coders, watermark information should be embedded in these signal characteristics rather than in the raw

1-4244-1513-06/07/$25.00 ©2007 IEEE

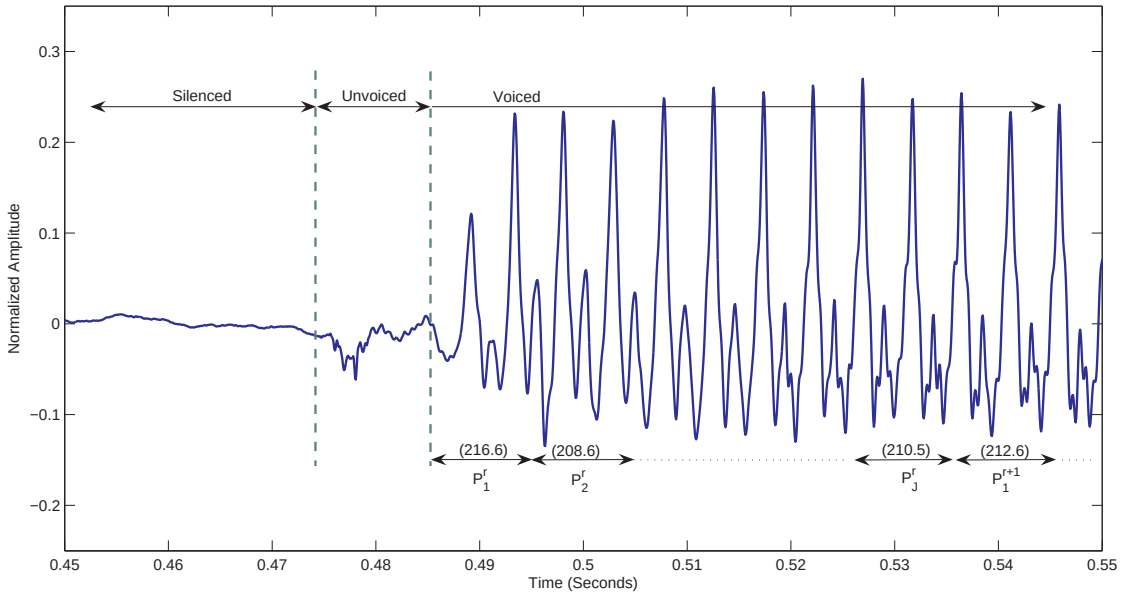

Figure 2: Silenced, Unvoiced, and Voiced regions identified in a segment of a speech signal. In the voiced regions, the corresponding estimated fudamental periods are indicated (•) in Hz.

waveform. For our speech watermarking implementation, we chose to embed the watermark in small imperceptible perturbations of the pitch [2][3][4]. Note that watermarking through modifications of the linear-predictive filter coefficients representative of the vocal tract has also been proposed [5], albeit for scenarios where the original speech signal is required for detection.

In Section 2, we describe our speech watermarking system. We proceed to describe in the subsequent section how a multilayered authentication scheme can employ our speech watermarking system. In Section 4, we give an overview of a concatenation coding scheme that is an integral part of our speech watermarking system and the multilayered authentication scheme. The penultimate section contains simulation results demonstrating the robustness of the speech watermarking system and the resiliency of the multilayered authentication scheme to key tampering. The last section contains a summary of this speech watermarking application.

2. SPEECH DATA EMBEDDING BY PITCH MODIFICATION

A speech signal segment is shown in Figure 2, where the abscissa (representing time) axis has been partitioned into non-overlapping sections corresponding to regions of silence, unvoiced speech, and voiced speech [6]. Data embedding is accomplished by altering the fundamental period of voiced regions containing at least M contiguous pitch estimates P. A bit t is embedded into block r by modifying the average pitch estimated from J pitch estimates ($J \leq M$). Data embedding is accomplished by applying binary scalar quantization index modulation (QIM) [7] to the average pitch for each block. Binary QIM is even/odd modulation and is illustrated in Figure 3. For a given block average pitch P^r, the continuous value is discretized to a bin determined by the bit value $t \in (0,1)$ and the quantization step size Δ. For instance, if $\Delta(\alpha - \frac{1}{2}) \leq P^r < \Delta(\alpha + \frac{1}{2})$ and t:=0, then pitch estimates $P^r_{\kappa \in (1,J)}$ are modified such that $P^r = \alpha\Delta$. Data is recovered at the receiving end by estimating the quantization bin in which the corresponding average pitch values lie.

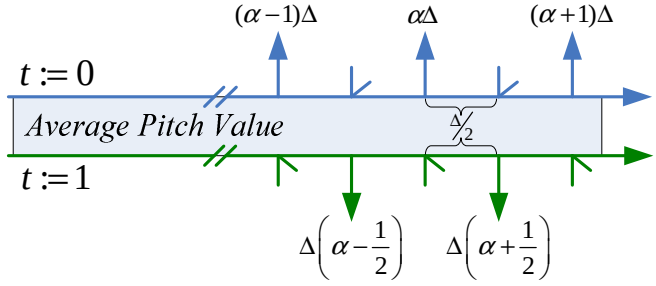

Figure 3: Scalar quantization index modulation (QIM)

Since pitch is an element of the speech signal preserved by most speech vocoders [8][9], the pitch modification based embedding is more resilient to the distortion impinged on the watermarked speech signal by low-bit rate compression than other embedding methods (e.g., spread-spectrum). One challenge, however, for the data embedding by pitch modification is that estimates of voiced segments at the receiver may differ from those at the embedder either due to the process of embedding or due to distortions introduced in the channel [2][3][4].

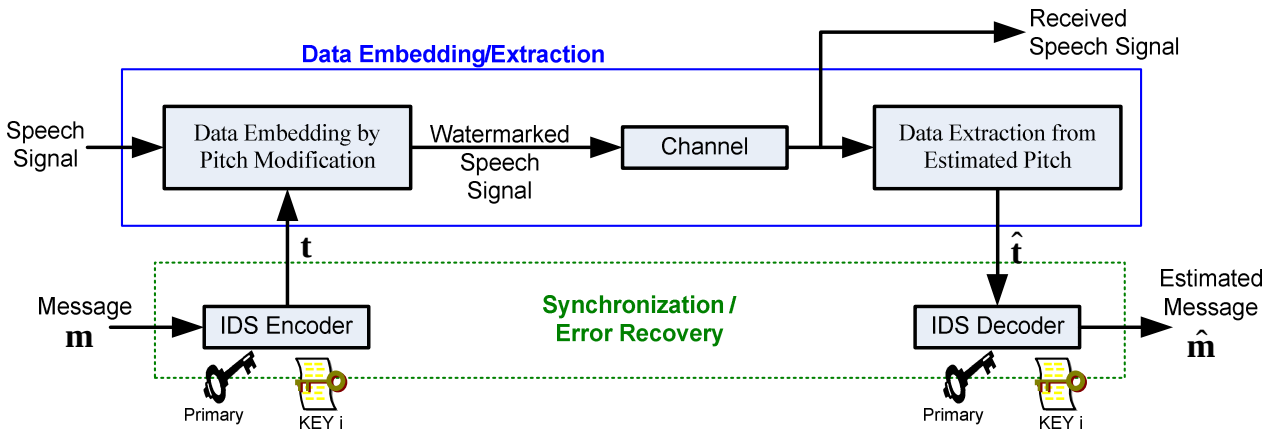

Figure 4: Channel assurance using key dependent elements for pitch data embedding with synchronization codes

For example, multiple voiced segments at the embedder may coalesce into a single voiced segment because small unvoiced regions detected at the transmitter between two voiced sections go undetected during the pitch estimation procedure. Similarly, relatively small voiced segments may be detected at one end and not the other. In general, these types of mis-matches result in insertion and deletion errors in the estimates of the embedded data. Insertion/deletion events are particularly insidious since they cause a loss of synchronization and cannot be corrected using conventional error correction codes. For this reason, we combine the pitch modification embedding with codes capable of correcting insertion, deletion, and substitution errors in order to facilitate synchronization at the watermark receiver. This results in an overall system as shown in Figure 4, where the IDS encoding and decoding processes are introduced as pre and post-processing steps at the transmitter and receiver respectively, (the keys shown in this figure are used for authentication as we describe subsequently in Section 4 and may be disregarded for the time being). We note that in general the capability to gracefully handle differences in feature estimates between the transmitter and receiver is a desirable property for feature based watermarking methods [4][10].

3. IDS CODES

The IDS codes facilitate synchronization at the speech watermark receiver, and as we shall subsequently describe, also form an integral component of our multilayered authentication scheme. Our speech watermarking scheme [3] utilizes Davey and MacKay's concatenated insertion, deletion, and substitution (IDS) codes [11][12]. We provide a brief summary of their operating principle, referring the reader to [4] for a more complete technical discussion.

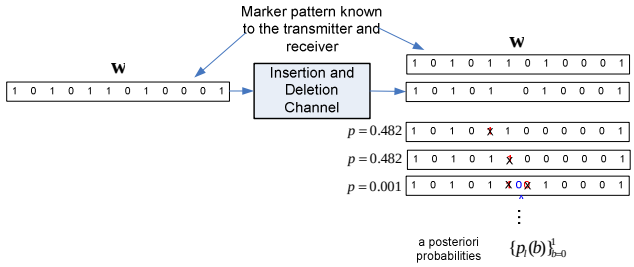

Figure 5: A posteriori probability computation for IDS error events using a marker pattern known at the transmitter and receiver

A key element of the IDS codes is a fixed *n* bit marker vector **w** that is known to both sender and receiver. As a start, consider the scenario where the marker vector represents the data embedded in the speech signal through the pitch modification process. Note that no information is communicated in this simplistic scenario since the marker pattern is known *a priori* at both ends. As illustrated in Figure 5 consider the result of communicating this pattern over our pitch-modification based speech channel, which as we indicated earlier introduces IDS errors. Using the received data with the known marker pattern along with a suitable channel model[1], one may compute a posteriori probabilities for the different possible IDS events that explain the received data. Figure 5 depicts three specific error events along with plausible values for probabilities that may be inferred for these. These probabilities illustrate a couple of aspects that are pertinent to synchronization. Note that for the specific data in the figure, the probabilities do not favor a single event; instead the bulk of the probability is split among two possible error events (this behavior "generalizes" to the observation that any deletion in a run of 1's or 0's cannot be localized within the run). Thus the synchronization cannot be

[1] In practice we use a hidden Markov model (HMM) for the IDS channel for this purpose [11][12].

exactly inferred at the receiver. However, the probabilities also illustrate that the uncertainty is localized among a (typically small) number of IDS error events (2 in the above example). This uncertainty in synchronization can be resolved through the use of additional error correction coding, which also enables data communication, as we illustrate next.

The recovery of synchronization with some uncertainty using a known marker pattern is also feasible when communicating data if the data is "piggy-backed" on to the marker pattern as a small fraction of deliberate substitutions. For this purpose, data is mapped through a look-up-table (LUT) that maps groupings of k bits to unique binary strings with length n greater than k, where **w** sparse refers to the property that a large majority of the bits within each n bit string are zero. These n bit strings from the LUT mapping is then added modulo 2 to the marker vector thereby embedding the k bits as deliberate substitutions in **w**.

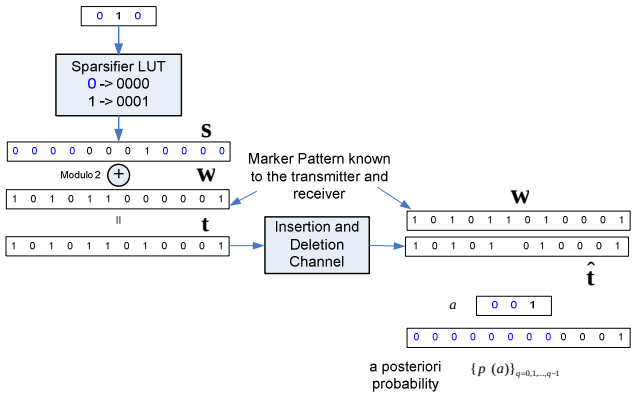

Figure 6: Data communication using a sparse LUT

Figure 6 illustrates this system. Since the string **s** is sparse, once again using the known marker vector and channel model, synchronization may be recovered with some uncertainty as before. In order to resolve this uncertainty and also recover the embedded data, a q-ary low density parity check (LDPC) code [11][12] is employed to provide the necessary redundancy. Each group of k information bits constitutes a q-ary symbol (where $q=2^k$). K symbols are encoded into a codeword of N ($N>K$) q-ary symbols which are then embedded in the marker vector via a sparsifier as indicated earlier. Using the channel model, posteriori probabilities for each of the q possible values for the symbol may be computed shown schematically at the bottom of Figure 6. These posterior probabilities form the soft inputs for the LDPC decoding process at the receiver.

The overall system is shown in Figure 7. The combination of the sparse LUT and the q-ary LDPC code constitutes a concatenated coding system that allows recovery from IDS errors with the LDPC forming the outer code and the sparsifier the inner (non-linear) code. The message data **m** is a block of K q-ary symbols and is encoded by a q-ary LDPC encoder that comprises the outer IDS encoder. The LDPC encoder generates dense q-ary codeword **d** with length N from a systematic generator matrix **G**. The generator matrix is specified by a sparse $(N-K)\times N$ parity check matrix **H** with entries selected from GF($q=2^k$), i.e. the Galois field of q elements. LDPC codeword **d** is converted to a binary sequence by mapping each q-ary element of **d** to a sparse binary string which is added element by element to the marker vector **w** to form the data **t** that is then embedded in the speech signal by modifying the mean pitch of an embedding block.

At the receiving end through the watermark extraction process, a string of bits $\hat{\mathbf{t}}$ is obtained as the estimate of the bits **t** embedded at the transmitter. Using $\hat{\mathbf{t}}$, posterior probabilities for q-ary codeword symbols are computed using an efficient forward-backward algorithm [11][12][4]. The procedure computes a symbol-by-symbol likelihood probabilities $P_j(a)=P(\hat{\mathbf{t}}\,|\,d_j=a,\hbar)$ for $1\le j\le N$, where $\hbar$ = ($\hbar'$,**w**) and $\hbar'$ constitutes the channel model parameters of the probability of insertion P_I, deletion P_D, transmission P_T, and substitution P_S.

These likelihood probabilities are utilized by the outer LDPC decoder. The LDPC decoder is a probabilistic iterative decoder that uses the sum-product algorithm [13] to estimate marginal posterior probabilities $P\left(d_j\,|\,\hat{\mathbf{t}},\mathbf{H}\right)$ for the codeword symbols $\left\{d_j\right\}_{j=1}^{N}$. Each iteration uses message passing on a graph for the code to update estimates of these probabilities. At the end of each iteration, tentative values for these symbols are computed by picking the q-ary value x_j for which the marginal probability estimate $P\left(d_j\,|\,\hat{\mathbf{t}},\mathbf{H}\right)$ is maximum. If the vector of estimated symbols $\mathbf{x}=[x_1,\ldots x_N]$ satisfies the LDPC parity check condition **Hx=0**, the decoding terminates and the message **m** is determined as the last K symbols of **x**. If the maximum number of iterations is exceeded without a valid parity check, a decoder failure occurs.

4. MULTI-LAYER CHANNEL ASSURANCE

The in-band communication enabled by our speech watermarking system has three random elements- the marker vector **w** for the inner code, sparse LUT mapping (which may be permuted or otherwise modified), and the parity check matrix **H** for the outer LDPC code. Through key-dependent generation of these elements we can enable

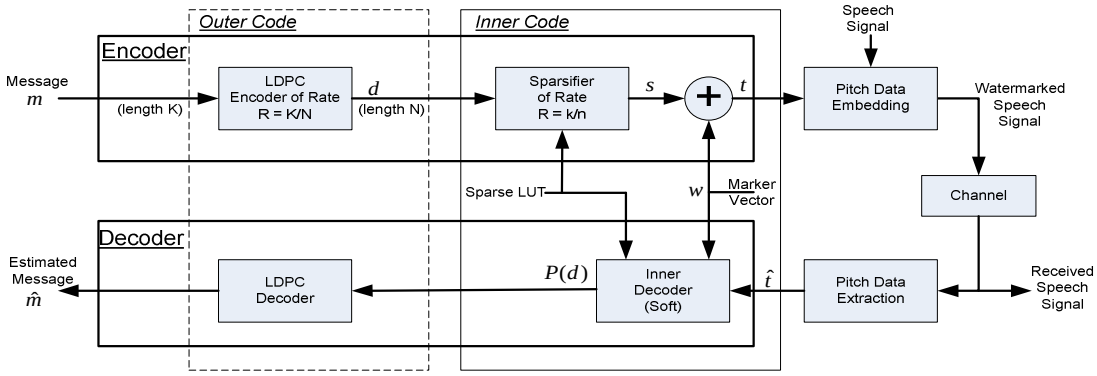

Figure 7: Pitch-based speech watermark with synchronization

authentication of the end-to-end voice channel. In addition to these two elements, the information communicated through the system offers the possibility of an additional level of hierarchical authentication[2]. By virtue of the design of our system, this authentication methodology is resilient to low bit-rate encoding of the speech that may commonly be encountered in open voice channels.

Specifically, we assume that the two communicating entities are time-synchronized and generate the (time-varying) pseudo-random marker vector **w** using a shared cryptographic key, the valid recovery of watermark information at the receiver (as evidenced by success in the LDPC decoding) provides an assurance to either end that the respective communicating source is indeed authentic. Conversely, if the receiving end is unable to recover the watermark, suspicion can be raised regarding the validity of the transmission source. Successful watermark recovery is also dependent on the parity check matrix and by incorporating it provides an additional measure of channel assurance. Note that time synchronization with time-varying generation of these elements is necessary to defeat replay attacks [14].

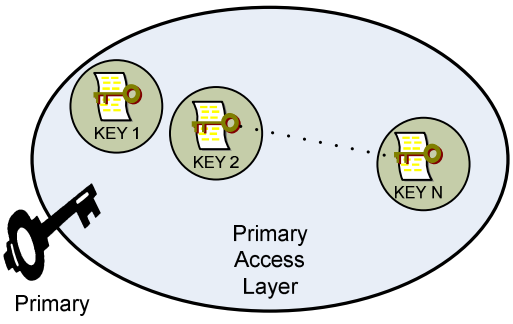

Figure 8: Multilayer authentication scheme

Various implementations can exploit the tandem arrangement of the marker vector, the sparse LUT, and parity check matrix for channel assurance. For instance, the cryptographic key, used to generate the marker vector, and the parity check matrix can be individually assigned to the various entities participating in the multi-party voice channel. An alternative configuration is shown in Figure 8. The marker vector is generated from a primary key that is shared with multiple entities in the voice communication scheme. The parity check matrix, devised as the second layer of authentication, is distributed as individual keys to each respective entity in the multi-party voice channel. Finally encryption/digital signatures for the embedded data can form yet another level in this authentication hierarchy.

5. EXPERIMENTAL RESULTS

The speech watermarking system was implemented using the PRAAT toolbox [15] for embedding and extracting data by pitch modification and MATLAB® for the inner and outer decoding processes. Sample speech files from audio books and various internet sources [16][17] and a database provided by the NSA for speech compression were utilized for testing [18]. The parity check matrix **H** was formulated for a coding rate of ¼ and the rows of the matrix were assigned q-ary symbol values from heuristically optimized sets [19]. The corresponding generator matrix for systematic encoding was obtained using Gaussian elimination. The marker vector **w** was generated using a pseudo-random number generator. The channel parameters were found by performing a sample pitch based embedding and extraction that was manually aligned to determine the number of IDS events.

Speech Watermarking Simulation

Monte Carlo (MC) simulation was performed using random message vectors of q:=16-ary message symbols. These were arranged in blocks of K:=25 and encoded as LDPC code vectors of length N:=100. The length of the sparse vectors was chosen as n:=10. The binary data

[2] By making the embedded data dependent on the signal being communicated, we can enable authentication of the speech signal in addition to providing channel assurance.

obtained from the sparsifier was embedded into the speech signal by QIM of the average pitch from J:=5 pitch estimates using a quantization step Δ that ranged between 6-15 Hz.

The channel was variously chosen as:

a) No compression;
b) GSM-06.10 at 13 and 17 kbps [8]; and
c) AMR (Adaptive Multi-Rate) at 5.1 kbps [9].

Low bit-rate coders b) and c) are standards for 2G and 3GPP cellular networks respectively.

An additive white Gaussian noise (AWGN) channel was also included with the MC simulation. Performance was based on the percentage of simulation runs for which the embedded data was successfully recovered.

Figure 9 illustrates the impact of varying the QIM quantizer step-size Δ for the three compression channels. In general an increase in the QIM step size also increases the embedding distortion. The embedding distortion is almost imperceptible for QIM step sizes of 15 Hz or less [4]. For a quantizer value of Δ:=15 Hz, data was successfully recovered over 95% of the simulations for all channels.

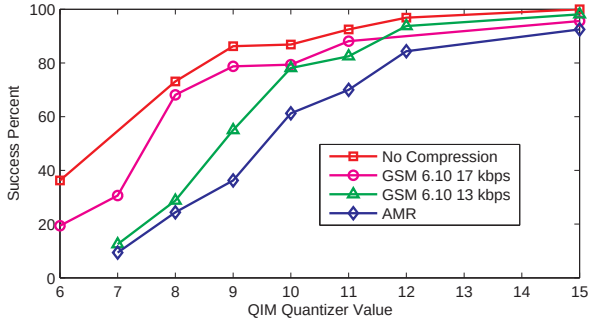

Figure 9: Speech watermark performance over low-bit rate compression channels

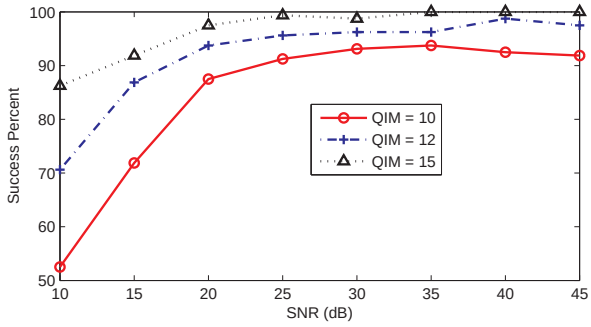

Figure 10: Speech watermark performance over an AWGN channel

Results for the AWGN channel for Δ of 10, 12 and 15 Hz are shown in Figure 10 where the abscissa indicates the AWGN signal-to-noise power ratio (SNR). Informally we determined an AWGN SNR below 27 dB produced a palpable distortion and at 20 dB resulted in objectionable audio quality. From Figure 10, we demonstrate a negligible degradation from the no compression result in Figure 9.

Channel Assurance

To evaluate the capability of our speech watermarking system to offer channel assurance, we conducted experiments that simulate attempts to infiltrate the primary authentication barrier. For our experiments, we assume the key used to generate the random marker vector **w** at the transmitting end of the open voice channel is unavailable to the attacker and the LDPC parity check matrix **H** and the sparse LUT are known. The attacker attempts to determine the watermark vector by arbitrarily generating marker vectors to recover the watermark information from the string of bits $\hat{\mathbf{t}}$. If the correct marker vector is produced, the attacker may then utilize this to mimic the authenticated channel for the corresponding entity.

Our simulation parameters used the same inner and outer encoder/decoder configuration parameters described in the previous subsection with the quantization step Δ:=15 Hz. After achieving a successful decoding using the correct marker vector and LDPC parity check matrix **H**, we subsequently generated 13000 binary random marker vectors and proceeded with the soft inner decoder followed by the outer LDPC decoder. Despite prior knowledge of **H**, all 13000 random vectors did not decode the watermark message **m** after 600 LDPC decoding iterations. With q:=16, the average symbol error was 94 (N:=100).

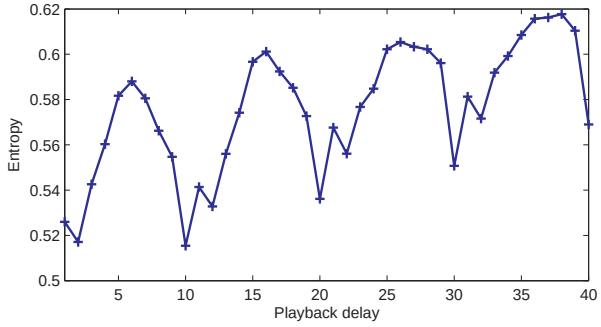

Figure 11: Inner decoder output entropy as a function of playback delay

In a second attack, we consider the scenario where the attacker attempts to replay the bit string $\hat{\mathbf{t}}$ extracted from a communicating entity, albeit with the one bit delay that is necessary. Once again the system fails to authenticate. As a further exploration of this attack we attempt to determine the entropy of the posterior probabilities computed for the inner decoder as a function of the play back delay in bits. We calculate entropy as $N^{-1}\sum_{j}^{N}\sum_{a=0}^{q-1}P_j(a)\log_q\left(P_j(a)\right)$.

The entropy is shown as a function of the playback delay in Figure 11. As seen in the figure the decoder fails. Also we can observe a dip in the entropy for the delays corresponding to a sparse symbol mapping (this does not however circumvent authentication).

5. CONCLUSION

We presented a speech watermarking method for end-to-end channel assurance for open voice communication channels. Our system is resilient to low bit rate coding. This is accomplished by using pitch based embedding in conjunction with IDS codes for synchronization. We demonstrate that by using key based generation of the random elements of the watermark code, end-to-end channel assurance may be guaranteed with high confidence.

6. REFERENCES

[1] K. Sayood; *Introduction to Data Compression*; 3rd Ed.; Morgan Kaufmann, San Francisco, CA; 2005

[2] M. Celik, G. Sharma, and A. M. Tekalp, "Pitch and duration modification for speech watermarking," *Proc. IEEE Intl. Conf. Acoustics Speech and Sig. Proc.*, Mar. 2005, pp. II, 17–20.

[3] D. Coumou and G. Sharma; Watermark synchronization for feature-based embedding: application to speech; *IEEE International Conference on Multimedia and Expo*; Toronto, Canada, July 9-12, 2006, pp. 849-852.

[4] D. J. Coumou and G. Sharma, "Insertion, Deletion codes with feature-based embedding: A new paradigm for watermark synchronization with applications to speech watermarking," Submitted to *IEEE Trans. on Information Forensics and Security*, 2006.

[5] A. Gurijala, J.R. Deller, Jr., M.S. Seadle, J.H.L. Hansen; "Speech watermarking through parametric modeling"; *Proc. of International Conference on Spoken Language Processing*; Denver, Sep. 2002. (CD-ROM)

[6] L. R. Rabiner and R. W. Schafer; *Digital Processing of Speech Signals*; Prentice Hall, Englewood Cliffs, NJ, 1978.

[7] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Info. Theory*, Vol. 47, No. 4, May 2001, pp. 1423–1443.

[8] 3GPP TS6.10: "Full Rate Speech Transcoding", http://www.3gpp.org/ftp/Specs/archive/06_series/06.10/

[9] 3GPP TS26.071: "AMR speech Codec; General description", www.3gpp.org/ftp/Specs/archive/26_series/26.071

[10] G. Sharma and D. J. Coumou; "Watermark synchronization: Perspectives and a new paradigm," in *Proc. 40th Annual Conf. on Info. Sciences and Systems (CISS)*; Princeton, NJ, Mar. 2006, pp. 1182-1187 (invited paper).

[11] M. C. Davey and D. J. C. MacKay; "Reliable communication over channels with insertions, deletions, and substitutions"; *IEEE Trans. Info. Theory*; pp. 687-698, Feb. 2001.

[12] M. C. Davey; *Error Correction using Low Density Parity-Check Codes*; Ph.D. thesis, University of Cambridge, Cambridge, UK, December, 1999.

[13] D. J. C. MacKay. Good error correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, Vol. 45, No. 2, pp. 399-431, March, 1999.

[14] A. Menezes, P. van Oorchot, S. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL; CRC, 1997.

[15] P. Boersma and D. Weenik, "Praat: doing phonetics by computer"; [Online]. Available: http://www.fon.hum.uva.nl/praat

[16] Ohio State University Speech Corpus; Available online: http://buckeyecorpus.osu.edu

[17] Open Speech Repository; Available online: http://www.voiptroubleshooter.com/open_speech

[18] http://www.uninett.no/voip/codec.html

[19] D. J. C. MacKay, "Optimizing Sparse Graph Codes over GF(q)", http://www.cs.toronto.edu/~mackay/gfqoptimize.pdf