Speaker Recognition in Encrypted Voice Streams

Michael Backes[1,2], Goran Doychev[1], Markus Dürmuth[1], and Boris Köpf[2]

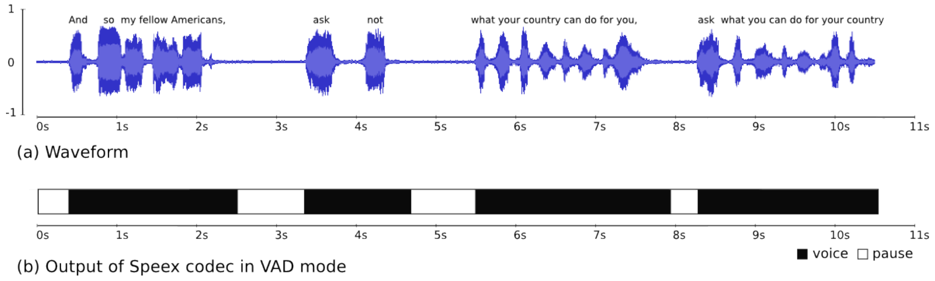
[1] Saarland University, Saarbrücken, Germany
[2] Max Planck Institute for Software Systems (MPI-SWS)

Abstract. Transmitting voice communication over untrusted networks puts personal information at risk. Although voice streams are typically encrypted to prevent unwanted eavesdropping, additional features of voice communication protocols might still allow eavesdroppers to discover information on the transmitted content and the speaker.

We develop a novel approach for unveiling the identity of speakers who participate in encrypted voice communication, solely by eavesdropping on the encrypted traffic. Our approach exploits the concept of voice activity detection (VAD), a widely used technique for reducing the bandwidth consumption of voice traffic. We show that the reduction of traffic caused by VAD techniques creates patterns in the encrypted traffic, which in turn reveal the patterns of pauses in the underlying voice stream. We show that these patterns are speaker-characteristic, and that they are sufficient to undermine the anonymity of the speaker in encrypted voice communication. In an empirical setup with 20 speakers our analysis is able to correctly identify an unknown speaker in about 48% of all cases. Our work extends and generalizes existing work that exploits variable bit-rate encoding for identifying the conversation language and content of encrypted voice streams.

1 Introduction

The past decades have brought dramatic changes in the way we live and work. The proliferation of networked devices, and the resulting abundance of exchanged information present significant opportunities, but also difficult conceptual and technical challenges in the design and analysis of the systems and programs that fuel these changes. A particularly important trend is the increasing need for protocols that run on open infrastructures such as wireless communication channels or the Internet and offer remote communication between different people anytime, anywhere. However, transmitting privacy-sensitive information over such open infrastructures raises serious privacy concerns. For instance, modern voice communication protocols should satisfy various security properties such as secrecy (the content of the voice communication should remain secret to eavesdroppers) or even anonymity (users participating in voice communications should remain anonymous to eavesdroppers in order to avoid being stigmatized or other negative repercussions). To achieve these security properties, voice communication is typically encrypted. For example, telephones based on the GSM [13] and UMTS [1] standards encrypt their voice data, and most implementations of VoIP



(a) Waveform

(b) Output of Speex codec in VAD mode

Fig. 1. Patterns of pauses in network traffic introduced when encoding an audio signal (above) with a VAD-enabled codec (below). Each audio packet is depicted as a 0.0145s long black- or white-colored bar, a black bar corresponding to a voice packet and a white bar corresponding to a pause packet. (Audio data: John F. Kennedy's inaugural address from January 20th, 1961)

telephony offer encryption on the application layer or support IPsec. The underlying rationale is that properly employing encryption hides both the content of the communication and the identity of the speaker, thereby ensuring both secrecy and anonymity. However, even properly deploying encryption does not exclude that additional features of voice communication protocols might still allow eavesdroppers to discover information about the transmitted content and the speaker.

1.1 Our contribution

We develop a novel approach for unveiling the identity of speakers who participate in encrypted voice communication, solely by eavesdropping on the encrypted traffic. Our approach exploits the concept of *voice activity detection* (VAD), which is a common performance-enhancing technique to detect the presence or absence of human speech. VAD-based techniques are used to reduce the volume of the transmitted data and are prevalent in standards for transmitting voice streams. For example, the GSM and UMTS standards use a VAD-technique called discontinuous transmission (DTX) to stop the transmissions if a speaker is idle, thereby saving battery power and reducing interference. Moreover, VoIP clients such as Skype [29], Google Talk [12], and Microsoft Netmeeting [20], as well as the US Army's Land Warrior system, employ voice codecs that decrease the number and/or size of packets when a speaker is idle. This reduces network utilization, which is a primary concern in packet-switched computer networks.

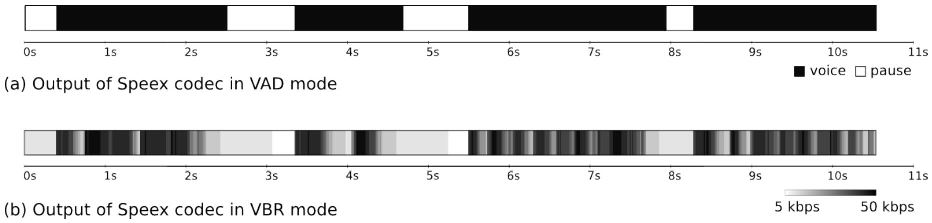
We show that – even when traffic is encrypted – the reduction of traffic caused by VAD techniques creates patterns in the traffic, which in turn reveal patterns of pauses in the underlying voice stream (see Figure 1). We show that these patterns are speaker-characteristic, and that they are sufficient to undermine the anonymity of the speaker in encrypted voice communication.

Our approach relies on supervised learning and works as follows. In a preparation phase, we collect encrypted samples of voice stream data from a set of candidate speakers, the so-called *training data*. From this training data, we build a stochastic model of the pause characteristics for each candidate, i.e., the relative frequencies of durations of pauses and continuous speech. In the attack phase, we are faced with a sample of encrypted voice stream data of one of these candidates, the so-called *attack data*. We use the attack data to create a corresponding stochastic model of the pause characteristics of this candidate; after that, we employ standard and application-specific classifiers to perform a goodness-of-fit test between the stochastic models of each candidate and the one of the target candidate. The goodness-of-fit test determines the candidate whose model best matches the model derived from the attack data, yielding our guess for the target's identity.

We implemented our approach and conducted a series of experiments to evaluate its effectiveness. Our data set was composed of about 200 recorded speeches of 20 politicians. (We chose this data set because many speeches of all candidates are freely available on the web.) We encoded these speeches using the VAD-enabled voice codec Speex [36]. Speex is used by popular VoIP applications, such as Google Talk [12], TeamSpeak [30], Ekiga [9] and Microsoft Netmeeting [20]. We built the speaker models using the lengths of the plain (unencrypted) packets delivered by Speex. Our experiments showed that omitting encryption in building up these models does not affect our results, since there are large differences in length between packets corresponding to pauses and packets corresponding to speech (around a factor of six). These differences are not obscured by the largely length-preserving ciphers used for VoIP (which have a typical overhead of less than 10%). Our results show that – even in the presence of encryption – the information that VAD reveals is a serious threat to the identity of the speakers: In about 48% of all cases, we were able to correctly identify the speaker from the set of candidates.

1.2 Related work

Most documented side-channel attacks against VoIP [35, 34, 17] target *variable bit-rate* (VBR) encoding. VBR is a bandwidth-saving encoding technique that is more specialized (and thus less often implemented) than VAD. When encoding a speech segment using VBR, the codec determines whether a high bit-rate is required for encoding the segment with sufficient audio quality, or if a lower bit-rate is already enough. The bit-rate used for encoding is reflected in the size of the resulting packets and is revealed despite encryption. The resulting patterns can be observed in the traffic of an encrypted VBR-encoded voice stream, as shown in Figure 2. For a comparison of VAD and VBR, observe that a VBR codec utilizes the lowest available bit-rate for encoding pauses; hence our attack against VAD also applies to VBR-codecs. However, in contrast to attacks against VBR, our attack also poses a threat in scenarios where pure VAD is used, e.g., in mobile communication.



(a) Output of Speex codec in VAD mode

(b) Output of Speex codec in VBR mode

Fig. 2. Patterns in network traffic introduced when using a VAD-enabled codec (above) and a VBR-enabled codec (below). Each audio packet is depicted as a 0.0145s long colored bar, a lighter bar corresponding to a smaller bit-rate.

Wright et al. [35] exploit the patterns in encrypted VBR-encoded VoIP conversations to reveal which language is being spoken. In subsequent work [34] they even show that a sophisticated analysis of encrypted VBR-encoded VoIP traffic allows an attacker to (partially) uncover spoken phrases in a conversation. In contrast, our attack targets anonymity, i.e., it allows one to unveil the identity of the speaker.

Khan et al. [17] show how to reveal speaker identities from encrypted VoIP traffic. They exploit patterns introduced by VBR encoding, which is a much richer data source than the VAD encoding used in our work. Their speaker models are built using triples of packets and capture time intervals of approximately $60ms$, whereas our speaker models are based on triples of voice-/pause segments and capture time intervals of up to multiple seconds. What is particularly interesting is that the identification rates obtained by Khan et al. (75% for 10 speakers and 51% for 20 speakers) are comparable to ours (65% for 13 speakers and 48% for 20 speakers), even though their work is based on VBR instead of VAD.

In independent and concurrent work, Lu [19] considers the anonymity of speakers in encrypted VoIP communications. In contrast to our approach, she uses Hidden Markov Models (HMMs) for classification. The observed identification rates seem to be comparable to ours, however, her paper does not contain sufficient detail to allow for an in-depth comparison.

In the field of speaker recognition, there has been significant research on so-called *temporal features*, which include pause duration and frequency; phone, segmental, and word durations [11, 23]; and patterns in the interaction between speakers, e.g., durations of turns in a conversation [23, 25]. Besides speaker recognition, temporal features have been considered for other types of speaker classification. Pause duration and frequency, as well as syllable, consonant, vowel and sub-phonemic durations have been used for determining a speaker's age [26, 27, 18]. Word durations have been used for determining stress levels of a speaker [14]. Pauses in speech have also been used to identify deception [4].

For completeness, we briefly mention other known security issues in mobile and VoIP scenarios. Most importantly, a large number of weaknesses have been found in the underlying, often proprietary cipher algorithms (A5/1-3) that are intended to ensure the secrecy of transmitted data in the GSM standard [5, 3, 8, 7]. Moreover, there are a variety of known attacks against VoIP systems, e.g., denial of service attacks [37] and billing attacks [38]. We refer to [10] for a recent survey on such attacks.

1.3 Outline

The remainder of the paper is structured as follows. Section 2 explains how the speakers models are built. Section 3 introduces several measures for goodness-of-fit, i.e., for comparing speaker models. We present the empirical evaluations of our attack in Section 4 before we conclude in Section 5.

2 Building speaker profiles

In this section, we describe how a stochastic model of pause characteristics is built from the stream of packets that is delivered by a VAD-enabled voice codec. To this end, we distinguish speech from pauses by the duration of the packet. Short packets are assumed to be pauses. We then transform the packet sequence into a simpler *abstract voice stream* that retains only the lengths of runs of speech and pauses, where length is measured in terms of packet numbers. From these abstract voice streams we construct basic speaker profiles by determining the relative frequency of each pause duration. Lastly, we refine the basic speaker profiles by incorporating context information and applying clustering techniques.

2.1 Abstract voice streams

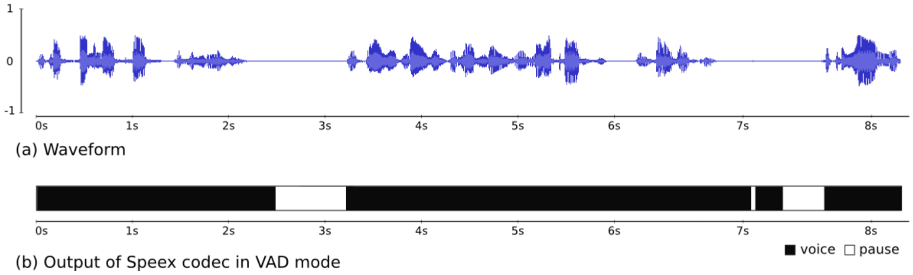
We assume that a voice stream is given as a sequence $v = [p_1, \dots, p_n]$ of packets delivered by a VAD-enabled codec. As a first step, we transform the packet sequence v into an *abstract voice stream* $abs(v)$, which is the sequence of natural numbers that correspond to the maximal numbers of adjacent packets in v corresponding to pauses, speech phases, pauses, etc. For this, we assume a threshold packet size t that distinguishes between pause and speech packets; i.e., a packet p with $|p| \leq t$ is classified as a pause packet, and as a speech packet otherwise.[3] We formalize $abs(v)$ by the following recursive definition, where + denotes list concatenation.

$$\begin{aligned} abs([]) &:= [] \\ abs([p_1, \dots, p_m] + w) &:= [m] + abs(w) \end{aligned}$$

where m is the largest integer with

$$\forall i \in \{1, \dots, m\} : \ |p_i| > t \quad \text{or} \quad \forall i \in \{1, \dots, m\} : \ |p_i| \leq t \ .$$

[3] For example, using the Speex codec, the length of speech packets exceeds the length of pause packets by a factor of 6, and it is thus easy to find a suitable threshold t.



(a) Waveform

(b) Output of Speex codec in VAD mode

Fig. 3. Correlation between the audio signal (top), and the size of packets (bottom).

For example, for a sequence of packets v of sizes $[40, 45, 41, 2, 2, 3, 2, 50, 43]$ and a threshold of $t = 3$, we obtain the abstract voice stream $abs(v) = [3, 4, 2]$, which models the lengths of maximal sequences of speech, pause, and speech packets, respectively, in v. We will assume for simplicity that each abstract voice stream begins and ends with a speech phase, i.e., a stream has an odd length and the entries at odd positions correspond to speech phases.

For each entry d of an abstract voice stream we obtain the duration of the corresponding (speech or pause) phase in real time by dividing d by the packet frequency f, i.e., the number of packets per second. For the example of the Speex codec, we have a packet frequency of $f = 69\,s^{-1}$.

2.2 Adapting abstract voice streams to the codec's characteristics

We have established a direct connection between packet numbers and durations of pause and speech phases using abstract voice streams. However, in order to capture the codec's characteristics, we must consider further extensions. Many codecs use a technique called *hangover* to avoid end-clipping of speeches and to bridge short pause segments such as those due to stop consonants [16]. When silence is detected, codecs delay for a *hangover period* before switching to pause mode. This delay can be either fixed or dynamic: dynamic hangover hardly influences the distribution of pause segments [16], and fixed hangover reduces the duration of pauses and increases the duration of voice segments. The hangover can be easily determined by comparing abstract voice streams to corresponding waveforms. In our experiments, we observed that the Speex codec uses a fixed hangover period of approx. 300ms ($\approx$ 21 packets) before switching to pause mode. (This can be seen at 2.5 sec. in Figure 3.) When the voice-signal starts again, there is (obviously) no delay for switching back to voice mode. (This can be seen at 3.2 sec. in Figure 3.)

Another artifact of the Speex codec can be observed when a very short noise signal occurs during a longer pause. In this case the codec immediately switches to voice, but switches back to pause mode earlier than after a long voice segment. (This can be seen at 7.1 sec. in Figure 3.)

To account for this codec behavior, we modify the abstract voice stream $abs(v) = [d_1, \ldots, d_k]$ as follows. First, we estimate the hangover period h; for Speex we obtain $h = 21$ packets; then

1. For each even i (corresponding to a pause), set $d_i := d_i + h$.
2. Update the previous (speech) entry d_{i-1} as follows. If $d_{i-1} > h$ then $d_{i-1} := d_{i-1} - h$. If $d_{i-1} \leq h$ then $d_{i-2} := d_{i-2} + d_{i-1} + d_i$ (ignore this assignment for $i = 2$), and delete the entries d_i and d_{i-1} from the sequence.

Modification (1) compensates for the hangover period h, as explained above. Modification (2) shortens the preceding speech entry accordingly and at the same time removes short noise signals from the stream.

Our experiments confirm that the resulting abstract voice stream more accurately captures the duration of pauses in the original voice stream, and we use it as the basis of our speaker profiles.

2.3 Basic speaker profiles

A basic speaker profile captures information about the relative frequencies of lengths of pauses or voice segments, respectively. As our experiments confirm, this information alone allows for good recognition results.

For an abstract voice stream $[d_1, \ldots, d_k]$, the relative frequency S^{pause} of pause durations d is defined as

$$S^{\text{pause}}[d] = \frac{\#\{j \mid d_{2j} = d\}}{(k-1)/2} \ .$$

Analogously, we define the relative frequency S^{voice} of the durations d of speech phases:

$$S^{\text{voice}}[d] := \frac{\#\{j \mid d_{2j+1} = d\}}{(k+1)/2} \ .$$

Given an abstract voice stream with packet lengths $[5, 10, 4, 7, 5, 7, 3]$ we obtain

$$\begin{array}{lll} S^{\text{pause}}[7] = \frac{2}{3} & S^{\text{pause}}[10] = \frac{1}{3} & \\ S^{\text{voice}}[3] = \frac{1}{4} & S^{\text{voice}}[4] = \frac{1}{4} & S^{\text{voice}}[5] = \frac{1}{2} \end{array}$$

By definition, S^{pause} and S^{voice} vanish for all packet sizes that do not occur in the abstract voice stream.

2.4 Advanced speaker profiles

The basic speaker profiles S^{pause} and S^{voice} presented above capture the relative frequencies of durations of pauses and continuous speech, respectively. As pauses and speech are considered in isolation, these models are oblivious of the context in which a pause or a speech phase occurs. To overcome this limitation, we construct a speaker profile based on the relative frequencies of three-tuples of durations of adjacent pause-voice-pause phases. By considering such three-tuples, we incorporate interdependencies between sequences of pauses and speech into our model, which captures the context in which pauses occur.

We formally define S^3 as follows

$$S^3[(x,y,z)] = \frac{\#\{j \mid d_{2j-1} = x, d_{2j} = y, d_{2j+1} = z\}}{(k-1)/2} \ .$$

It is straightforward to generalize S^3 to arbitrary n-tuples. In our experiments, however, speaker profiles based on three-tuples have proven sufficient.

2.5 Clustering

The distributions of pause and voice durations are characteristic for a speaker. However, as with most natural processes, they are subject to small random disturbances. We therefore group the pause lengths to clusters: Given a sequence $[d_1, \ldots, d_k]$ we create a clustered version (with cluster-size s) of this sequence as

$$[\lceil d_1/s \rceil, \ldots, \lceil d_k/s \rceil] \ .$$

Unless otherwise specified, in the remainder of this paper we use a cluster-size of 80 (determined experimentally to yield good results). Applying this technique has the additional advantage of reducing the support of the distribution function. This is particularly relevant for the S^3 speaker model, as its support grows cubically in the number of observed durations.

3 Measuring distance of speaker profiles

This section introduces three classifiers that serve as *goodness-of-fit* tests in this work; i.e., they compare how well the probability distribution over segment durations of the unknown speaker matches distributions over durations collected in the training phase. Thus these classifiers constitute tools to identify the victim of our attack from a set of candidate speakers.

3.1 The L_1-distance

The simplest distance measure is the metric d_{L_1} induced by the L_1-norm. For probability distributions P, Q with finite support T, the metric d_{L_1} is defined as the sum of the absolute differences between the values of P and Q, i.e.,

$$d_{L_1}(P,Q) = \sum_{x \in T} |P[x] - Q[x]| \ .$$

Even though d_{L_1} is a rather simple measure, it performs reasonably well on our experimental data, as shown in Section 4.

3.2 The χ^2-distance

A more sophisticated distance measure is the χ^2-distance, which is based on the χ^2-test. For two probability distributions P, Q with support T we define $d_{\chi^2}(P, Q)$ as the sum of the squared and normalized absolute differences between the values of P and Q, i.e.,

$$d_{\chi^2}(P, Q) = \sum_{x \in T} \frac{(P[x] - Q[x])^2}{Q[x]} \ .$$

Note that d_{χ^2} is not a metric in the mathematical sense, because it lacks symmetry. Besides this fact, the measure d_{χ^2} shows two main differences from the metric d_{L_1}. First, squaring the numerator for χ^2 gives more weight to large differences in the relative frequency of a given packet size. Second, dividing by the trained probability $Q[x]$ amplifies differences whenever $Q[x]$ is small, effectively giving the relative difference rather than the absolute difference. In our experiments, of the three classifiers, the χ^2-distance has shown the most robust performance.

3.3 The *K*-*S*-distance

Finally, we derived a distance measure based on the Kolmogorov-Smirnov test, which is known to outperform the χ^2 on samples that are small or that are sparsely distributed throughout a large number of discrete categories [21]. We define the *K*-*S*-distance of two probability distributions P, Q with support $T = \{t_1, \ldots, t_n\}$ and $t_i \leq t_j$ whenever $i < j$, by

$$d_{K\text{-}S}(P, Q) = \max_{l \leq n} \left\{ \left| \sum_{i=1}^{l} (P(t_i) - Q(t_i)) \right| \right\} \ .$$

The K-S test searches for the maximal difference between the cumulation of two distributions. In our experiments, the *K*-*S* distance performed well, but slightly worse than the χ^2-distance.

3.4 Classifier evaluation

Using the classifiers described above to compare the unknown speaker's model to the N trained models, we obtain a vector of scores $\langle s_1, s_2, \ldots, s_N \rangle$, s_i corresponding to the score of the unknown speaker's model when compared to the model of speaker i. From this vector, we compute the *rank*, representing the position at which the correct speaker was ranked. In case of a score tie, we take the lowest ranking position among all speakers with the same score. After t trials, we obtain the ranks $r_1, r_2, \ldots, r_t$, where r_i is the rank in the i-th trial. In the following we present several techniques for evaluating the performance of the classifiers using those values.

Identification rate The simplest evaluation metric we consider is *identification rate* (IR). It is computed as the percentage of the trials where the classifier guessed correctly the unknown speaker, i.e.,

$$\mathrm{IR} := \frac{\#\{i|r_i = 1\}}{t} \ .$$

The identification rate is an intuitive measure for the accuracy of classifiers. However, it is a quite conservative measure, as it ignores all results where the speakers are not ranked first. For our purposes we are not only interested in the best-scored speaker, but in a subset of the highest-ranked speakers. For example, if a classifier constantly gives the unknown speaker a rank of 3 out of 15 speakers, this still leaks information about the speaker's identity.

Average rank An evaluation method that takes into consideration all obtained ranks is the *average rank* (AR) over all obtained ranks, i.e.,

$$\mathrm{AR} := \sum_{i=1}^{t} \frac{r_i}{t} \ .$$

The results of this measure are very intuitive since they depict which position is output by the classifier on average; results closer to position 1 are preferred. However, as we are only interested in the few highest ranks, the use of average ranks may not be appropriate, as it puts equal weight on higher and lower ranks.

Top *x* results To overcome the shortcomings of average ranks, we could observe only the top x obtained ranks. Thus, we obtain the *top_x*-metric which measures the percentage of trials where the correct speaker was ranked x-th or better, i.e.,

$$\mathrm{top\_}x := \frac{\#\{i|r_i \leq x\}}{t} \ .$$

The plot with the rank x on the horizontal axis and the top_x metric on the vertical axis is called *cumulative match characteristic* (CMC) curve in the literature (e.g., see [22]), and we use it to illustrate the results of our experiments in Section 4.

Discounted cumulative gain Alternatively, we could use an adapted version of *discounted cumulative gain* (DCG), a scoring technique used mainly in information retrieval for rating web search engine algorithms [15].

Let for $i \in \{1, \ldots, N\}$, the relevance rel_i be defined as number of trials where the correct speaker was ranked i-th. The DCG-measure is defined as

$$\mathrm{DCG} := \sum_{i=1}^{N} \frac{rel_i}{d(i)} \ ,$$

where $d(i)$ is called *discounting function* and usually $f(i) = \log_2(i+1)$ is applied. Using this measure, top-ranked speakers will have a higher weight than lower-ranked ones, but lower ranks will still have a relevance to the final score of a classifier.

4 Experimental Evaluation

In this section we report on experimental results where we evaluate the feasibility of breaking anonymity in encrypted voice streams. We first describe our experimental setup and proceed by discussing the results we obtained using the speaker profiles and distance measures presented in the previous sections.

4.1 Experimental Setup

We use speeches of 20 different politicians as a data basis for our experiments: Among those 20 speakers, 18 speakers are male and 7 languages are spoken, English being the best represented language with 12 speakers, see Table 1. This set of voice recordings is homogeneous with respect to the setting in which the speeches were given, as they are official addresses to the nation that were broadcast on radio or television. The collected speeches are available online, e.g. on [33], [2], [31] and [32]. The length of the collected audio data per speaker varied between 47 and 114 minutes; on average we have about ten speeches per speaker. The speeches for each speaker were recorded in the course of several months or even years in multiple recording situations.

We simulate a unidirectional voice conversation by encoding the speeches using Speex (version 1.2rc1). We build our speaker models based on (sequences of) the sizes of audio packets output by Speex. These packet sizes correspond to the lengths of the speech packets in encrypted VoIP traffic, except for a constant offset. To see this, note that the encryption schemes used for VoIP are largely length-preserving. Moreover, typical protocols for transmitting audio packets on the application layer add headers of constant size, e.g., the commonly used Real-time Transport Protocol (RTP) [28]. As a consequence, the speaker models built from sequences of plain audio packets are equivalent to the models built from real VoIP traffic.

4.2 Results and Discussion

We performed our experiments on the full set of 20 speakers and on a subset of 13 speakers.[4] We divided the voice data of each speaker into two halves, each consisting of several speeches; we used the first half for training and the second half for the attack and vice versa, resulting in a total of 26 experiments with 13 speakers and 40 experiments with 20 speakers, respectively. We performed experiments with all speaker models discussed in Section 2, i.e., based on sequences

[4] The 13 speakers were our data set for the initial version of this paper [6], which we extended to 20 speakers for the final version.

Speaker	Nationality	Language	Number speeches	Duration (mm:ss)
Angela Merkel	Germany	German	15	53:53
Barack Obama	USA	English	15	68:33
Cristina Fernández de Kirchner	Argentina	Spanish	5	99:04
Dmitry Medvedev	Russia	Russian	12	66:40
Donald Tusk	Poland	Polish	10	92:38
Dwight D. Eisenhower	USA	English	7	67:28
Franklin D. Roosevelt	USA	English	4	80:38
George W. Bush	USA	English	15	50:24
Harry S. Truman	USA	English	5	60:48
Jimmy Carter	USA	English	6	47:56
John F. Kennedy	USA	English	8	47:10
Kevin Rudd	Australia	English	16	68:55
Luiz Inácio Lula da Silva	Brazil	Portuguese	7	105:27
Lyndon B. Johnson	USA	English	8	50:25
Nicolas Sarkozy	France	French	5	102:58
Richard Nixon	USA	English	6	113:43
Ronald Reagan	USA	English	12	51:06
Stephan J. Harper	Canada	English/French	13	100:07
Vladimir Putin	Russia	Russian	13	113:55
William J. Clinton	USA	English	20	82:05

Table 1. Speech data used in the experiments

of pause lengths (S^{pause}), sequences of speech lengths (S^{voice}), and three-tuples thereof (S^3). Moreover, we considered variants of each model based on clustering, and we compensated for the hangover technique used by Speex, as discussed in Section 2. As distance measures, we used the L_1 distance, the χ^2 classifier, and the Kolmogorov-Smirnov (K-S) classifier. Moreover, we evaluated and compared the performance of these classifiers when conducting those experiments, i.e., we analyzed the classifiers' identification rate (IR), average rank (AR), and the discounted cumulative gain (DCG), as discussed in Section 3.4.

For a data set of 13 speakers, we obtained the following results. Using the speaker model S^{pause}, the identification rate ranged between 26.9% and 38.5% depending on the used classifier, see Table 2(a). Using the speaker model S^{voice}, the identification rate ranged between 30.8% and 50%, see Table 3(a).

For 13 speakers, our best results were obtained using the speaker model S^3 and applying a clustering with cluster size 80 to reduce the support of the distribution function. For S^3, the identification rate ranged between 30.8% and 65.4%, as shown in Table 4(a). For comparison, observe that the probability of randomly guessing the correct speaker is $\frac{1}{13} \approx 7.7\%$, i.e., we achieve an 8.5-fold improvement over random guessing.

For a data set of 20 speakers, we obtained the following results. Using the speaker model S^3, we obtained identification rates between 22.5% and 40% de-

(a) Results with 13 speakers

Classifier	IR	AR	DCG
L_1	0.269	3.000	0.619
χ^2	0.385	2.423	0.697
K-S	0.308	3.615	0.613
Random	0.077	7	0.412
Best case	1	1	1

(b) Results with 20 speakers

Classifier	IR	AR	DCG
L_1	0.275	5.050	0.572
χ^2	0.175	4.475	0.545
K-S	0.225	5.525	0.542
Random	0.050	10.5	0.352
Best case	1	1	1

Table 2. Experimental results with different classifiers using speaker models based on pauses (S^{pause}). (IR = identification rate, AR = average rank, DCG = discounted cumulative gain)

(a) Results with 13 speakers

Classifier	IR	AR	DCG
L_1	0.500	2.808	0.729
χ^2	0.577	2.692	0.763
K-S	0.308	3.731	0.611
Random	0.077	7	0.412
Best case	1	1	1

(b) Results with 20 speakers

Classifier	IR	AR	DCG
L_1	0.425	4.675	0.652
χ^2	0.475	4.625	0.679
K-S	0.325	5.050	0.594
Random	0.050	10.5	0.352
Best case	1	1	1

Table 3. Experimental results with different classifiers using speaker models based on voice segments (S^{voice}). (IR = identification rate, AR = average rank, DCG = discounted cumulative gain)

pending on the classifier, as shown in Table 4(b). As in the setting with 13 speakers, S^3 outperforms the speaker model S^{pause}. However, as opposed to the setting with 13 speakers, we obtained the best identification rates for 20 speakers using the S^{voice} model: with this model, the identification rate ranged between 32.5% and 47.5%, see Table 3(b). The probability of randomly guessing the correct speaker is $\frac{1}{20} = 5\%$, i.e., we achieve a 9.5-fold improvement over random guessing. Although the identification rate decreases when considering 20 speakers instead of 13 (which was expected), the improvement over random guessing is almost constant for both data sets.

Our discussion so far has focused on identification rate as a metric for evaluating classifiers. The reason for choosing identification rate is its direct and intuitive interpretation. The results of evaluating classifiers and speaker models using different metrics are also given in Tables 2, 3, 4, and Figure 4, respectively. We believe that these alternative metrics are relevant in terms of their security interpretation. For example, the top_x-metric seems to be closely connected to the notion of anonymity sets of size x [24]. We leave a thorough investigation of this connection to future work.

(a) Results with 13 speakers

Classifier	IR	AR	DCG
L_1	0.654	2.692	0.789
χ^2	0.615	2.192	0.801
K-S	0.308	3.731	0.615
Random	0.077	7	0.412
Best case	1	1	1

(b) Results with 20 speakers

Classifier	IR	AR	DCG
L_1	0.300	4.725	0.619
χ^2	0.400	4.050	0.670
K-S	0.225	5.075	0.547
Random	0.050	10.5	0.352
Best case	1	1	1

Table 4. Experimental results with different classifiers using speaker models based on three-tuples of pauses and voice segments (S^3). (IR = identification rate, AR = average rank, DCG = discounted cumulative gain)

5 Conclusion

Performance-enhancing techniques such as voice activity detection create patterns in the volume of telephone traffic that are observable by eavesdroppers even if the traffic is encrypted. In turn, these patterns reveal patterns of pauses in the underlying voice stream. We have developed a novel approach for unveiling the identity of speakers who participate in encrypted voice communication: we show that these patterns are characteristic for different speakers, and that they are sufficient to undermine the anonymity of the speaker in encrypted voice communication. In an empirical setup with 20 speakers our analysis is able to correctly identify an unknown speaker in about 48% of all cases. This raises serious concerns about the anonymity in such conversations and is particularly relevant for communication from mobile and public devices.

References

1. 3GPP. The 3rd Generation Partnership Project. `http://www.3gpp.org/`.
2. Administration of the President of the Russian Federation. Videoblog of the President of the Russian Federation. `http://blog.kremlin.ru/`.
3. E. Barkan, E. Biham, and N. Keller. Instant ciphertext-only cryptanalysis of GSM encrypted communication. *Journal of Cryptology*, 21(3):392–429, 2008.
4. S. Benus, F. Enos, J. Hirschberg, and E. Shriberg. Pauses in deceptive speech. In *Proc. of ISCA 3rd International Conference on Speech Prosody*, 2006.
5. A. Biryukov, A. Shamir, and D. Wagner. Real time cryptanalysis of A5/1 on a PC. In *Fast Software Encryption (FSE)*, pages 1–18, Berlin, Heidelberg, 2000. Springer.
6. G. Doychev. Speaker recognition in encrypted voice streams, Bachelor's thesis, Department of Computer Science, University of Saarland, Saarbrücken, Germany, December 2009.
7. O. Dunkelman, N. Keller, and A. Shamir. A practical-time attack on the A5/3 cryptosystem used in third generation GSM telephony. Cryptology ePrint Archive, Report 2010/013, 2010. `http://eprint.iacr.org/`.

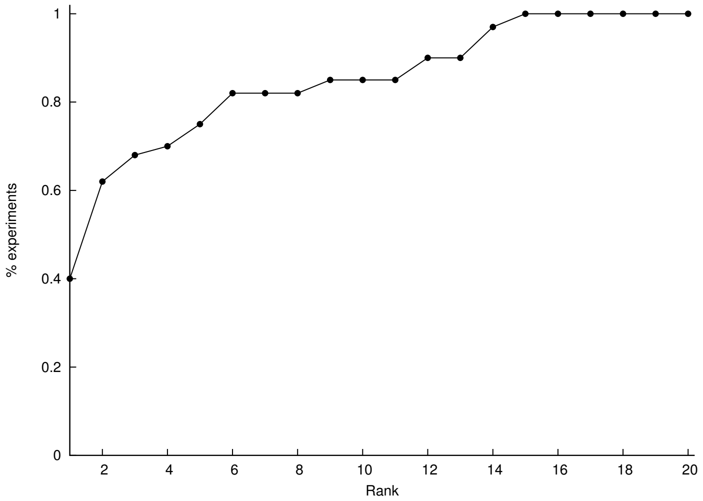


Fig. 4. Cumulative match characteristic (CMC) curve for the χ^2 classifier with 20 speakers, using speaker models based on three-tuples (S^3): the horizontal axis depicts the rank i assigned by the classifier; the vertical axis denotes the percentage of experiments in which the correct speaker was assigned at least rank i.

8. P. Ekdahl and T. Johansson. Another attack on A5/1. *IEEE Transactions on Information Theory*, 49(1):284–289, 2003.
9. Ekiga. `http://ekiga.org/`.
10. F. El-Moussa, P. Mudhar, and A. Jones. Overview of SIP attacks and countermeasures. In *Information Security and Digital Forensics*, LNICST, pages 82–91, Berlin, Heidelberg, 2010. Springer.
11. L. Ferrer, H. Bratt, V. R. R. Gadde, S. Kajarekar, E. Shriberg, K. S. Andreas, and S. A. Venkataraman. Modeling duration patterns for speaker recognition. In *Proc. of the EUROSPEECH*, pages 2017–2020, 2003.
12. Google Inc. Google Talk. `http://www.google.com/talk/`.
13. GSM-Association. GSM - Global System for Mobile communications. `http://www.gsmworld.com/`.
14. J. H. Hansen and S. Patil. Speech under stress: Analysis, modeling and recognition. In *Speaker Classification I: Fundamentals, Features, and Methods*, pages 108–137. Springer, Berlin, Heidelberg, 2007.
15. K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proc. of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48, New York, NY, USA, 2000. ACM Press.
16. W. Jiang and H. Schulzrinne. Analysis of on-off patterns in VoIP and their effect on voice traffic aggregation. In *Proc. of the 9th International Conference on Computer Communications and Networks (ICCCN 2000)*, pages 82–87, 2000.
17. L. Khan, M. Baig, and A. M. Youssef. Speaker recognition from encrypted VoIP communications. *Digital Investigation*, In Press, 2009.
18. S. E. Linville. *Vocal Aging.* Singular, 2001.
19. Y. Lu. On traffic analysis attacks to encrypted VoIP calls. Master's thesis, Cleveland State University, Nov. 2009.

20. Microsoft Corporation. Microsoft Netmeeting. http://www.microsoft.com/downloads/details.aspx?FamilyID=26c9da7c-f778-4422-a6f4-efb8abba021e&displaylang=en.
21. B. Mitchell. A comparison of chi-square and Kolmogorov-Smirnov tests. *Area*, 3:237–241, 1971.
22. H. Moon and P. J. Phillips. Computational and performance aspects of PCA-based face-recognition algorithms. *Perception*, 30:303–321, 2001.
23. B. Peskin, J. Navratil, J. Abramson, D. Jones, D. Klusacek, D. A. Reynolds, and B. Xiang. Using prosodic and conversational features for high-performance speaker recognition. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, pages 792–795, 2003.
24. A. Pfitzmann and M. Köhntopp. Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology. In *Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *LNCS*, pages 1–9, Berlin, Heidelberg, 2000. Springer.
25. D. Reynolds, J. Campbell, B. Campbell, B. Dunn, T. Gleason, D. Jones, T. Quatieri, C. Quillen, D. Sturim, and P. Torres-Carrasquillo. Beyond cepstra: Exploiting high-level information in speaker recognition. In *Proc. of the Workshop on Multimodal User Authentication*, pages 223–229, Santa Barbara, Calif, USA, December 2003.
26. S. Schötz. Acoustic analysis of adult speaker age. In *Speaker Classification I: Fundamentals, Features, and Methods*, pages 88–107. Springer, Berlin, Heidelberg, 2007.
27. S. Schötz and C. Müller. A study of acoustic correlates of speaker age. In *Speaker Classification II*, pages 1–9. Springer, Berlin, Heidelberg, 2007.
28. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications, 1996.
29. Skype Limited. Skype. http://www.skype.com/.
30. TeamSpeak Systems GmbH. TeamSpeak. http://www.teamspeak.com/.
31. The American Presidency Project. Audio/Video Archive. http://www.presidency.ucsb.edu/media.php.
32. The Press and Information Office of the German Federal Government. Podcasts. http://www.bundeskanzlerin.de/Webs/BK/De/Aktuell/Podcasts/podcast.html.
33. The White House. Your weekly address. http://www.whitehouse.gov/briefing-room/weekly-address.
34. 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 *Proc. of the 2008 IEEE Symposium on Security and Privacy*, pages 35–49. IEEE Computer Society, 2008.
35. 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 *Proc. of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 1–12. USENIX Association, 2007.
36. Xiph.Org. Speex: A free codec for free speech. http://speex.org/.
37. G. Zhang, S. Ehlert, T. Magedanz, and D. Sisalem. Denial of service attack and prevention on SIP VoIP infrastructures using DNS flooding. In *Proc. of 1st international conference on principles, systems and applications of IP telecommunications*, pages 57–66. ACM, 2007.
38. R. Zhang, X. Wang, X. Yang, and X. Jiang. Billing attacks on SIP-based VoIP systems. In *Proc. of the first USENIX workshop on Offensive Technologies*, pages 1–8. USENIX Association, 2007.