2011 IEEE Symposium on Security and Privacy

Phonotactic Reconstruction of Encrypted VoIP Conversations: Hookt on fon-iks

Andrew M. White* Austin R. Matthews*† Kevin Z. Snow* Fabian Monrose*
*Department of Computer Science †Department of Linguistics
University of North Carolina at Chapel Hill
Chapel Hill, North Carolina
{amw, kzsnow, fabian}@cs.unc.edu, armatthe@email.unc.edu

***Abstract*—In this work, we unveil new privacy threats against Voice-over-IP (VoIP) communications. Although prior work has shown that the interaction of variable bit-rate codecs and length-preserving stream ciphers leaks information, we show that the threat is more serious than previously thought. In particular, we derive approximate transcripts of encrypted VoIP conversations by segmenting an observed packet stream into subsequences representing individual phonemes and classifying those subsequences by the phonemes they encode. Drawing on insights from the computational linguistics and speech recognition communities, we apply novel techniques for unmasking parts of the conversation. We believe our ability to do so underscores the importance of designing secure (yet efficient) ways to protect the confidentiality of VoIP conversations.**

I. Introduction

Over the past decade, Voice-over-IP (VoIP) telephony has witnessed spectacular growth. Today, VoIP is being used everywhere, and is making steady headway as a replacement for traditional telephony in both the residential and commercial sectors. The popularity of free online services such as Skype, Fring, and Google Talk is a case in point. Indeed, several analysts predict that VoIP will remain the fastest growing industry over the next decade, and some forecast that the subscriber base will top 225 million by 2013.[1] Yet, even with this widespread adoption, the security and privacy implications of VoIP are still not well understood. In fact, even with the attention VoIP security (or lack thereof) has received in the past, the concerns have mostly centered on the lack of authenticity in the call setup phases of the signal and session negotiation protocol(s) or susceptibility to denial of service attacks [33]. Regarding the confidentiality of the data streams themselves, the prevailing wisdom is that, due to the open nature of traffic traveling over the Internet, VoIP packets should be encrypted before transmission.

However, current practices for encrypting VoIP packets have been shown to be insufficient for ensuring privacy. In particular, two common design decisions made in VoIP protocols—namely, the use of variable-bit-rate (VBR) codecs for speech encoding and length-preserving stream ciphers for encryption—interact to leak substantial information about a given conversation. Specifically, researchers have shown that this interaction allows one to determine the language spoken in the conversation [55], the identity of the speakers [2, 41], or even the presence of *known* phrases within the call [56].

Rightfully so, critics have argued that the aforementioned threats do not represent a significant breach of privacy. For example, the language of the conversation might easily be determined using only the endpoints of the call—a call from Mexico to Spain will almost certainly be in Spanish. While the identification of target phrases is more damning, it still requires the attacker to know (in advance) what she is looking for within the stream. In this work, we make no such assumption about *a priori* knowledge of target phrases. Rather, our ultimate goal is to reconstruct a hypothesized transcript of the conversation from the bottom up: our approach segments the observed sequence of packets into subsequences corresponding to individual phonemes (i.e., the basic units of speech). Each subsequence is then classified as belonging to a specific phoneme label, after which we apply speech and language models to help construct a phonetic transcription of parts of the conversation. To assess the quality of our reconstruction, we apply widely accepted translation scoring metrics that are designed to produce quality scores at the sentence level that correlate well with those assigned by human judges.

The approach we take has parallels to how infants find words in a speech stream. As Blanchard et al. [8] point out, adults effortlessly break up conversational speech into words without ever realizing that there are no pauses between words in a sentence. This feat is possible because we have a lexicon of familiar words that we can use to segment the utterance. Infants have no such luxury. Instead, they must use perceptual, social, and linguistic cues to segment the stream of sounds. Amazingly, the linguistic cues come from learned language-specific constraints (or *phonotactics*) that determine whether a word is well-formed or not; infants use this knowledge of well-formedness to help segment speech.

The fascinating problem here is that infants must learn these rudimentary, language-specific, constraints while si-

[1] See, for example, Infonetics Research's *VoIP and UC Services and Subscribers Report* at http://www.infonetics.com.

1081-6011/11 $26.00 © 2011 IEEE
DOI 10.1109/SP.2011.34

multaneously segmenting words. They use familiar words (e.g., their own names) to identify new words which are subsequently added to their small vocabulary. Interestingly, the Linguistic and Psychological Sciences literature abounds with studies (e.g., [9, 23]) which show that, as early as six months of age, infants use knowledge of which basic phonemes occur together, as well as learned knowledge of within-word versus between-word sounds, to segment perceived utterances into words. As we show later, we apply a similar methodology when tackling the problem of reconstructing words from strings of phonemes.

II. Background Information

Before proceeding further, we first present some necessary background that is helpful in understanding the remainder of the paper. The background material covers basic notions in linguistics, pertinent VoIP details, and information about the datasets we use throughout the paper.

Phonetic Models of Speech

The ideas in this paper rely heavily on insights from modern theories of phonology. In particular, we draw from a vast body of work on phonetics—i.e., the study of linguistic sounds. From a computational perspective, phonetics involves studying how sound is produced by the articulators of the vocal tract and how they are realized acoustically [30]. In phonetics, the pronunciations of words are modeled as strings of symbols representing individual speech units called *phones*. While several alphabets exist for representing phones (e.g., ARPAbet for American English), the de facto standard is the International Phonetic Alphabet (IPA).

For the remainder of the paper, what is particularly important is that each phone is based on articulatory processes, and that phones are divided into two main classes: consonants and vowels. Both kinds of sounds are formed by the motion of air through the mouth, throat and nose. Consonants, for example, are made by restricting airflow in some way, and can be both *voiced* (meaning they involve vibrations of the vocal cords) or *unvoiced*. By contrast, vowels usually involve less obstruction of air flow, and are louder and longer lasting than consonants. Moreover, because all consonants are sounds made by restricting airflow, they can be distinguished from each other by where the restriction is made (the *place* of articulation) as well as how the restriction is made (the *manner* of articulation). In English, for example, the "hissing" sound of [*f*] in 'fish' is made by pressing the lower lip against the upper teeth. There are several major manners (e.g., stops, nasals, and fricatives) used to distinguish consonants.

Likewise, vowels can also be characterized by articulatory processes (see Figure 1), the most important of which are vowel *height* (i.e., roughly the height of the highest part of the tongue), *backness* (i.e., roughly indicating where the tip of the tongue is relative to the vocal track), and *roundness*

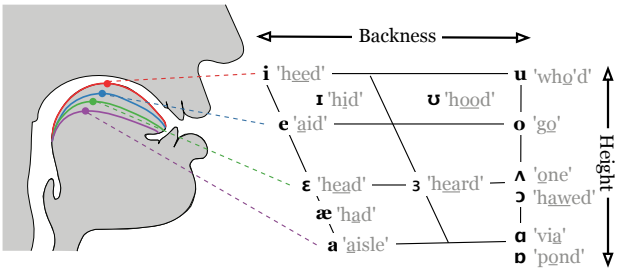


Figure 1. Vowels in American English (IPA format), differentiated by their *height* and *backness*. Left: the relative tongue positions.

(i.e., whether the shape of the lips is rounded or not). For example, compare how your mouth feels as you say 'beat' and 'boot'. If you hold the vowels in these two words, you should be able to feel a difference in the *backness* of your tongue. Similarly, if you compare the words 'beat' and 'bat', you should feel your chin moving up and down; this is a difference in *height*. To feel a difference in *rounding*, compare the words 'put' and 'pool'. As you say 'pool' you should feel your lips pucker into a round shape; in 'put', your lips should be loose.

Consonants and vowels are combined to make syllables, which are governed by the *phonotactics* of the language — that is, language-specific conditions that determine whether a word is well-formed or not. At a high level, phonotactics are constraints on which phones can follow which, i.e., rules that govern how phones may be combined to form well-formed words. In English, for example, there are strong constraints on what kinds of consonants can appear together: [st] (as in 'stop') is a very common consonant cluster, but some consonant sequences, like [zdr] (as in 'eavesdrop'), are not legal word-initial sequences in English.[2]

Lastly, in linguistics and speech processing, an abstraction called a *phoneme* (typically written between slashes) is used to represent similar phones with a single symbol. For example, the phoneme /t/ can be pronounced as any of three phones in English; which of these three phones is uttered depends on the position within a syllable: /t/ is pronounced as [t^h] at the beginning of a syllable (as in 'top'=[t^h op']), [t] in the middle of a syllable (as in 'stop'=[stɒp']), and [t'] at the end of a syllable (as in 'pot' = [p^h ot']). Phones belonging to the same phoneme are called *allophones*: [t^h], [t], and [t'] are allophones of the phoneme /t/.

In Section V, we leverage such linguistic insights to build a string matching technique based on phonetic edit distance. In addition, we use phonotactics of English (e.g., what sequences of phonemes or allophones are allowable within words) to assist with phoneme classification.

[2]Of course, [zdr] may exist word-initially in other languages, such as in the Bulgarian word [zdraf], which means 'health'.

Voice over IP

In VoIP, voice data and control messages are typically transmitted through separate channels. The control channel generally operates using an application-layer protocol, such as the Extensible Messaging and Presence Protocol (XMPP) used by Google Talk or the Session Initiation Protocol (SIP). The voice channel typically consists of a Real-time Transport Protocol (RTP) stream transmitted over UDP. We concern ourselves only with the voice channel in this work.

Typically, the audio for VoIP conversations is encoded using an audio codec designed specifically for speech, such as Skype's SILK, the Enhanced Full Rate (EFR) codec specified by the GSM standard, or the open-source Speex used in many VoIP applications (including Google Talk). Speech codecs differ from general audio codecs since human speech can be represented much more efficiently than general audio due to the periodic nature of certain speech signals and the relatively limited number of potential sounds. For speech, sound is usually sampled at between 8 and 32 kHz (i.e., between 8,000 and 32,000 samples are recorded per second). This sample stream is then segmented into *frames*, or blocks, of a certain duration and each frame is compressed by the speech codec for transmission. The duration is a fixed value generally between 10 and 100ms; a typical value, and the one used in this work, is 20ms, which corresponds to 320 samples per frame when sampling at 16kHz.

Many modern speech codecs are based on variants of a well-known speech coding scheme known as *code-excited linear prediction* (CELP) [49], which is in turn based on the *source-filter* model of speech prediction. The source-filter model separates the audio into two signals: the *excitation* or source signal, as produced by the vocal cords, and the *shape* or filter signal, which models the shaping of the sound performed by the vocal tract. This allows for differentiation of phonemes; for instance, vowels have a periodic excitation signal while fricatives (such as the [sh] and [f] sounds) have an excitation signal similar to white noise [53].

In basic CELP, the excitation signal is modeled as an entry from a fixed codebook (hence *code*-excited). In some CELP variants, such as Speex's VBR mode, the codewords can be chosen from different codebooks depending on the complexity of the input frame; each codebook contains entries of a different size. The filter signal is modeled using *linear prediction*, i.e., as a so-called adaptive codebook where the codebook entries are linear combinations of past excitation signals. The "best" entries from each codebook are chosen by searching the space of possible codewords in order to "perceptually" optimize the output signal in a process known as *analysis-by-synthesis* [53]. Thus an encoded frame consists of a fixed codebook entry and gain (coefficient) for the excitation signal and the linear prediction coefficients for the filter signal.

Lastly, many VoIP providers (including Skype) use VBR codecs to minimize bandwidth usage while maintaining call quality. Under VBR, the size of the codebook entry, and thus the size of the encoded frame, can vary based on the complexity of the input frame. The specification for Secure RTP (SRTP) [3] does not alter the size of the original payload; thus encoded frame sizes are preserved across the cryptographic layer. The size of the encrypted packet therefore reflects properties of the input signal; it is *exactly* this correlation that our approach leverages to model phonemes as sequences of lengths of encrypted packets.

III. High-level Overview of Our Approach

The approach we pursue in this paper leverages the correlation between voiced sounds and the size of encrypted packets observed over the wire. Specifically, we show that one can segment a sequence of packet sizes into subsequences corresponding to individual phonemes and then classify these subsequences by the specific phonemes they represent. We then show that one can segment such a phonetic transcript on word boundaries to recover subsequences of phonemes corresponding to individual words and map those subsequences to words, thereby providing a hypothesized transcript of the conversation.

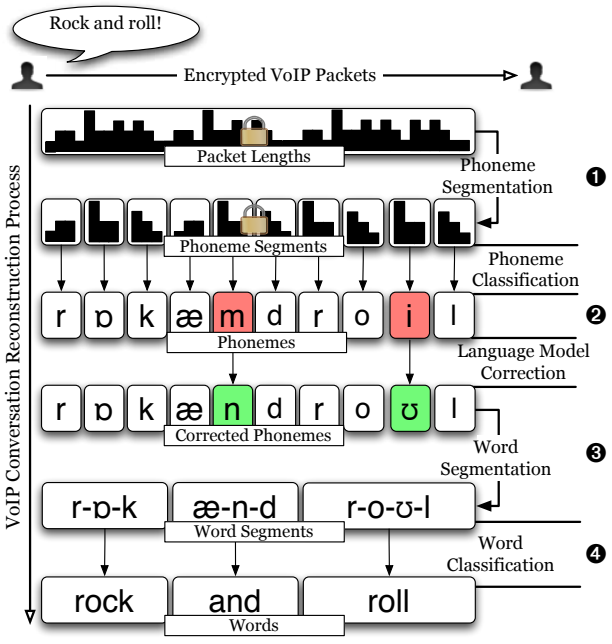


Figure 2. Overall architecture of our approach for reconstructing transcripts of VoIP conversations from sequences of encrypted packet sizes.

Our work draws from advances in several areas of computational science. A simplified view of our overall process is shown in Figure 2. As an example, we use the phrase 'rock and roll', the dictionary pronunciation for which is represented as [ɹɒk ænd ɹoʊl] in IPA. Our basic strategy is as follows. First, we use a *maximum entropy* model (Stage ❶)

to segment the sequence of packet sizes into subsequences corresponding to individual phonemes. We then apply (Stage ❷) a combination of maximum entropy and *profile hidden Markov* models to classify each subsequence of packet sizes according to the phoneme the subsequence represents, resulting in an approximate phonetic transcript of the spoken audio. In our example, this transcript is [ɹɒkæmdɹoil].

The hypothesized transcript is improved by applying a trigram language model over phonemes (and phoneme types) which captures contextual information, such as likely phoneme subsequences, and corrects the transcript to represent the most likely sequence of phonemes given both the classification results and the language model. In our example, this results in [ɹɒkændɹoʊl]. Notice the unlikely phonetic sequence [æmd] has been replaced with the far more likely [ænd]. Next, we segment (Stage ❸) the resulting transcript into subsequences of phonemes corresponding to individual words using a phonetic constraint model, resulting in the more recognizable string [ɹɒk ænd ɹoʊl].

Finally, we match each subsequence to the appropriate English word using a phonetic edit distance metric (Stage ❹), giving us the desired 'rock and roll'. In the general case, a trigram language model over words (and parts-of-speech) is then applied to the resulting transcript to correct tense and disambiguate between homophones (i.e., words which sound alike) by finding the most likely sequence of words given both the hypothesized transcript and the language model.

Data and Adversarial Assumptions

The *TIMIT Acoustic-Phonetic Continuous Speech Corpus* [21], a collection of recorded speech with time-aligned word and phonetic transcripts (allowing us to label segments by phoneme), provides the audio samples used in our experiments. The TIMIT corpus is comprised of 6,300 speech recordings from 630 speakers representing eight major dialects of American English. Each speaker reads ten predetermined, phonetically-rich sentences, such as 'Alimony harms a divorced man's wealth', 'The drunkard is a social outcast', and 'She had your dark suit in greasy wash water all year'. The transcripts contain labels for 58 distinct phoneme-level[3] sounds. Following the standard approach used in the speech recognition community, we folded the original TIMIT classes into 45 labels [36] by combining some allophones and combining closures and silences. ARPAbet, the phonetic alphabet on which the labeling systems of TIMIT is based, does not map directly to the articulatory features in Section II; therefore, we convert the phoneme sequences to their IPA representations for the latter stages of our evaluation. In order to generate sequences of encoded frame lengths from the (16kHz, single-channel) audio samples, we encode each sample using the reference version of the Speex encoder, instrumented to output the sizes of the encoded frames, in wideband (i.e., 16kHz) VBR mode. The phonetic labels from the time-aligned transcripts are then used to identify subsequences corresponding to individual phonemes for training; this encoding process gives us a number of sequences for each phoneme.

We note that the approach we take assumes that the adversary has access to (i) the sequence of packet lengths for an encrypted VoIP call (ii) knowledge of the language spoken in the call, (iii) representative example sequences (or models derived therefrom) for each phoneme, and (iv) a phonetic dictionary. The first assumption can be readily met through any number of means, including the use of a simple packet sniffer. Knowledge of the language of interest can be gleaned using the ideas in [32, 55] or by simple endpoint analysis. Lastly, obtaining representative example sequences for each phoneme is fairly straightforward: one can use prerecorded, phonetically-labeled audio files as input to a speech codec to produce the examples. In fact, using labeled examples from prerecorded audio is exactly the approach we take in this paper in order to model phonemes. Note that our primary goal is to build speaker-independent models and thus we do not require speaker-specific audio. Finally, phonetic dictionaries (e.g., `CELEX`, `CMUdict` and `PRONLEX`) are readily available; we use data from TIMIT and from the `PRONLEX` dictionary (containing pronunciations from over 90,000 words) as our phonetic dictionary.

IV. Related Work

Traffic analysis of encrypted network communications has a long and rich history. Much of that work, however, is focused on identifying the application protocol responsible for a particular connection (e.g., [7, 12, 17, 31, 42, 43, 54]). It was not until recently that researchers [10, 38, 48, 50, 52] began exploring techniques for inferring sensitive information within encrypted streams using only those features that remain intact after encryption—namely packet sizes and timing information. Song et al. [50], for example, used the inter-arrival time between packets to infer keystrokes in SSH sessions; Sun et al. [52] and Liberatore and Levine [38] showed that identification of web sites over encrypted HTTP connections (e.g., SSL) is possible using the sizes of the HTML objects returned by HTTP requests; Saponas et al. [48] showed how to identify the movie being watched over an encrypted connection.

More pertinent to this paper, however, is the work of Wright et al. [55, 56] that showed that encrypted VoIP calls are vulnerable to traffic analysis wherein it may be possible to infer the spoken language of the call or even the presence of certain phrases. In the latter case, the approach of Wright et al. assumes that the objective is to search an encrypted packet stream for subsequences matching a target phrase or word, such as 'attack at dawn', and therefore requires that a probabilistic model of likely corresponding packet

[3]In addition to phonemes, the corpus contain some labels for sounds, such as pauses and recording errors, unrelated to human speech.

length sequences (i.e., representing the target phrase in its entirety) be generated *in advance*. As discussed earlier, no such *a priori* information is necessary under our approach: we construct transcripts from the bottom up rather than matching phrases from the top down.

Several other approaches for exploring information leakage in encrypted VoIP calls (working under different environmental assumptions than Wright et al.) have also been proposed. For example, if silence suppression is assumed (i.e., packet transmission is suppressed when a party is silent), researchers posit that the duration of talk spurts for words spoken in isolation makes identification of specific "speeches" [37, 41] possible. In a recent study with 20 speakers, Backes et al. [2] show that speaker-specific pause patterns might be sufficient to undermine the anonymity of speakers in encrypted VoIP calls. That said, it is well accepted in the speech community that continuous speech (i.e., everyday communication) lacks identifiable pauses between words [11]. In fact, speakers generally talk faster (and typically shorten or run sentences together) as speech becomes more natural and colloquial. This observation is even more important in our context where there are no within-word pauses. Hence, we make no assumptions about voice activation detection and/or silence suppression.

Lastly, Dupasquier et al. [15] investigate the extent of information leakage from Skype voice traffic. The authors conclude that the general concept we pursue here "seems quite difficult" because classification of phonemes is too challenging. Thus, they revert to the prior setting of knowing the target phrase in advance and use dynamic time warping to validate the work of Wright *et al.* A focus of this paper is showing that such statements were premature, and that phoneme-level reconstruction can be successful in undermining the privacy of encrypted VoIP conversations.

For conciseness, the relevant literature on speech and language models will be presented elsewhere in this paper.

V. Overall Methodology

We now turn our attention to explaining the details behind the key ideas explored in this paper. Wherever possible, we provide the intuition that drives our design decisions.

A. Finding Phoneme Boundaries (Stage ❶)

Given the sequence of packet sizes from a VoIP conversation, the first challenge is to identify which of these packets represent a portion of speech containing a boundary between phonemes. While automatic segmentation of speech waveforms on phonetic boundaries has received much attention in the speech recognition community, in our context we have no access to the acoustic information and must operate on the sequence of packet sizes. However, recall that many speech codecs, and Speex in particular, are based on CELP (code-excited linear prediction), which encodes speech with two different signals: the excitation signal and the filter signal. As mentioned earlier, the filter signal for a given frame is modeled as a linear combination of past excitation signals. Thus more information must be encoded for frames in which the sound changes drastically—such as at the transition between two phonemes. Similarly, less information is encoded for intra-phoneme frames, where the sound changes relatively little. Figure 3 illustrates how changes in frame size can indicate a phonetic boundary.

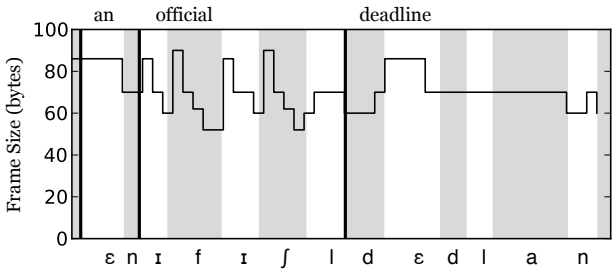


Figure 3. Frame size sequence for the first few words of an utterance of 'an official deadline cannot be postponed', illustrating how the sizes of frames differ in response to phoneme transitions. Notice the distinct changes (e.g., a sharp rise) in frame sizes near some phoneme boundaries (e.g., [ɪ], [f], and [ʃ] in 'official'). Near other phoneme boundaries (e.g., [d], [l], and [a] in 'deadline'), however, frame size remains constant.

Methodology

To perform the segmentation, we apply a probabilistic learning framework known as *maximum entropy modeling*[4] [6, 28] that simultaneously captures many contextual features in the sequence of frames, as well as the history of classifications in the sequence, to decide which frames represent phoneme boundaries. Such models have been successfully applied to problems like part-of-speech tagging [46] and text segmentation [5].

Maximum entropy modeling estimates the posterior probability $p(y|x)$, where x is an observation and y a label. In order to do so, one calculates the empirical distribution $\tilde{p}(x, y)$ from training data as the relative frequency of examples with value x and label y. One then defines binary indicator functions, $f(x, y)$, to describe features of the data relevant to classification.

In the case of phonetic boundary segmentation, we represent a given frame with w. The labels, i.e., *boundary* or *interior frame*, are represented by the binary variable v. An indicator function $f(w, v)$ then describes a feature of the frame which is relevant to whether that frame represents a phoneme boundary, for example:

$$f(w, v) = \begin{cases} 1, & \text{if } v \text{ is } \textit{boundary} \text{ and } w \text{ has size } n, \\ 0, & \text{otherwise.} \end{cases}$$

[4]Also known as multinomial logistic regression.

Given an indicator function, one can compute the expected value of a feature, f, with respect to the training data as:

$$\tilde{p}(f) = \sum_{x,y} \tilde{p}(x,y) f(x,y)$$

One can thus represent any statistical phenomena in the training data with $\tilde{p}(f)$. The expected value of f with respect to the target model, $p(y|x)$, may be represented as:

$$p(f) = \sum_{x,y} \tilde{p}(x) p(y|x) f(x,y)$$

Requiring that $\tilde{p}(f) = p(f)$ imposes the constraint that the model agree with the training data with respect to feature f; over all features, this yields a set of constraints for the target model:

$$C = \left\{ p \in P \mid p(f_i) = \tilde{p}(f_i) \text{ for } i \in \{1, 2, \cdots, n\} \right\}$$

Many models may satisfy the set of constraints. However, the principle of maximum entropy states that the model that best represents the data given the current state of knowledge is the one with the most entropy. This yields a constrained optimization problem of the form $argmax_{p \in C} H(p)$, where $H(p)$ is the entropy of y conditioned on x in the model p.

Phoneme Segmentation Feature Templates	
1	size of frame w_i (i.e., the current frame size)
2	size of frame w_{i-1} (i.e., the previous frame size)
3	size of frame w_{i+1} (i.e., the next frame size)
4	bigram of sizes for frames w_{i-1}, w_i
5	bigram of sizes for frames w_i, w_{i+1}
6	trigram of sizes for frames w_{i-1}, w_i, w_{i+1}
7	sequence of frame sizes since the last hypothesized boundary
8	number of frames since since the last hypothesized boundary

Table I
FEATURE TEMPLATES FOR THE PHONETIC SEGMENTATION, WHERE w_i REPRESENTS THE iTH FRAME.

For boundary identification, we define several feature *templates* which specify features that we hypothesize correlate with phoneme boundaries. The templates we use are given in Table I, and some features are illustrated in Figure 4 for clarity. Although each frame only gives us one observable feature (namely, the size of the frame), we leverage the surrounding frames, and the history of previously classified frames, in the sequence to create a much richer feature set. The templates are used to automatically generate the full feature set directly from the data.

As per our hypothesis regarding the interaction between linear prediction and frame size, notice that feature templates 1-6 capture the frame sizes in the proximity of the current frame. The frame size unigrams, bigrams, and trigrams must be explicitly captured because maximum entropy models do not model feature interactions, i.e., they only consider individual features in isolation. Some phonemes may always exhibit a certain frame size sequence; we capture this behavior with feature template 7. Lastly, because some boundaries

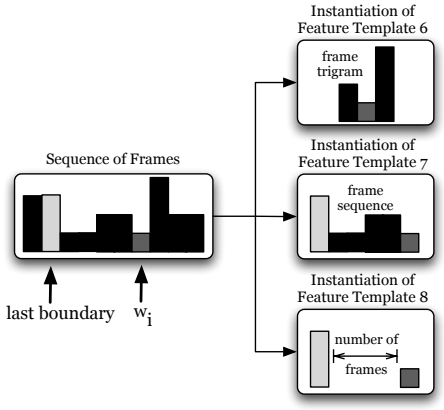


Figure 4. Classification example for current frame, w_i. The label for w_i is dependent on a number of features, including the frame size sequence since the last hypothesized boundary (shown here in gray). An example feature derived from each of templates 6-8 is depicted on the right hand side.

are not detectable by frame size changes (such as the long sequence of same-sized frames in Figure 3), we also model features such as phoneme length (feature template 8).

To efficiently solve the optimization problem posed by maximum entropy modeling, we use the *megam* framework with the limited memory BGFS [39] algorithm to obtain the model $p(w|v)$. Having built a model, we estimate the probability of each frame, in order, being a phoneme boundary by evaluating the estimated posterior $p(w|v)$. Since feature templates 7 and 8 depend on previously classified labels, we use a dynamic programming algorithm to maximize the likelihood of the sequence as a whole rather than greedily selecting the most likely label for each frame. The algorithm, a beam search, stores a list of the l most likely candidate segmentations up to the current frame; this list is updated after each frame is evaluated. We choose as our final segmentation the most likely candidate at the last frame.

Evaluation

In order to provide rigorous assessments of our methodology, we perform *cross-validation* in the segmentation and classification stages of our experiments. Cross-validation is a method for estimating the generalization performance of a classifier by partitioning the available data into complementary subsets, training with one subset, and testing with the other. In particular, we perform k-fold cross-validation, in which the data is partitioned into k complementary subsets. For each fold, one subset is selected for testing and the remainder used for training. The training and testing are performed as many times as there are subsets, with each acting as the testing set in one fold. The results of all iterations are then averaged to give the expected generalization performance, which mitigates the possibility of experimental

results being unduly influenced by fortuitous selection of training and testing data.

To evaluate the performance of our phonetic segmentation model, we perform a 5-fold cross-validation experiment for each dialect in the TIMIT corpus. Using a holdout set of female speakers from the New England dialect, we experimentally determined an optimal value of 8 for the beam width l. We report the performance using precision (i.e., the fraction of boundaries in the transcription that are present in the reference transcription) and recall (i.e., the fraction of boundaries in the reference that appear in the transcription) as our metrics.

	$n = 1$		$n = 2$	
Dialect	Precision	Recall	Precision	Recall
New England	0.8539	0.7233	0.9443	0.8735
Northern	0.8555	0.7332	0.9458	0.8837
North Midland	0.8509	0.7372	0.9402	0.8901
South Midland	0.8452	0.7086	0.9352	0.8627
Southern	0.8525	0.7037	0.9405	0.8586
New York City	0.8530	0.7096	0.9386	0.8628
Western	0.8586	0.7259	0.9439	0.8652
Army Brat	0.8465	0.7540	0.9389	0.8985

Table II
PHONETIC SEGMENTATION PERFORMANCE FOR EACH DIALECT IN THE TIMIT CORPUS.

While interpreting these results, we note that Raymond et al. [47] have shown that phoneme boundaries are inexact even at the frame level—in fact, in their study, human transcribers agreed (within 20ms) on less than 80% of the boundaries. For this reason, a frame classified as a boundary is considered as correct if it occurs within n frames of an actual boundary; likewise, it is incorrect if there are no actual boundaries within n frames. Table II summarizes, for each dialect, our segmentation performance for $n = 1$ and $n = 2$. For the sake of comparison, we also note that state-of-the-art classifiers (operating on the raw acoustic signal) are able to recall approximately 80% (again, within 20ms) of phoneme boundaries in TIMIT [18, 44] with error rates similar to our own. Unfortunately, the comparison is not direct: our labels are necessarily at the granularity of frames (each 20ms), rather than samples, which means that the within-n-frame requirement for agreement is looser than the within-20ms requirement.

The results in Table II show that our performance is on par with these other techniques. More importantly, the imprecision in the transcription boundaries does not negatively impact the performance of the next stage of our approach since the frames in question, i.e., the beginning and ending frames of each phoneme, are precisely those that will contribute the most variance to a phoneme model. In other words, the transition frames are likely to incorporate a significant amount of noise, due to their proximity to surrounding phonemes, and are therefore unlikely to be useful for classifying phonemes. It is for exactly this reason that we explicitly exclude the transition frames in the phoneme classification stage that follows. Finally, for the remainder of this paper we make the simplifying assumption that phoneme boundaries can be recognized correctly; this assumption is revisited in Section VI.

B. Classifying Phonemes (Stage ❷)

We remind the reader that our overall approach requires that we segment a sequence of encrypted packet lengths into subsequences corresponding to individual phonemes, and then classify these subsequences based on empirical models derived from labeled training data. We therefore have a classification problem where the classes of interest are the various phonemes.

For classification, we employ a combination of two systems: one context-dependent, wherein the labeling of a segment is dependent on the labelings of its neighbors, and another context-independent, wherein a single segment is considered in isolation. We combine these two approaches in order to leverage the strengths of each. Our context-dependent classifier is also based on maximum entropy modeling, while the context-independent classifier is based on *profile hidden Markov modeling*. Profile HMMs have been used widely in both the biological sequence analysis [16] and speech recognition communities [29].

Aside from the ability to incorporate contextual information, maximum entropy modeling is *discriminative*. Discriminative models are often used for classification tasks because they model only the parameters of interest for classification and thus can often encode more information. Hidden Markov models, on the other hand, are *generative* models. Generative models are sometimes preferable to discriminative models because they model the entire distribution over examples for a given class rather than just the information necessary to discriminate one class from another.

To combine the two models, we utilize a form of Bayesian inference to update the posterior given by the maximum entropy classifier with the "evidence" given by the profile HMM classifier. The updated posterior is then passed to a language model, as described below. By utilizing both types of models we enjoy increased classification accuracy while providing input to the language model with a valid statistical interpretation. Next, we discuss each stage in turn.

Maximum Entropy Discrimination of Phonemes

We discriminate between phonemes in a manner similar to the segmentation process described in Section V-A. Specifically, we define a new set of feature templates over sequences of phonemes (which are themselves composed of sequences of frame sizes). For pedagogical reasons, the specifics are given in Table III and an example feature is illustrated in Figure 5.

Feature templates 1-3 capture the exact frame sequence of the current and surrounding phonemes to identify phonemes

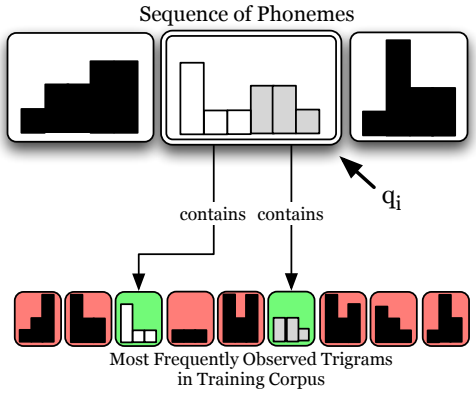


Figure 5. An example instantiation of feature template 10 which illustrates how the template models the presence of common trigrams.

Phoneme Classification Feature Templates	
1	q_i (i.e., the current phoneme's frame size sequence)
2	q_{i-1} (i.e., the previous phoneme's frame size sequence)
3	q_{i+1} (i.e., the next phoneme's frame size sequence)
4	q_i, excluding the first and the last frames
5	q_{i-1}, excluding the first and the last frames
6	length of q_i (in frames)
7	length of q_{i-1} (in frames)
8	frequency of frame size n in q_i
9	bigram b of frame sizes is in q_i, for top 100 bigrams
10	trigram t of frame sizes is in q_i, for top 100 trigrams
11	bigram b of frame sizes is in q_{i-1}, for top 100 bigrams
12	trigram t of frame sizes is in q_{i-1}, for top 100 trigrams
13	bigram b of frame sizes is in q_{i+1}, for top 100 bigrams
14	trigram t of frame sizes is in q_{i+1}, for top 100 trigrams

Table III
FEATURE TEMPLATES FOR THE MAXIMUM ENTROPY PHONEME CLASSIFIER. WE DENOTE AS q_i THE SEQUENCE OF FRAME SIZES FOR THE iTH PHONEME.WE LIMIT THE NUMBER OF n-GRAMS TO 100 FOR PERFORMANCE REASONS.

that frequently encode as exactly the same frame sequence. Feature templates 4 and 5 encode similar information, but drop the first and last frames in the sequence in accordance with our earlier hypothesis (see Section V-A) that the beginning and ending frames of the phoneme are the most variable. Feature templates 6 and 7 explicitly encode the length of the current and previous phonemes since some types of phonemes are frequently shorter (e.g., glides) or longer (e.g., vowels) than others. Feature template 8 captures the frequency of each possible frame size in the current sequence. Feature templates 9-14 encode the presence of each of the 100 most frequent frame size bigrams or trigrams observed in the training data; we limit the number of bigrams and trigrams to maintain manageable run-time performance. Finally, since we later incorporate high-level contextual information (such as neighboring phonemes) explicitly with a language model, we do not attempt to leverage that information in the classification model.

Profile HMM Modeling of Phonemes

To provide generative models of the various phonemes, we train a profile HMM for each. A profile HMM is a hidden Markov model with a specific topology that encodes a probability distribution over finite sequences of symbols drawn from some discrete alphabet. In our case, the alphabet is the different sizes at which a speech frame may be encoded; in Speex's wideband VBR mode, there are 19 such possibilities. Given the topology of a hidden Markov model, we need to estimate the parameters of the model for each set of sequences. Towards this end, we utilize a well-known algorithm due to Baum et al. [4] that iteratively improves the model parameters to better represent the example sequences.

Classification

To label an observed sequence of packet sizes, we find the posterior probability $p(r|q)$, where q represents the observed sequence of frame sizes, for each class label r. For the standalone maximum entropy classifier, the output for a given observation and label is an estimate of the desired quantity. For the profile HMM classifier, we calculate, using Bayesian inference, the posterior $p(r|q) = p(r)p(q|r)$ using the likelihood[5] $p(q|r)$, given by the profile HMM. This "updates" a prior probability $p(r)$ with the new "evidence" from the profile HMM. For the stand-alone classifier evaluation, we estimate the prior $p(r)$ as the proportion of examples belonging to the class in our training data. When using both the profile HMM and maximum entropy classifiers in conjunction, we use the estimated $p(r|q)$ from the maximum entropy model as the prior $p(r)$. In all cases, we choose the label whose model has the maximum posterior probability as the predicted label for a given sequence. These posterior probabilities also give a probability distribution over candidate labels for each phoneme in an utterance; these serve as the language model input.

Enhancing Classification using Language Modeling

Lastly, in order to incorporate contextual information on surrounding phonemes, we apply a trigram language model using the `SRILM` language modeling toolkit [51]. In particular, we train a trigram language model over both phonemes and phoneme types (e.g., vowels and stops). We disambiguate between candidate labels by finding the maximum likelihood sequence of labels given both the estimated distributions output by the classifier and the phonetic language model.

Evaluation

Our preliminary results show that we can correctly classify 45% of phonemes in a 10-fold cross-validation experiment on the New England dialect.[6] For this experiment,

[5]The likelihood given by an HMM is scaled by the marginal $p(q)$.

[6]For brevity, we omit the other dialects as the results do not differ significantly.

we operate on input with perfectly segmented phonetic boundaries so as to provide a baseline for our classifiers when evaluated independently from the other stages in our method. As can be seen from Figure 6, the combination of the profile HMM and maximum entropy classifiers with the language model outperforms the individual classifiers.

While this classification performance might sound lackluster, these results are quite surprising given the limited context we operate under (i.e., packet sizes only). For instance, recent approaches working directly on the *acoustic signal* report 77% accuracy on the TIMIT dataset in the context-dependent case (which corresponds roughly to our approach after application of the language model). In the context-independent case (analogous to our profile HMM classification approach without the language model), accuracy rates as high as 67% have been achieved [26] on the TIMIT dataset. Similarly, expert human transcribers achieve rates only as high as 69% [36].

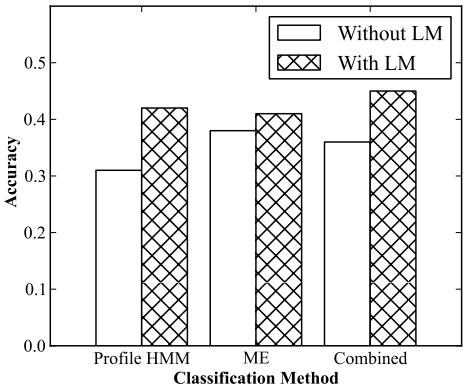


Figure 6. Phoneme classification accuracy on the New England dialect for the profile HMM and maximum entropy classifiers alone, in combination, and with the language model applied.

C. Segmenting Phoneme Streams into Words (Stage ❸)

In this stage, our task is to identify likely word boundaries from the stream of classified phonemes. To do so, we follow the methodology suggested by Harrington et al. [24] that, until very recently, was among the best approaches for word boundary identification. We also extend their approach to incorporate an additional step that makes use of a pronunciation dictionary.

Harrington et al. identify word breaks with a two-step process. The first step consists of inserting potential word breaks into the sequence of phonemes in positions that would otherwise produce invalid phonemic triples, i.e., triples that do not occur within valid words in English. Each such identified triple then causes the insertion of a pair of potential word breaks, one between each pair of phonemes in the triple. To resolve which of the potential word breaks are actual boundaries, we match the surrounding phonemes with all possible phonemes and pairs of phonemes which can begin or end words, and remove potential word breaks which would result in invalid word beginnings or endings.

We then perform an additional step whereby we use a pronunciation dictionary to find valid word matches for all contiguous subsequences of phonemes. For each such subsequence, we insert word breaks at the positions that are consistent across all the matches. For example, suppose the sequence [mɔɪliɹæg] (‘an oily rag’) has the following three possible segmentations:

- [ɪn ɔɪli ɹæg] (‘an oily rag’)
- [ɪn ɔɪl i ɹæg] (‘an oil E. rag’)
- [ɪn ɔ ɪl i ɹæg] (‘an awe ill E. rag’)

Since these choices have two words in common, we segment the phrase as [ɪn ɔɪli ɹæg].

Dialect	Precision	Recall
New England	0.7251	0.8512
Northern	0.7503	0.8522
North Midland	0.7653	0.8569
South Midland	0.7234	0.8512
Southern	0.7272	0.8455
New York City	0.7441	0.8650
Western	0.7298	0.8419
Army Brat	0.7277	0.8461

Table IV
WORD BREAK INSERTION PRECISION AND RECALL

The results of a 10-fold cross-validation experiment are given in Table IV. Overall, we achieve average precision and recall of 73% and 85%, respectively. Very recent results, however, by Blanchard et al. [8] and Hayes and Wilson [25] suggest that accuracy above 96% can be achieved using more advanced techniques than implemented here. Due to time and resource constraints, we make the simplifying assumption that word breaks can be correctly recognized. We revisit this assumption in Section VI.

D. Identifying Words via Phonetic Edit Distance (Stage ❹)

The final task is to convert the subsequences of phonemes into English words. To do so, we must identify words that best match the pronunciation dictated by the recovered phonemes. Towards this end, we design a novel metric of phonetic distance based on the difference in articulatory features (i.e., the associated physiological interactions discussed in Section II) between pairs of phonemes. Our approach has some similarities to ideas put forth by Oakes [45], which itself builds upon the work of Gildea and Jurasky [22] and Zobel and Dart [58, 59]. Oakes [45] proposes a phonetically-based alignment algorithm, though there is no notion of relative distance between various places or manners of articulation. In Zobel and Dart [59], the distances between phonemes are handcrafted, and their matching algorithm considers only the single most likely pronunciation.

In our approach, we define the distance between a vowel and a consonant as one unit, with a few exceptions: we assign a cost of 0 for converting an [i] to a [j] (or vice-versa) as well as for converting a [u] to a [w]. We do so because [w] and [j] are what are known as *semi-vowels*, and are essentially very short realizations of their corresponding vowels. Moreover, we assign a cost of 0 for [ɾ] (i.e., flap 'r') and [t], as well as for [ɾ] and [d]. This is because [ɾ] is an allophone of [t] and [d]. Hence, we would like such minor phonetic alterations to have little effect on the distance between two pronunciations.

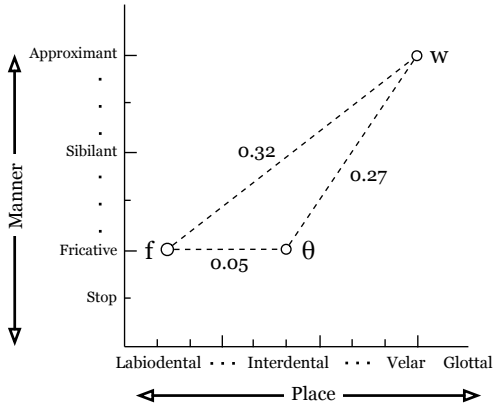


Figure 7. Illustration of distance between consonants [f], [θ], and [w].

To measure the distance between two vowels or two consonants, we use three different articulatory features as axes and calculate the Euclidean distance (see Figure 7) between the points corresponding to the two phonemes (scaled to a maximum of one unit). For consonants these features are *voice*, *manner*, and *place* of articulation. For vowels they are *rounding*, *backness* and *height*. Thus we differentiate between substitution of a phonetically similar segment, such as replacement of [s] (as in 'see') by [ʃ] (as in 'she'), or of a completely different segment, such as of [s] (as in 'seen') with [k] (as in 'keen').

Word 1	Word 2	Phonetic Distance	Primary Difference
bæt 'bat'	mæt 'mat'	0.0722	manner
bit 'beat'	bæt 'bat'	0.1042	height
did 'deed'	bid 'bead'	0.1050	place
bʌt 'but'	bɔt 'bought'	0.1250	rounding
bʌt 'but'	bæt 'bat'	0.1267	backness
bid 'bead'	bit 'beat'	0.5774	voicing
fɒðɝ 'father'	mʌðɝ 'mother'	0.7292	n/a
hʊkt 'hooked'	fɒnɪks 'phonics'	2.9573	n/a
hɛloʊ 'hello'	wɝld'world'	3.1811	n/a

Table V
EXAMPLES OF OUR PHONETIC EDIT DISTANCE BETWEEN PAIRS OF EXAMPLE WORDS. THE LAST COLUMN LISTS THE PRIMARY DIFFERENCE (IN TERMS OF ARTICULATORY PROCESSES).

To compare two sequences of phonemes, we use the Levenshtein distance with insertions and deletions weighted at one unit and edits weighted according to their phonetic distance as defined above. Table V gives example word comparisons along with their primary differences (in terms of articulartory processes).

In order to determine the optimal values for the insertion and deletion weights for our phonetic edit distance metric, we performed a simple parameter space exploration. We hypothesized that the absolute insertion and deletion costs were less significant than the difference between them. As such we tuned based on two parameters, *base cost* and *offset*. Each insertion costs the base cost plus half the offset and each deletion costs the base cost minus half the offset. The effectiveness of each set of parameters is shown in Figure 8. Somewhat surprisingly, a base cost of 1.0 and offset of 0.0 (corresponding to insertion and deletion weights of 1.0) provided the highest average word accuracy.

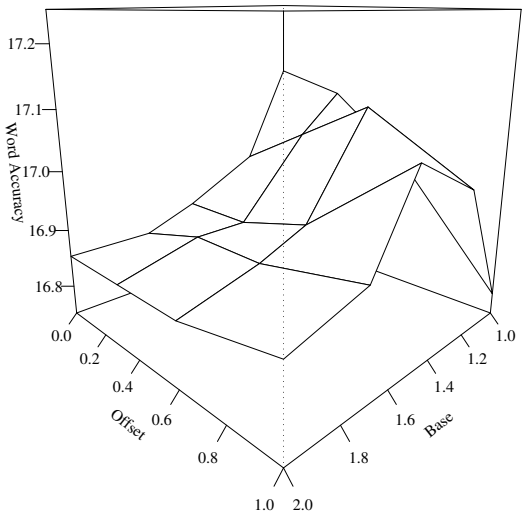


Figure 8. Parameter space exploration, in terms of average word accuracy, for our phonetic edit distance.

To match a sequence of phonemes to an English word, we compute the phonetic distance between the sequence and each pronunciation in our dictionary in order to obtain a list of the closest pronunciations to the sequence. However, the existence of homophones means that, even if the pronunciation is correct, we may have many choices for the word spoken. For example, 'ate' and 'eight' are indistinguishable phonetically: both are pronounced [eɪt].

In order to disambiguate between homophones, we incorporate a word and part-of-speech based language model to choose between the candidate words using contextual information from the sentence as a whole. Thus we can disambiguate between 'ate' and 'eight' by finding the most likely part of speech (e.g., noun, verb, pronoun, or adverb) for that position in the sentence. Using the `SRILM` language

modeling toolkit, we train a trigram language model over both words and parts-of-speech on the well-known Brown corpus [20]. The part of speech tags used are those currently implemented in `NLTK` [40]. To improve the ability of the language model to disambiguate between candidate words, we assign each word a weight which estimates the conditional probability of the observed pronunciation given the candidate word.

To find these weights, we need a measure of how likely an observed pronunciation is given the phonetic distance to the actual pronunciation of the given word; therefore, we estimate the cumulative distribution function (CDF) over phonetic distances by deriving an empirical CDF (see Figure 9) from the distances of a large number of pronunciation pairs. We then transform the given distance between pronunciations into a probability estimate by evaluating the empirical CDF at that distance. For each pronunciation in the candidate list for an observed word, we weight the associated words with the probability estimate for that pronunciation.[7] Thus we have, for each word in an utterance, a list of candidate words with associated conditional probability estimates. Disambiguation is performed by finding the maximum likelihood sequence of words given the candidates, their probability estimates, and the language model.

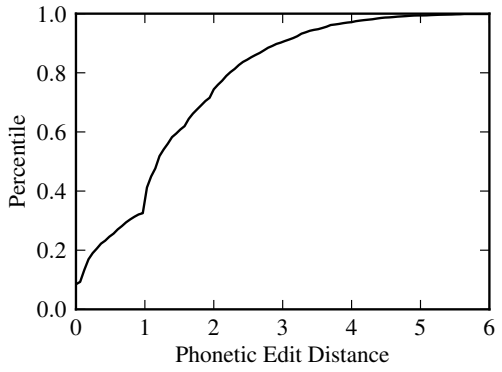


Figure 9. Empirical CDF of phonetic edit distance.

At this point, the observant reader will have surely noted that the overall process is fundamentally inexact because, in the end, some sort of human judgement is required to evaluate the hypothesized output. That is, we need some way to measure the quality of our guesses, say, as assessed by a human judge who compares them to the actual transcript. Thankfully, the closely related problem of scoring machine translations has been extensively studied. In what follows, we discuss how we measure the accuracy of our guesses.

[7]A word which is associated with multiple pronunciations is weighted according to the closest pronunciation, i.e., we take the maximum weight of all associated weights for the given word.

E. Measuring the Quality of Our Output

Since the early 1990s, much work has gone into finding appropriate metrics for scoring machine transcriptions from automatic speech recognition and transcription systems. In that context, the main task is to generate a literal transcription of every word that is spoken. The closer the machine transcription is to a human translation, the better it is. Early approaches for automatically measuring such performance simply relied on examining the proportion of word errors between the actual and transcribed conversations (i.e., the Word Error Rate (WER)), but WER has been shown to be a poor indicator of the quality of a transcript since good performance in this context depends not only on the amount of errors, but also on the types of errors being made. For example, from the perspective of human interpretation, it often does not matter if the transcribed word is 'governed' instead of 'governing'.

Hence, modern automatic scoring systems reward candidate text based on the transcription's *adequacy* (i.e., how well the meaning conveyed by the reference transcription is also conveyed by the evaluated text) and *fluency* (i.e., the lengths of contiguous subsequences of matching words). To date, many such scoring systems have been designed, with entire conferences and programs dedicated solely to this topic. For instance, NIST has coordinated evaluations under the Global Autonomous Language Exploitation (GALE) program since the mid-nineties. While the search for better metrics for translation evaluation remains an ongoing challenge, one widely accepted scoring system is the *METEOR Automatic Metric for Machine Translation* by Lavie and Denkowski [35]. METEOR was designed to produce quality scores at the sentence level which correlate well with those assigned by human judges. We evaluate the quality of our guesses using METEOR; for concreteness, we now review pertinent details of that scoring system.

Lavie and Denkowski's method evaluates a hypothesized transcription by comparison with a reference transcription. The two transcripts are compared by aligning first exact word matches, followed by stemmed word matches, and finally synonymous word matches. The alignment is performed by matching each unigram string in the reference transcription to at most one word in the hypothesis transcription. To compute the score from such an alignment, let m be the number of matched unigrams, h the number of unigrams in the hypothesis, and r the number of unigrams in the reference. The standard metrics of unigram precision ($P = m/h$) and recall ($R = m/r$) are then computed.

Next, the parameterized f-score, i.e., the harmonic mean of P and R given a relative weight (α) on precision, is computed:

$$F_{mean} = \frac{P * R}{\alpha * P + (1 - \alpha) * R}.$$

To penalize hypotheses which have relatively long sequences

of incorrect words, Lavie and Denkowski count the number c of 'chunk' sequences of matched unigrams which are adjacent, and in the correct order in the hypothesis. A *fragmentation penalty* is then computed as $P_{frag} = \gamma * (c/m)^{\beta}$, where γ and β are parameters determining the maximum penalty and relative impact of fragmentation, respectively. The final METEOR score is then calculated as $S_m = (1 - P_{frag}) * F_{mean}$ for each hypothesis.

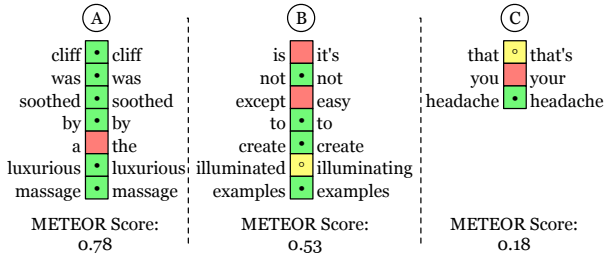


Figure 10. Example scoring of three hypothesized guesses. For each, the hypothesized guess is on the left, with the reference on the right. Filled circles represent exact matches. Hollow circles show matches based on stemming.

Denkowski and Lavie [13] performed extensive analysis to determine appropriate values for the parameters α, β, and γ which optimize the correlation between METEOR score and human judgments. In our experiments, we use the parameter set that is optimized to correlate with the *Human Targeted translation Edit Rate* (HTER) metric for human judgement on the GALE-P2 dataset [14]. We disable synonym matching as our system does no semantic analysis, and thus any such matches would be entirely coincidental. Some examples are shown in Figure 10. Notice that even a single error can result in scores below 0.8 (e.g., in part (a)). Moreover, in some cases, a low score does not necessarily imply that the translation would be judged as poor by a human (e.g., one can argue that the translation in part (c) is in fact quite decent). Finally, Lavie indicates that scores over 0.5 "generally reflect understandable translations" and that scores over 0.7 "generally reflect good and fluent translations" in the context of machine translation [34].

VI. Empirical Evaluation

In the analysis that follows, we explore both content-dependent and content-independent evaluations. In both cases, we assume a speaker-independent model wherein we have no access to recordings of speech by the individual(s) involved in the conversation. In the content-dependent case, we perform two experiments, each incorporating multiple different utterances of a particular sentence. We use TIMIT's `SA1` and `SA2` sentences for these experiments because each is spoken exactly once by each of the 630 speakers, providing a rare instance of sufficient examples for evaluation. In the content-independent case, we incorporate all TIMIT utterances.[8] Except where explicitly specified, all experiments are 10-fold cross-validation experiments and are performed independently on each dialect. As discussed in Section V, for these experiments we assume that the segmentation of phonemes is correct to within human transcriber tolerances. However, the effects of this assumption are specifically examined in a small experiment described separately below.

`SA1`:"She had your dark suit in greasy wash water all year."	Score
She had year dark suit a greasy wash water all year.	0.67
She had a dark suit a greasy wash water all year.	0.67
She had a dark suit and greasy wash water all year.	0.67
`SA2`:"Don't ask me to carry an oily rag like that."	Score
Don't asked me to carry an oily rag like that.	0.98
Don't ask me to carry an oily rag like dark.	0.82
Don't asked me to carry an oily rag like dark.	0.80

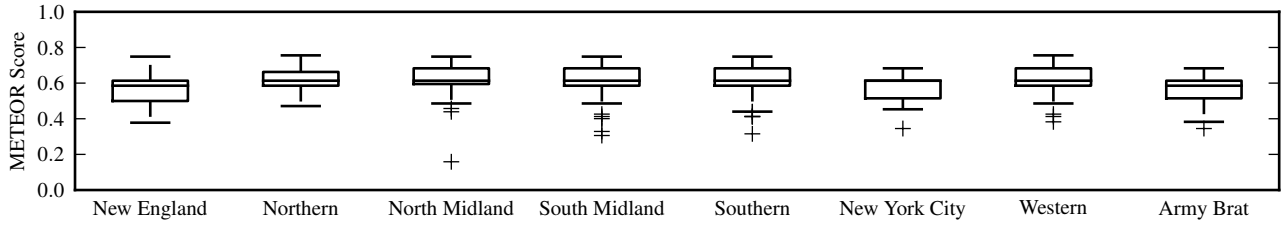
Table VI
Top scoring hypotheses from the New England dialect.

Figure 11 shows the distributions of METEOR scores under each of the dialects for the two content-dependent experiments. For `SA1`, the results are fairly tightly grouped around a score of 0.6. The `SA2` scores show significantly more variance; while some hypotheses in this case were relatively poor, others attained perfect scores. To ease interpretation of the scores, we provide the three highest-scoring hypotheses for each sentence, along with their scores, in Table VI. In addition, recall that sentences with scores over 0.5 are generally considered understandable in the machine translation context; 91% of our `SA1` reconstructions and 98% of our `SA2` reconstructions exceed this mark.

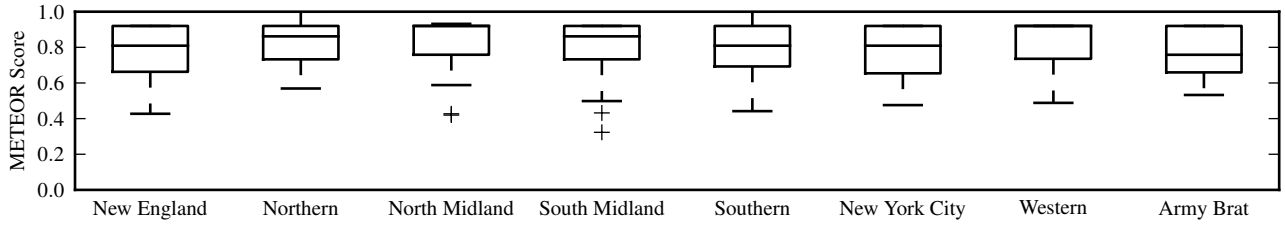
The independent case, on the other hand, proves to be a more challenging test for our methodology. However, we are still able to reconstruct a number of sentences that are easily interpretable by humans. For instance, Table VII shows the five highest-scoring hypotheses for this test on the New England dialect. In addition, a number of phrases within the sentences are exactly correct (e.g., 'the two artists'). For completeness, we note that only 2.3% of our reconstructions score above 0.5. However, the average score for the top 10% (see Figure 12) is above 0.45. That said, we remind the reader that no reconstruction, even a partial one, should be possible; indeed, any cryptographic system that leaked as much information as shown here would immediately be deemed insecure.

To mitigate any concern regarding our two previous simplifying assumptions, namely, the accurate segmentation of frame size sequences on phoneme boundaries and of (noisy) phoneme sequences on word boundaries, we perform one final experiment. We believe sufficient evidence has been given to show that we can accomplish these tasks in isolation; however, one possible critique stems from the

[8]We follow the standard practice in the speech recognition community and use the `SA1` and `SA2` sentences for training only.



(a) SA1: "She had your dark suit in greasy wash water all year."



(b) SA2: "Don't ask me to carry an oily rag like that."

Figure 11. METEOR scores for all hypothesized transcripts of sentences SA1 and SA2 for each dialect in the TIMIT dataset.

Hypothesis	Reference Sentence	METEOR Score
① Codes involves the displacement of aim.	Change involves the displacement of form.	0.57
② The two artists instance attendants.	The two artists exchanged autographs.	0.49
③ Artificial intelligence is carry all.	Artificial intelligence is for real.	0.49
④ Bitter unreasoning dignity.	Bitter unreasoning jealousy.	0.47
⑤ Jar, he whispered.	Honey, he whispered.	0.47

Table VII
THE FIVE HIGHEST SCORING HYPOTHESES FROM THE NEW ENGLAND DIALECT UNDER THE CONTENT-INDEPENDENT MODEL.

potential effects, when these assumptions are lifted, on the efficacy of the methodology as a whole. Thus we remove these assumptions in a small, content-independent experiment comprised of the audio samples spoken by female speakers in the "Army Brat" dialect of the TIMIT corpus. The average score for the top 10%, in this case, is 0.19, with a high score of 0.27. We remind the reader that even such low scoring hypotheses can be interpretable (see Figure 10), and we stress that these results are preliminary and that there is much room for improvement—in particular, recently proposed techniques can be directly applied in our setting (see Section V-C). Moreover, there are opportunities for extensions and optimizations at every stage of our approach, including, but not limited to, weighting the influence of the different classification and language models. In addition, other scoring systems for machine translation exist (e.g., NIST and BLEU), which may be appropriate in our context. We plan to explore these new techniques, optimizations and metrics in the future.

A. An Adversarial Point of View (Measuring Confidence)

Due to the difficult nature of our task (i.e., numerous factors influencing phonetic variation and the fact that we operate on *encrypted* data), an adversary is unlikely to be able to construct an accurate transcript of every sentence uttered during a conversation. Therefore, she must have some way to measure her confidence in the output generated, and only examine output with confidence greater than some threshold. To show this can be done, we define one such confidence measure, based on our phonetic edit distance, which indicates the likelihood that a given transcript is approximately correct.

Our confidence measure is based on the notion that close pronunciation matches are more likely to be correct than distant matches. We use the mean of the probability estimates for each word in a given hypothesized transcript as our confidence value for that hypothesis. Analysis indicates that this confidence measure correlates (see Figure 13) with the maximum METEOR score obtained from the 10 best hypotheses output by the word-level language model (Stage ❹). This implies that, given a set of training data such as the TIMIT dataset, an adversary can determine an appropriate threshold for the calculated confidence values to suit her preference as to the balance between precision and recall in the hypothesized transcripts. The results in Figure 14 provide one such analysis under the content-dependent model. We note that the threshold reduces the set of hypotheses to a subset with improved METEOR scores.

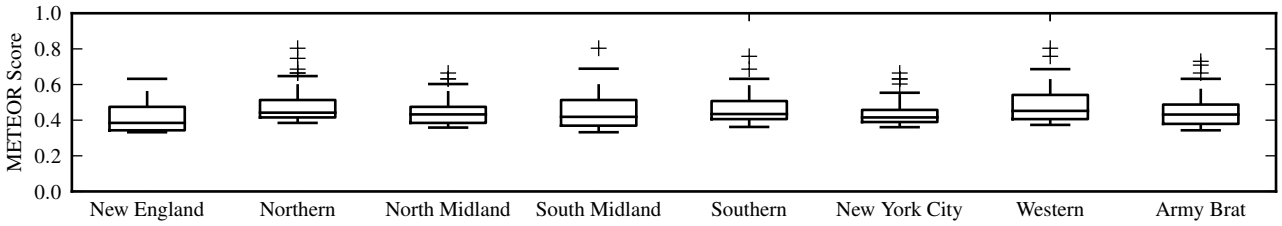


Figure 12. The top 10% of METEOR scores for hypothesized transcripts under the content-independent assumption.

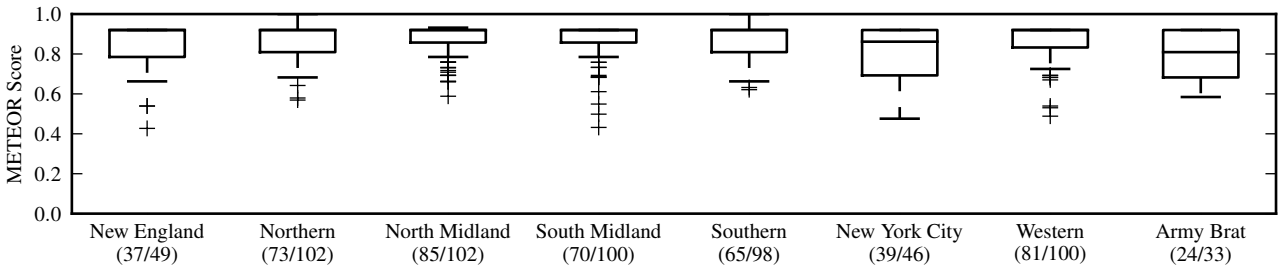


Figure 14. METEOR scores for hypothesized transcripts for sentence SA2 with confidence values above the threshold of .90 for each dialect in the TIMIT dataset. The number of transcripts with confidence values above the threshold compared to the total for each dialect is shown in parentheses.

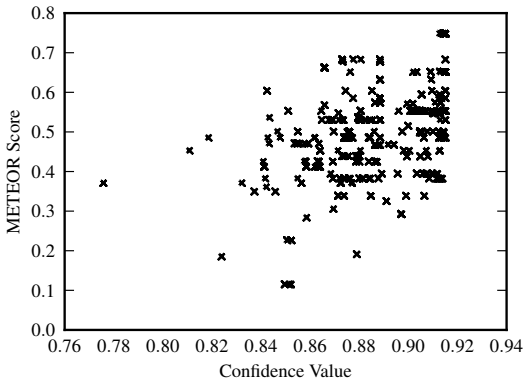


Figure 13. Scatter plot of METEOR scores against our confidence values (Pearson's r-value of 0.43).

Unfortunately, the correlation of this particular confidence metric does not extend well to the content-independent model. However, we note that there are many other methods to measure confidence which an adversary could leverage, including those based on the posteriors output by the classification stage, the likelihoods given by the language models, and ex post facto analysis of the well-formedness (in terms of syntax, i.e., grammar) of the hypotheses. We hope to explore these strategies in the near future.

In closing, we note that one could also apply the notion of confidence values to interpreting the results at the word, rather than the sentence, level. In particular, we could filter our hypotheses at the word-level by only outputting those words for which we have high confidence. Preliminary results indicate that such "masking" may provide benefits to interpretation, for example, outputting 'nonprofit $\star$ all $\star$ $\star$ raisers' instead of 'nonprofit organizations all swiftly fairy raisers' as the hypothesis for 'nonprofit organizations have frequent fund raisers'. We forego such analysis at this time since the METEOR metric does not allow for unknown words—an automated method of evaluating such hypotheses is necessary before we can make any claims.

B. Discussion & Mitigation

We note, like other work in this area (e.g., [2, 37, 41, 55, 56]), that we assume each packet contains a single frame. However, some recently designed codecs, such as Skype's new codec (dubbed SILK), can vary the number of frames per packet based on network conditions. It therefore remains to be seen if the approach outlined herein can be adapted to that setting; exploring that, however, requires a substantial data collection effort that is beyond the scope of this work.

Further, our experiments assume that packets are observed in correct order and are not fragmented or combined, i.e., the adversary can observe packets at the level of the local network (e.g., between VoIP endpoint and hub or PBX) or can perform IP defragmentation or TCP stream reassembly. The inherently limited fidelity of the channel, however, suggests that our technique would be robust to reasonable noise in the form of packet reordering and fragmentation.

Lastly, a knee-jerk reaction to thwarting this and other aforementioned threats to VoIP is to simply use constant bit-rate codecs or block ciphers. However, variable bit-rate encoded audio encrypted under a block cipher with a

small block size is theoretically vulnerable to our attack. Packet sizes in that scenario still correlate with input signals, albeit at a reduced fidelity; thus relatively large block sizes are necessary to ensure privacy. For this reason, the use of constant bit-rate codecs is important to consider as an alternative to simple block ciphers for VoIP, since such codecs might improve call quality given a relatively large fixed packet size. Another alternative might even be to drop or pad packets [19, 27, 57], though, in that case, the effect on perceived call quality is unclear. We note, however, that VoIP providers have made no move to employ any such measures: Skype's `SILK`, for instance, is a VBR codec. Similarly, one of the leading proposals for 4G, the LTE Advanced standard, specifies a VBR codec for audio [1] and the use of SRTP to secure voice data channels.

VII. Conclusion

In this paper, we explore the ability of an adversary to reconstruct parts of encrypted VoIP conversations. Specifically, we propose an approach for outputting a hypothesized transcript of a conversation, based on segmenting the sequence of observed packets sizes into subsequences corresponding to the likely phonemes they encode. These phoneme sequences are then mapped to candidate words, after which we incorporate word and part-of-speech based language models to choose the best candidates using contextual information from the hypothesized sentence as a whole. Our results show that the quality of the recovered transcripts is far better in many cases than one would expect. While the generalized performance is not as strong as we would have liked, we believe the results still raise cause for concern: in particular, one would hope that such recovery would not be at all possible since VoIP audio is encrypted precisely to prevent such breaches of privacy. It is our belief that with advances in computational linguistics, reconstructions of the type presented here will only improve. Our hope is that this work stimulates discussion within the broader community on ways to design more secure, yet efficient, techniques for preserving the confidentiality of VoIP conversations.

VIII. Acknowledgements

We thank the anonymous reviewers for their insightful comments. We also thank Elliot Moreton of the Department of Linguistics at UNC-Chapel Hill for helpful suggestions. This work is supported in part by NSF grants CCF-1017318 and CNS-0852649.

References

[1] 3GPP. Extended adaptive multi-rate wideband (AMR-WB+) codec. Technical Report 26.290, 3rd Generation Partnership Project (3GPP), 2009.

[2] M. Backes, G. Doychev, M. Dürmuth, and B. Köpf. Speaker recognition in encrypted voice streams. In *Proc. 15th European Symposium on Research in Computer Security*, pages 508–523, 2010.

[3] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The secure real-time transport protocol (SRTP). RFC 3711, Internet Engineering Task Force, 2004.

[4] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[5] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Mach. Learn.*, 34(1-3), 1999.

[6] A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71, 1996.

[7] L. Bernaille and R. Teixeira. Early recognition of encrypted applications. In *PAM*, pages 165–175, 2007.

[8] D. Blanchard, J. Heinz, and R. Golinkoff. Modeling the contribution of phonotactic cues to the problem of word segmentation. *The Journal of Child Language*, 37(3):487–511, 2010.

[9] H. Bortfeld, J. Morgan, R. Golinkoff, and K. Rathbun. Mommy and me: Familiar names help launch babies into speech-stream segmentation. *Psychological Science*, 16:298–304, 2005.

[10] S. Chen, R. Wang, X. Wang, and K. Zhang. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 191–206, 2010.

[11] R. A. Cole and J. Jakimik. *A Model of Speech Perception*, chapter 6, pages 133–163. Lawrence Erlbaum Associates, 1980.

[12] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli. Traffic classification through simple statistical fingerprinting. *SIGCOMM Comput. Commun. Rev.*, 37(1):5–16, 2007.

[13] M. Denkowski and A. Lavie. Choosing the right evaluation for machine translation: an examination of annotator and automatic metric performance on human judgment tasks. In *Proceedings of AMTA*, 2010.

[14] M. Denkowski, A. Agarwal, S. Banerjee, and A. Lavie. *The METEOR MT Evaluation System, Version 1.2*. Carnegie Mellon University, Pittsburgh, PA, 2010.

[15] B. Dupasquier, S. Burschka, K. McLaughlin, and S. Sezer. Analysis of information leakage from encrypted Skype conversations. *International Journal of Information Security*, pages 1–13, 2010.

[16] R. Durbin, S. Eddy, A. Grogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[17] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli. Detection of encrypted tunnels across network boundaries. In *ICC*, pages 1738–1744, 2008.

[18] A. Esposito and G. Aversano. Text independent methods for speech segmentation. In *Nonlinear Speech Modeling and Applications*, volume 3445 of *Lecture Notes in Computer Science*, pages 261–290. Springer, 2005.

[19] P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, and W. Lee. Polymorphic blending attacks. In *Proceedings of the USENIX Security Symposium*, pages 241–256, 2006.

[20] W. N. Francis and H. Kucera. Brown corpus manual. Technical report, Department of Linguistics, Brown University, 1979.

[21] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue. TIMIT acoustic-phonetic continuous speech corpus, 1993.

[22] D. Gildea and D. Jurasky. Learning bias and phonological-rule induction. *Computational Linguistics*, 22(4):497–530,

1996.

[23] M. Halle. Knowledge unlearned and untaught: What speakers know about the sounds of their language. *Linguistic Theory and Psychological Reality*, pages 294–303, 1978.

[24] J. Harrington, G. Watson, and M. Cooper. Word boundary identification from phoneme sequence constraints in automatic continuous speech recognition. In *Computational linguistics - Volume 1*, pages 225–230, 1988.

[25] B. Hayes and C. Wilson. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry*, 39 (3):379–440, 2008.

[26] Y. Hifny and S. Renals. Speech recognition using augmented conditional random fields. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(2):354 –365, 2009.

[27] A. Iacovazzi and A. Baiocchi. Optimum packet length masking. In *22nd International Teletraffic Congress (ITC)*, pages 1–8, 2010.

[28] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106(4):620–630, May 1957.

[29] F. Jelinek. *Statistical Methods for Speech Recognition*. Massachusetts Institute of Technology, 1997.

[30] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2008.

[31] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: multilevel traffic classification in the dark. *SIGCOMM Comput. Commun. Rev.*, 35(4):229–240, 2005.

[32] T. Kempton and R. K. Moore. Language identification: Insights from the classification of hand annotated phone transcripts. In *Speaker and Language Recognition Workshop*, Jan. 2008.

[33] A. D. Keromytis. A survey of Voice over IP security research. In *Proceedings of the 5th International Conference on Information Systems Security*, pages 1–18, 2009.

[34] A. Lavie. Evaluating the output of machine translation systems. AMTA Tutorial, 2010.

[35] A. Lavie and M. J. Denkowski. The METEOR metric for automatic evaluation of machine translation. *Machine Translation*, 23:105–115, September 2009.

[36] K.-F. Lee and H.-W. Hon. Speaker-independent phoneme recognition using hidden Markov models. *The Journal of the Acoustical Society of America*, 84(S1):62, 1988.

[37] T. Leila and R. Bettati. Privacy of encrypted Voice-over-IP. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 3063 –3068, 2007.

[38] M. Liberatore and B. N. Levine. Inferring the source of encrypted HTTP connections. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 255–263, 2006.

[39] D. C. Liu, J. Nocedal, and D. C. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

[40] E. Loper and S. Bird. NLTK: The natural language toolkit, May 2002.

[41] Y. Lu. On traffic analysis attacks to encrypted VoIP calls. Master's thesis, Cleveland State University, Fenn College of Engineering, 2009.

[42] G. Maiolini, A. Baiocchi, A. Iacovazzi, and A. Rizzi. Real time identification of SSH encrypted application flows by using cluster analysis techniques. In *Proceedings of the International IFIP-TC 6 Networking Conference*, pages 182–194, 2009.

[43] A. McGregor, M. Hall, P. Lorier, and J. Brunskill. Flow clustering using machine learning techniques. In *PAM*, pages 205–214, 2004.

[44] I. Mporas, T. Ganchev, and N. Fakotakis. Speech segmentation using regression fusion of boundary predictions. *Computer Speech & Language*, 24(2):273–288, 2010.

[45] M. Oakes. Computer estimation of vocabulary in protolanguage from word lists in four daughter languages. *Journal of Quantitative Linguistics*, 7(3):233–243, 2000.

[46] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, 1996.

[47] W. D. Raymond, M. Pitt, K. Johnson, E. Hume, M. Makashay, R. Dautricourt, and C. Hilts. An analysis of transcription consistency in spontaneous speech from the Buckeye corpus. In *Proceedings of the International Conference on Spoken Language Processing*, 2002.

[48] T. S. Saponas, J. Lester, C. Hartung, S. Agarwal, and T. Kohno. Devices that tell on you: privacy trends in consumer ubiquitous computing. In *Proceedings of the USENIX Security Symposium*, pages 1–16, 2007.

[49] M. R. Schroeder and B. S. Atal. Code-excited linear prediction (CELP): High-quality speech at very low bit rates. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 10, pages 937–940, April 1985.

[50] D. X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on SSH. In *Proceedings of the USENIX Security Symposium*, pages 25–25, 2001.

[51] A. Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings of the ICSLP*, volume 2, pages 901–904, 2002.

[52] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu. Statistical identification of encrypted web browsing traffic. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2002.

[53] J.-M. Valin. The Speex codec manual, 2007.

[54] C. V. Wright, F. Monrose, and G. M. Masson. On inferring application protocol behaviors in encrypted network traffic. *Journal of Machine Learning Research*, 6:2745–2769, 2006.

[55] C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson. Language identification of encrypted VoIP traffic: Alejandra y Roberto or Alice and Bob? In *Proceedings of the USENIX Security Symposium*, 2007.

[56] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2008.

[57] C. V. Wright, S. E. Coull, and F. Monrose. Traffic morphing: An efficient defense against statistical traffic analysis. In *Proceedings of the Network and Distributed System Security Symposium*, 2009.

[58] J. Zobel and P. Dart. Finding approximate matches in large lexicons. *Software—Practice and Experience*, 25(3):331–345, 1995.

[59] J. Zobel and P. Dart. Fnetik: An integrated system for phonetic matching, 1996. RMIT, Technical Report 96-6.