7-2012

On Privacy of Encrypted Speech Communications

Ye Zhu
Cleveland State University, y.zhu61@csuohio.edu

Yuanchao Lu
Cleveland State University, y.lu@csuohio.edu

Anil Vikram
Cleveland State University, a.vikram@csuohio.edu

Original Citation

Ye Zhu, Yuanchao Lu and A. Vikram, "On Privacy of Encrypted Speech Communications," Dependable and Secure Computing, IEEE Transactions on, vol. 9, pp. 470-481, 2012.

On Privacy of Encrypted Speech Communications

Ye Zhu, *Member*, *IEEE*, Yuanchao Lu, and Anil Vikram

Abstract—Silence suppression, an essential feature of speech communications over the Internet, saves bandwidth by disabling voice packet transmissions when silence is detected. However, silence suppression enables an adversary to recover talk patterns from packet timing. In this paper, we investigate privacy leakage through the silence suppression feature. More specifically, we propose a new class of traffic analysis attacks to encrypted speech communications with the goal of detecting speakers of encrypted speech communications. These attacks are based on packet timing information only and the attacks can detect speakers of speech communications made with different codecs. We evaluate the proposed attacks with extensive experiments over different type of networks including commercial anonymity networks and campus networks. The experiments show that the proposed traffic analysis attacks can detect speakers of encrypted speech communications with high accuracy based on traces of 15 minutes long on average.

Index Terms—Traffic analysis, speaker detection, RTP.

1 Introduction

In this paper, we investigate privacy leakage through silence suppression, an essential feature for speech communications over the Internet. Speech communications over the Internet are needed in a wide variety of Internet applications such as audiocast [1], e-learning, Internet radio, online chat, online gaming, and VoIP telephony. To save bandwidth used for speech communications, the silence suppression technique is designed to disable transmissions of speech packets when silence is detected.

The increasing popularity of speech communications over the Internet has brought a lot of attention and concern over security and privacy issues of these speech communications. To protect confidentiality of speech communications, tools and protocols such as Zfone [2], a tool capable to encrypt voice packets, and SRTP [3], the secure version of the realtime transport protocol (RTP), are developed or implemented. To further protect privacy of speech communications, advanced users are using anonymity networks to anonymize speech communications. For this purpose, low-latency anonymity networks such as Tor [4] and JAP [5] can be used. One of the common anonymizing techniques used in anonymity networks is rerouting which usually routes packets through a randomly selected and usually longer path instead of the shortest path.

In this paper, we propose a class of passive traffic analysis attacks to compromise privacy of encrypted speech communications. The procedure of the proposed attacks is as follows: First an adversary collects traces of encrypted speech communications made by a speaker, say Alice. The adversary then extracts application-level features of Alice's speech communications and trains a Hidden Markov Model (HMM) with the extracted features. To test whether one speech communication of interest is made by Alice, the adversary can extract features from the trace of interest and calculate the likelihood of the speech communication being made by Alice. The proposed attacks can detect speakers of encrypted speech communications with high probabilities. In comparison with traditional traffic analysis attacks, the proposed traffic analysis attacks are different in the following aspects: 1) The proposed traffic analysis attacks do not require *simultaneous* accesses to one traffic flow of interest at both sides. 2) The attacks can detect speakers of encrypted speech communications made with different codecs.

The major difference between the new attacks and previous traffic analysis attacks on speech communications over the Internet is: Previous traffic analysis attacks are based on packet size information and the new attacks are based on packet timing information only. In this paper, we assume packet size information is not available for traffic analysis attacks because 1) voice packets generated by constant bit rate (CBR) codecs are of the same size, 2) encryption can pad voice packets to the same size during the encryption process, and 3) packets in anonymity networks such as Tor [4] are of the same size to prevent traffic analysis attacks based on packet size information.

The contributions made in this paper are summarized as follows:

- We propose a class of traffic analysis attacks to compromise privacy of encrypted speech communications. The attacks are passive and based on the HMM, a powerful tool to model temporal data. We also propose a method to extract application-level features from traffic flows for application-level traffic analysis attacks.

- *Y. Zhu is with the Department of Electrical and Computer Engineering, Cleveland State University, 2121 Euclid Ave, SH 433, Cleveland, OH 44113. E-mail: y.zhu61@csuohio.edu.*
- *Y. Lu is with the Department of Electrical and Computer Engineering, Cleveland State University, 1655 Hannum Dr, Streetsboro, OH 44241. E-mail: y.lu@csuohio.edu.*
- *A. Vikram is with the Department of Electrical and Computer Engineering, Cleveland State University, 1131 W 12th St, Tempe, AZ 85281. E-mail: a.vikram@csuohio.edu.*

TABLE 1
Major Parameters of G.729B Silence Detector

Parameter	Meaning	Default
Min Threshold	Frame energy below which any signal is considered silence	-55 dB
Silent Threshold	Threshold used in detecting silence in signals	Dynamic
Hangover Time	Delay of silence decision	Dynamic

- We evaluate the proposed traffic analysis attacks through extensive experiments over the Internet and commercial anonymity networks. For most of speech communications made in the experiments, the two communication parties are at least 20 hops away and the end-to-end delay is at least 80 ms. Our experiments show that the traffic analysis attacks are able to detect speakers of encrypted speech communications with high probabilities based on only a small amount of encrypted traffic.
- We propose intersection attacks to improve the effectiveness of the proposed traffic analysis attacks.
- We discuss possible countermeasures to mitigate the proposed traffic analysis attacks and analyze the effect of the countermeasures on the quality of speech communications.

The rest of the paper is organized as following: Section 2 reviews speech coding and silence suppression in speech communications. In Section 3, we formally define the problem. The details of the proposed traffic analysis attacks are described in Section 4. In Section 5, we evaluate the effectiveness of the proposed traffic analysis attacks with experiments on commercial anonymity networks. Section 6 describes speaker detection without the candidate pools. In Section 7, we discuss possible countermeasures to mitigate the proposed traffic analysis attacks. Section 8 reviews the related work. We conclude the paper in Section 9.

2 Background

In this section, we review the key principles in speech coding and the silence suppression technique related to both speech communications and the proposed traffic analysis attacks.

2.1 Speech Coding

In speech communications, an analog voice signal is first converted into a voice data stream by a chosen codec. Typically in this step, compression is used to reduce the data rate. The voice data stream is then packetized in small units of typically tens of milliseconds of voice, and encapsulated in a packet stream over the Internet. In this paper, we focus on constant bit rate codecs since most codecs used in current speech communications are CBR codecs.[1]

2.2 Silence Suppression

Silence suppression, also called voice activity detection (VAD), is designed to further save bandwidth. The main idea of the silence suppression technique is to disable voice packet transmissions when silence is detected. To prevent the receiving end of a speech communication from suspecting that the speech communication stops suddenly, comfort noise is generated at the receiving end. Silence suppression is a general feature supported in codecs, speech communication software, and protocols such as RTP.

A silence detector makes voice-activity decisions based on the voice frame energy, equivalent to average voice sample energy of a voice packet. If the frame energy is below a threshold, the voice detector declares silence. Traditional silence detectors [7] use fixed energy thresholds. Because of the changing nature of background noise, adaptive energy thresholds are used in modern silence detectors such as NeVoT SD [9] and G.729B [10]. The major parameters of the G.729B silence detector, one of the most popular silence detectors, are listed in Table 1.

Hangover techniques are used in silence detectors to avoid sudden end-clipping of speeches. During *hangover time*, voice packets are still transmitted even when the frame energy is below the energy threshold. Traditional silence detectors use fixed-length hangover time. For modern silence detectors such as G.729B, the length of hangover time dynamically changes according to the energy of previous frames and noise.

Fig. 1 shows an example of the silence suppression. Fig. 1a shows the waveform of a sheriff's voice signal extracted from a video published at cnn.com [11]. Fig. 1b shows the packet train generated by feeding the voice signal to X-Lite [12], a popular speech communication tool. From Fig. 1, we can easily observe the correspondence between the silence periods in the voice signal and the gaps in the packet train. The length of a silence period is different from the length of the corresponding gap in the packet train because of the hangover technique.

The proposed traffic analysis attacks exploit the silence suppression technique. Different speakers have different talk patterns in terms of talk spurts and silence gaps. For

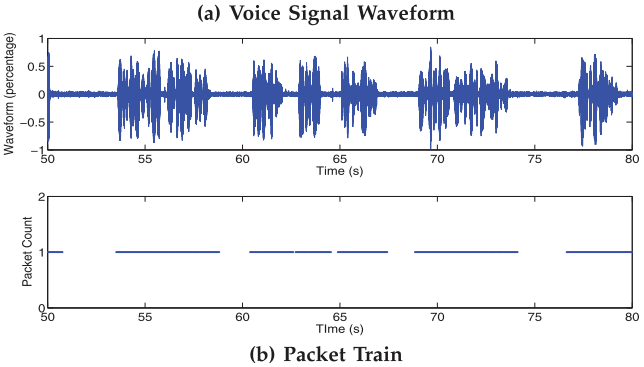


Fig. 1. An example of silence suppression.

1. Variable bit rate (VBR) codecs are primarily used for coding audio files instead of real-time speech communications [6], [7]. Recently, there are interests in using VBR codec such as Speex [8] for speech communications. But majority of existing Internet applications uses CBR codecs for speech communications. We believe the proposed traffic analysis attacks can also be launched against speech communications using VBR codecs since silence suppression is a general feature of speech codecs.

example, some speakers speak very fast with only a couple of short silence gaps while some speak with long silence gaps. As shown in Fig. 1, an eavesdropper can learn a speaker's talk pattern from the packet timing. Based on talk patterns learned from packet timing, the proposed traffic analysis attacks can detect speakers of encrypted speech communications with high accuracy.

3 Problem Definition

In this paper, we are interested in analyzing the traffic of encrypted speech communications through anonymity networks. We focus on detecting speakers of encrypted speech communications by analyzing talk patterns, the application-level patterns recovered from network traffic.

Speaker detection aims to detect the speaker of one specific encrypted speech communication such as the presenter of a presentation through audiocast or the instructor of an e-learning course. For simplification, we assume that the speaker of interest is Alice. To launch speaker detection attacks, the adversary collects traces of Alice's previous encrypted speech communications in advance so that he or she can detect whether Alice is the speaker in one specific encrypted speech communication by comparing the trace of the specific speech communication with the traces of Alice's previous speech communications.

In this paper, we assume: 1) The traces used in detection can be collected at different time and in different network environments. 2) Speech communications of interest can be possibly made with different codecs.

3.1 Network Model

We assume Alice's speech communications are encrypted with one of the secure versions of the RTP protocol such as SRTP [3] or ZRTP used in Zfone [2] to protect confidentiality of her speech communications.

To better protect privacy of her speech communications, we assume Alice routes her encrypted speech communications through anonymity networks. For better voice quality, Alice can use low-latency anonymity networks[2] such as Tor [4] and JAP [5].

3.2 Threat Model

We focus on passive attacks in this paper. In other words, the attacks launched by the adversary will not disturb existing network traffic. In comparison with active attacks, the proposed attacks are harder to be detected.

We assume that the adversary can access the links connected to the callers to collect traces used for training. This assumption is widely used in traffic analysis attacks such as attacks on anonymity networks and tracing VoIP calls [13], [14], [17], [18], [15]. The threat model is weaker than the threaten models defined for traditional privacy-related traffic analysis attacks: The threat model does not require *simultaneous* access to the links connected to participants of speech communications. Instead, we assume the adversary can collect Alice's speech communication traces in advance and use these collected traces to detect whether Alice is the speaker of the speech communication of interest.

Our model is similar as the model for identifying a human being by fingerprints: Fingerprints of human beings are collected in advance by various means such as collecting finger prints during the driver license applications. To identify a specific person, the fingerprint of interest such as a fingerprint in a crime scene will be compared against the person's fingerprints collected in advance.

As described in Section 1, we assume the packet size information is not available for traffic analysis because of the CBR codecs and the packet encryption. Since packet encryption also prevents access to packet content by an adversary, only packet timing information is available to an adversary to launch privacy attacks.

3.3 Formal Definition

The goal of the proposed traffic analysis attacks is to identify Alice's speech communication trace from a pool of candidate traces including the trace of Alice's speech communication. We define the pool size as the number of candidate traces in the pool.[3] The performance of the speaker identification can be evaluated with detection rate, false positive rate, and false negative rate. We define the *detection rate* as the ratio of the number of successful detections to the number of attempts. One detection is defined as successful if Alice's trace is correctly identified from the pool and we defined attempt as one trial of the identification. The *false negative rate* is defined as the proportion of Alice's speech communications that are detected as speech communications made by other speakers. The *false positive rate* is defined as the proportion of speech communications made by other speakers that are detected as Alice's speech communications.

4 Detecting Speakers of VoIP Calls

In this section, we describe the traffic analysis attacks to detect speakers of encrypted speech communications. We begin the section with an overview of the proposed traffic analysis attacks and then proceed with the details of each step in the attacks.

4.1 Overview

The proposed traffic analysis attacks are based on packet timing information only. As described in Section 2.2, the silence suppression technique enables adversaries to recover talk patterns in terms of talk spurts and silence gaps from packet timing. Adversaries can create a Hidden Markov Model to model Alice's talk pattern recovered from her encrypted speech communications. When adversaries want to determine which trace of encrypted speech communications in a pool of candidate traces is made by Alice, adversaries can check talk patterns recovered from the candidate traces against Alice's model.

2. Despite of the existing traffic analysis attacks [13], [14], [15], [16], low-latency anonymity networks can still effectively protect communication confidentiality and anonymity.

3. The model without the assumption of the pool and the corresponding performance evaluation are described in Section 6.

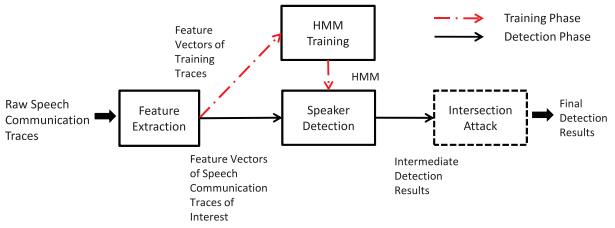


Fig. 2. Steps of the proposed attack.

The proposed attacks can be divided into two phases: the training phase and the detection phase as shown in Fig. 2. The two steps in the training phase are feature extraction and HMM training. The detection phase consists of three steps: feature extraction, speaker detection, and intersection attack. The last step, the intersection attack, is optional. We describe the details of each step below.

4.2 Feature Extraction

The input and the output of the feature extraction step are raw traces of encrypted speech communications and feature vectors, respectively. The feature vector used in the proposed attacks is shown below

$$\begin{bmatrix} ts_1 & ts_2 & \cdots & ts_{n\psi} \\ sg_1 & sg_2 & \cdots & sg_n \end{bmatrix}, \psi$$

where $ts_{i\psi}$ and $sg_{j\psi}$ denote the length of the ith talk spurt and the jth silent gap, respectively, and $n\psi$ is the length of a feature vector.

Talk spurts and silent gaps are differentiated by a silence threshold $T_{silence\psi}$. If an interpacket time (IPT) is larger than the threshold, the IPT is declared as a silence gap. Otherwise the IPT is declared as a part of one talk spurt.

Obviously the threshold $T_{silence\psi}$ is critical to the overall detection performance. Our initial experiments focus on the suitable range of the threshold for detection: We code voice signals with different codecs and collect generated voice packets. Different values of the threshold $T_{silence\psi}$ are used to determine silence gaps. Actual silence gaps can be found by checking the marker bits in RTP packets which indicate the beginnings of talk spurts.[4] We evaluate a value of the threshold by two metrics: false positive rate and false negative rate. The *false positive rate of the silence test* is the fraction of talk spurts that are erroneously declared as silence gaps. The *false negative rate of the silence test* is the fraction of silences gaps that are erroneously declared as talk spurts. The experiment results with different codecs[5] are shown in Fig. 3.

We can observe that for a wide range of the threshold $T_{silence\psi}$ both the false positive rate and the false negative rate are low: When $T_{silence\psi}$ is larger than 70 ms, the false positive rates are below 10 percent for all the codecs. The false negative rates are below 20 percent when $T_{silence\psi}$ is less than 100 ms. The range of the threshold $T_{silence\psi}$ suitable for silence tests is wide because of the big difference between

4. Only in our initial experiments, voice packets are not encrypted so that we can determine actual silence gaps from marker bits and then find the suitable range of the threshold for detection. For all the other experiments, voice packets are encrypted and the proposed traffic analysis attacks have no access to packet headers such as the marker bit in the RTP protocol.

5. Details of these codecs can be found in Table 2.

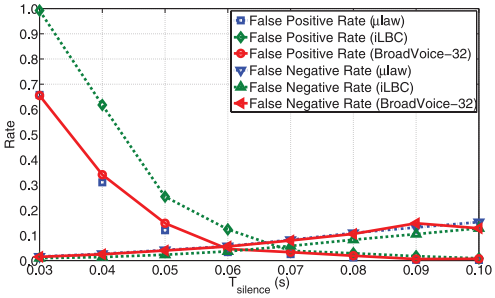


Fig. 3. Experiment results on the Threshold $T_{silence\psi}$

IPTs of silent gaps and IPTs during talk spurts: Silence gaps are in order of seconds. The length of the IPTs during talk spurts is usually close to the packetization delay of 20 or 30 ms for most codecs.

We can also observe that increasing the threshold $T_{silence\psi}$ decreases the false positive rate and increases the false negative rate. The changes in these two rates are again because the IPTs of silence gaps are longer than the IPTs during talk spurts.

A big challenge in feature extraction is to filter out the noise caused by the random network delay in the silence tests since the random network delay can cause inconsistency in the silence tests based on voice packets collected from different links on the path of a speech communication. For example, an IPT during a talk spurt at the sending side may be declared as a silence gap at the receiving side just because the random network delays can possibly cause the IPT to be larger than the threshold $T_{silence\psi}$. Obviously, the inconsistency in silence tests will in turn degrade the performance of the speaker detection based on voice packets collected from different links on the path.

The main idea of filtering the noise caused by random network delays in the silence tests is to determine a silence gap based on $N\psi$ successive IPTs instead of $one\psi$ IPT. The silence test with the filtering technique works as follows: If one IPT is larger than the threshold $T_{silence\psi}$, we declare a new silence gap only when none of the following $\lfloor \frac{T_{silence}}{packetization\,delay} \rfloor - 1$[6] IPTs are shorter than $T_{spurt\psi}$, the threshold used to filter out long IPTs caused by the network delay. The rationale behind the new silence test method is as follows: If an IPT is erroneously declared as a silence gap because the random network delay increases the length of the IPT, then the following IPTs must likely be shorter than the normal IPTs during talk spurts.

We compare the new silence test with the original silence test through empirical experiments: The two communication parties in a speech communication through the Internet are at least 20 hops away from each other. In this set of experiments, we evaluate the choices of the parameter $T_{spurt\psi}$ with the match rate $R_{match\psi}$

$$R_{match\psi} = \{\text{number-of-gaps-found-at-both-the-sending-side-and-the-receiving-side}\}/\{\text{number-of-gaps-found-at-the-sending-side}\}.\psi \quad (1)$$

6. We use $\lfloor \rfloor$ to denote the floor function.

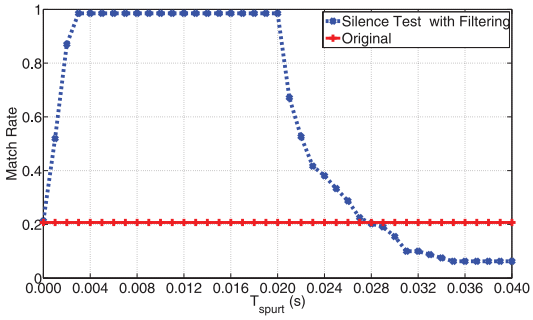


Fig. 4. Match rate versus Threshold $T_{spurt\psi}$(μlaw-Codec).

Ideally, the match rate $R_{match\psi}$should be 1 meaning that silence gaps detected at the sending side can match silence gaps detected at the receiving side exactly. The experiment results are shown in Fig. 4.

Fig. 4 shows that the filtering technique can significantly increase the match rate $R_{match\psi}$. The original silence test can only achieve a match rate of 0.21. The silence test with the filtering technique can achieve a match rate of 0.99 when $T_{spurt\psi}$is between 3 and 20 ms. The match rate is low when $T_{spurt\psi}$is less than 0.3 ms because the threshold $T_{spurt\psi}$is too low to filter out large IPTs caused by network delays. The match rate is also low when $T_{spurt\psi}$is larger than 20-ms. It is because normal IPTs during talk spurts are of 20-ms for the μlaw-codec and a threshold $T_{spurt\psi}$larger than 20-ms filters out most of actual silence gaps. In following experiments, we set the threshold $T_{spurt\psi}$to be 10-ms.

Feature vectors generated in this step are used for training or detection in future steps.

4.3 HMM Training

The input and the output of this step are feature vectors and trained HMMs, respectively.

The Markov Model is a tool to model a stochastic process with the Markov property that the transition from the current state to the next state depends only on the current state and not on the past states. In a Hidden Markov Model, the state is not directly visible, but outputs influenced by the state are observed. Each state has a probability distribution over the possible output. Therefore, the sequence of outputs generated by an HMM gives some information about the sequence of states. The HMM is a well-known tool to model temporal data and it has been successfully used in temporal pattern recognition such as speech recognition [19], handwriting recognition [20], and gesture recognition [21]. In the proposed attacks, HMMs are trained to model talk patterns used for the speaker detection.

In our paper, we consider each talk period including one talk spurt and the following silence gap as a hidden (invisible) state. The output observation from one state is the length of a talk spurt and the length of the following silence gap. Since each state corresponds to a talk period, a trace of one speech communication is a process going through these hidden states. So we use HMMs to model talk patterns. With the use of HMMs in our modeling, we assume the Markov property holds. This assumption is widely used in speech and language modeling. Even when the assumption does not hold strictly, the extended HMM can still work well [22].

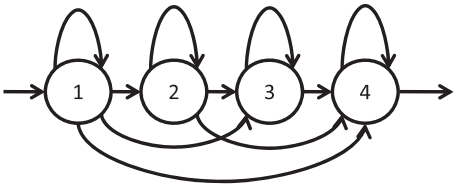


Fig. 5. HMM.

The HMM used in traffic analysis attacks is the modified left-right HMM [22] as shown in Fig. 5. It is based on left-right models because of the nonergodic nature of speech signals [22], i.e., the attribute of signals whose properties change over time. The fundamental property of all left-right HMMs is that the state transition coefficient from the ith state to the jth state (denoted as a_{ij}) is zero, when $j\psi$is less than i. Additional constraints are placed on the state transitions in left-right models to make sure that large changes in state indices do not occur, i.e., $a_{ij\psi=0}$, when $j > i \psi \Delta$. For the well-known banded left-right model [22] and Bakis model [23], Δ-is 1 and 2, respectively.

We extended classical left-right models to allow transitions from the ith state to the $(i+3)$th state, i.e., $\Delta = 3$, as shown in Fig. 5. Our modification on the left-right model is because of the possible false negative errors made in the feature extraction step and the adaptive silence thresholds used in silence detectors as described in Section 2:

False negative errors are made when some silent gaps are not detected in feature extraction. The false negative errors can be caused by a large threshold $T_{silence\psi}$or the hangover time as described in Section 2. The hangover time reduces the length of silence gaps recovered from the packet timing since speech packets are still being sent during the beginning and the end of a silence duration to avoid end clipping of speeches. The reduction can cause false negative errors in silence tests.

Adaptive silence thresholds used in silence detectors can cause different silence detection results for the same speech in different speech communications. For modern codecs, the threshold used in a silence detector dynamically changes to adapt to the changes in background noise. Because of the dynamically changing threshold, one silent gap in the same speech can be detected as silence in one speech communication or as a part of a talk spurt in another speech communication by the same codec. Although the inconsistent detection results because of the adaptive silence threshold are not very often observed, it can cause low speaker detection performance.

To take into account the possible false negative errors made in the feature extraction and the possible inconsistency of silence detectors, we allow state transitions from the ith state to the $(i+3)$th state because up to three actual neighboring talk periods can be detected as one talk period in our analysis of speech communication traces. Our experiments with different left-right models also show that the modified left-right model can achieve better detection performance than the other left-right models. In the

modified HMM, the number of states are heuristically set to be 10 according to the length of feature vectors.[7]

We use Gaussian distributions to model the observation distributions. The mean and variance of the Gaussian distributions are estimated from the training data.

In this step, a speaker-specific model can be obtained by training the HMM with traces of Alice's speech communications. The trained HMMs are used in the following speaker detection step.

4.4 Speaker Detection

The inputs to this step are the Alice's HMM trained in the previous step and the feature vectors generated from a pool of raw speech communication traces of interest. The output of this step is the intermediate detection result, i.e., $K_{top\psi}$ speakers from the candidate pool with talk patterns closest to Alice's talk pattern.

The detection step can be divided into two phases: 1) First, the likelihood of each feature vector is calculated with the trained HMM. 2) The trace with the highest likelihood is declared as Alice's trace if the intersection step is not used. To improve the detection accuracy, the intermediate detection results can be fed into the optional intersection attack step.

4.5 Intersection Attack

The intersection step is designed to improve detection accuracy. The input to this step is the intermediate detection result from the previous step. The output is a final detection result.

The main idea of the intersection attack step is similar as described in [24], [25], [26]: Instead of making a detection decision based on one trial, we can improve detection accuracy with a number of trials and the final detection result is determined by combining (or intersecting) the results from all trials.

More specifically, for the proposed attacks, suppose it is possible to get $m\psi$ speech communication traces made by the same speaker, $m\psi$ trials can generate $m\psi$ intermediate detection results as described in Section 4.4. In other words, from each trial, the adversary can obtain $K_{top\psi}$ traces with the $K_{top\psi}$ highest likelihood values. The overall rank for each speaker is calculated by adding up the ranks in the $m\psi$ trials. The speaker with the highest rank is determined to be Alice. A tie can be broken by comparing the sum of the likelihood values in the $m\psi$ trials.

In summary, the proposed traffic analysis attacks can be divided into two phases: the training phase and the detection phase. Since the attack is based on the talk patterns, the traffic analysis attacks are independent from codecs used in speech communications. In other words, it is possible to train HMMs with traces of speech communications made with one codec and then use the trained HMMs to detect speakers of speech communications made with another codec. We evaluate the proposed traffic analysis attacks with the empirical experiments described below.

7. Following the principle of Occam's razor, the number of states should be small enough to avoid overfitting and large enough to model the ergodic nature of speech communications. We get similar detection performance for different number of states when the number of states is larger than five. When the number of state is too large, the training of HMMs fails to converge to an optimal solution.

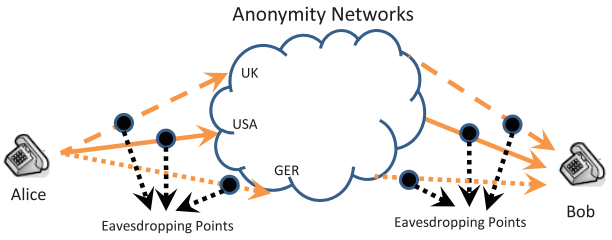


Fig. 6. Experiment Setup.

5 Empirical Evaluation

In this section, we evaluate the effectiveness of the proposed traffic analysis attacks with empirical experiments.

5.1 Experiment Setup

The experiment setup is as shown in Fig. 6. Speech packets are first directed to the anonymity network managed by findnot.com before arriving at the receiving side. We use the commercial anonymous communication services provided by findnot.com[8] mainly because it is possible to select entry points into the anonymity network [28]. In our experiments, speech packets are directed through entry points in England, Germany, and United States as shown in Fig. 6. For these speech communications made through the anonymity networks, the end-to-end delay is at least 80 ms and the two communication parties are at least 20 hops away from each other. About a quarter of the speech communications are made through the campus network so that traces of speech communications over different type of networks are available for our experiments.

The audio signals are extracted from videos hosted on Research Channels [29] and these audio signals can be downloaded from [30]. Traces used in both training and detection are 14.7 minutes long on average if not specified.[9] At least three different speeches are available for each speaker and the speeches are sent through four different network entry points.[10]

We choose three popular and representative codecs of high, medium, and low bit rates for our experiments. More information about these three codecs is listed in Table 2.

For better training, all the traces used in training are collected on the link connected to Alice's computer. The traces used in the detection phase can be collected on any link in the path from the sending side to the receiving side. The timing of packets collected on the link connected to the receiving side is usually the noisiest due to the accumulated randomness of the network delay. If not specified, the traces used in the detection phase are all collected on the links connected to the receiving side.

8. We did not use Tor [4] to anonymize speech communications because Tor has no direct support of anonymizing UDP packets and in general, speech communication packets are sent as UDP packets. Wang et al. [27] experimented on the anonymous communication services provided by find.com instead of Tor for the same reason.

9. For fair comparison, traces used in experiments should contain the same number of talk periods. In other words, feature vectors generated from these traces should be of the same size. Because of the difference in the length of talk periods in different traces, traces used in experiments are of different length in minutes.

10. The campus network entry point is one of the choices.

TABLE 2
Codec Information

Codec	Sampling Frequency (kHz)	Frame Size (ms)	Bit Rate (Kbit/s)	Payload Size (bits)	Packetization Delay (ms)
μlaw	8 (Narrowband)	10	64	1280	20
iLBC	8 (Narrowband)	20/30	15.2/13.3	304/400	30
BroadVoice-32	16 (Wideband)	5	32	160*n	20

5.2 Metrics

We use detection rate to measure the effectiveness of the proposed attacks. In this paper, the detection rate is defined as the ratio of the number of successful detections to the number of attempts.

For speaker detection with traces generated by the same codec, the detection rate for a random guess is about $\frac{1}{109}$, because in each trial, there are around 109 candidate traces in the pool if the pool size is not specified. One of the traces in the pool is the "right" trace, i.e., Alice's trace. In each trial of speaker detection, one trace of Alice's speech is used as one of the candidate traces and Alice's traces generated from Alice's other speeches are used for training.

If not explicitly specified, all detection rates reported in this section are averaged over experiments of all possible combinations of the training traces and the candidate traces. In all the experiments below, the training traces and the candidate traces are all collected from *different* speech communications.

5.3 Threshold $T_{silence\psi}$

This series of experiments are designed to test the effect of the parameter $T_{silence\psi}$ the threshold used in silence tests.

Fig. 7 shows the speaker detection performance with different values of the threshold $T_{silence\psi}$ Each detection rate in Fig. 7 is obtained based on 120 trials with 109 traces in the candidate pool. The length of traces used in both training and detection is 14.7 minutes on average.

From Fig. 7, we can observe: 1) The detection rate for speaker detection can reach 0.32, about 35-fold improvement over a random guess, when the size of the candidate pool is 109. 2) In general, the detection rate increases when the threshold $T_{silence\psi}$increases. When $T_{silence\psi}$becomes large, the detection rate may drop simply because shorter feature vectors are used for training and detection. When $T_{silence\psi}$is larger than 0.512-s, feature vectors are too short for detection so that the HMM training cannot converge for certain traces. 3) The detection rate for candidate traces collected from the sending end is comparable with the detection rate for candidate traces collected from the receiving end. It is because the filtering technique used in the silence test can largely filter out noise caused by the random network delay at the receiving end. In the following experiments, we set $T_{silence\psi}$to be 0.412 seconds.

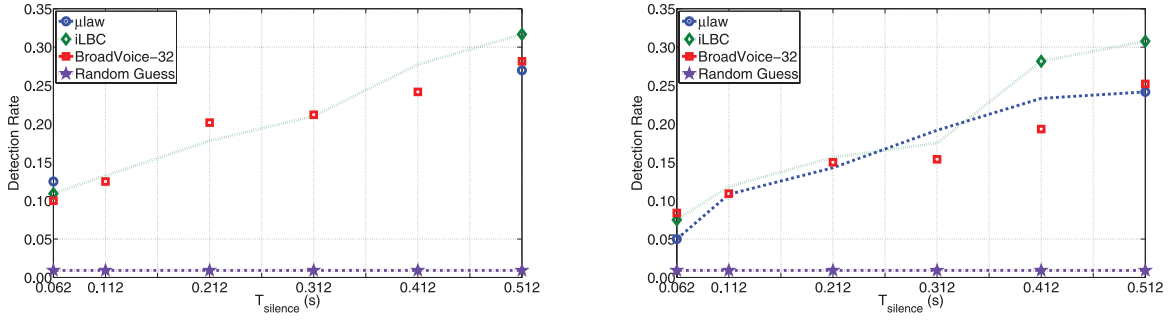
5.4 Length of Training and Test Traces

This set of experiments is designed to investigate the effect of the length of the training and the test traces on the detection performance. Since in general, training traces should be longer than test traces for better training, we vary the average length of the training traces from 5.4 to 14.7 minutes and the average length of the test traces varies from 1.9 minutes to the average length of the training traces used in the same detection.

From experiment results shown in Fig. 8, we can observe that even for five-minute-long training and test traces, the detection rate for the speaker detection can achieve 0.12, about 13-fold improvement over a random guess. Fig. 8 also shows that the detection rate increases with the length of training traces and the length of test traces. In the following experiments, we fix the average length of the training traces and the test traces to be 14.7 minutes.

5.5 Pool Size

In this set of experiments, we investigate the detection performance with different sizes of the candidate pool. From the experiment results shown in Fig. 9, we can observe that when the pool size increases, the detection rate slightly decreases for all the codecs. The reason is that it is harder to find the right one from a larger pool. But the ratio between the speaker detection rate and the detection rate of a random guess changes from 12 to 37, when the pool size changes from 28 to 109. It means that the traffic analysis attacks are more effective when the pool size is larger. We can also observe that for the μlaw-codec, one of the most frequently used codecs in speech communications, the detection rate can reach 0.42 when the pool size is 28, approximately 37-fold improvement over a random guess.



(a) Candidate Traces Collected from Sending Side

(b) Candidate Traces Collected from Receiving Side

Fig. 7. Speaker detection performance with different Threshold $T_{silence\psi}$

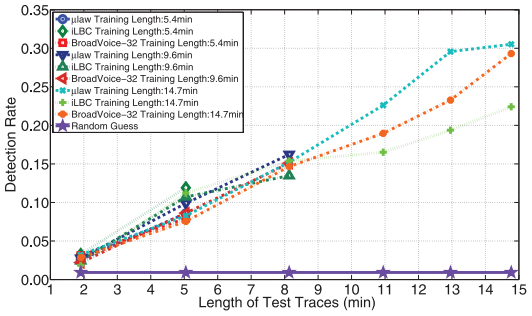


Fig. 8. Speaker detection performance with different length of training traces and test traces.

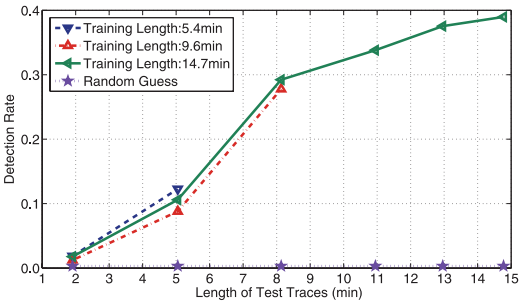


Fig. 10. Cross-codec detection performance with different length of training trace and test traces.

5.6 Cross-Codec Detection

In this set of experiments, the training traces and the traces to be detected are generated with different codecs. We believe this set of experiments is important because: 1) Practically training traces and the traces to be detected can be collected from speech communications made with different codecs. 2) Since speech packets are encrypted and possibly padded to a fixed length, adversaries may not be able to differentiate speech communications made with different codecs.

In this set of experiments, the threshold $T_{silence\psi}$is set to 0.412 s. If the size of the candidate pool is not specified, then there are around 325 candidate traces in the pool including the "right" trace. So the detection rate for a random guess is about $\frac{1}{325}$. In each trial of the speaker detection, one trace of Alice's speech is used as one of the candidate traces and Alice's traces generated from Alice's other speeches are used for training. If the length of the traces used for training and detection are not specified, then the length of these traces is 14.7 minutes on average.

Fig. 10 shows the detection performance with different length of training traces and test traces. We can again observe that the detection rate increases with the length of training traces and test traces. The detection rate of the speaker detection with only five minutes of training traces and test traces can reach 0.12, about 40-fold improvement over a random guess.

Fig. 11 shows the detection performance with different sizes of the candidate pool. We can observe that the detection rate decreases slightly with the increase of pool size. When the pool size is 82, the detection rate can reach 0.60. By comparing the performance results shown in Fig. 9 with the performance results shown in Fig. 11, we can also observe that even for larger pool sizes, the detection rate for the cross-codec detection is higher than the single-codec detection. It is mainly because more traces are available for training HMMs in the cross-codec detection and only traces made by the same codec are available for training in the single-codec detection.

5.7 Intersection Attack

In this set of experiments, we evaluate the effectiveness of the intersection attacks on the cross-codec speaker detection. On average, there are 37 candidate speakers in each trial. So the detection rate for a random guess is about $\frac{1}{37}$. Each candidate speaker has nine traces available for detection. So the final detection result is obtained by combining the intermediate detection results of nine trials. In this set of experiments, the length of traces used for training and detection is 14.7 minutes long on average and $T_{silence\psi}$=0.412 s.

Table 3 shows the performance of the intersection attack: First, the intersection attacks greatly improve the performance of the cross-codec speaker detection. Second, the detection rate can reach 0.625, about 25-fold improvement over a random guess.

In summary, the proposed traffic analysis attacks can detect speakers of encrypted speech communications with high accuracy based on traces of about 15 minutes long on average.

6 Detecting Speaker without Candidate Pools

The initial threat model assumes that the "right" speaker is in the candidate pool. Although the assumption is valid for applications similar as identifying a human being with a group of fingerprints collected from a crime scene, we would like to investigate the detection performance without the assumption of the candidate pool. Instead, we

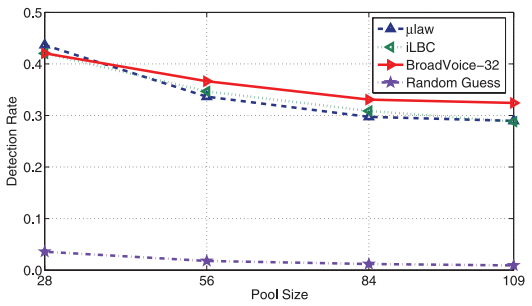


Fig. 9. Detection performance with different pool sizes.

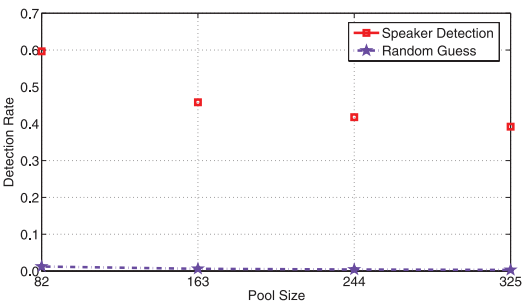


Fig. 11. Cross-codec detection performance with different pool sizes.

TABLE 3
Performance of Intersection Attacks Combined with Cross-Codec Speaker Detection

$T_{silence}$ (ms)	$K_{top}=1$	$K_{top}=2$	$K_{top}=4$	$K_{top}=8$
212	0.4250	0.3750	0.3750	0.4000
412	0.5500	0.4750	0.6000	0.6250

assume that the adversary possesses traces of speech communications made by Alice and other speakers. We call these traces as labeled traces since these traces are collected in advance and the adversary knows the identities of the speakers. The goal of the adversary is to detect whether Alice is the speaker of a speech communication of interest.

6.1 Detection Approach

We modify the detection approach for the new traffic analysis attack as follows:

1. The adversary splits the labeled traces of Alice's speech communications into two halves. An HMM to model Alice's talk pattern is established based on the first half of the traces.
2. A detection threshold $T_{lik\psi}$ is determined based on the remaining labeled traces including the second half of the labeled traces of Alice's speech communications. The adversary evaluates each of these traces against Alice's model and calculates its likelihood. Given a threshold T_{lik}, the adversary calculates the false positive rate and the false negative rate on the remaining labeled traces as follows: a) False negative rate is defined as the proportion of Alice's speech communications detected as speech communications made by other speakers, i.e., the proportion of Alice's speech communications with likelihood values less than T_{lik}. b) False positive rate is defined as the proportion of speech communications made by other speakers detected as Alice's speech communications, i.e., the proportion of other speakers' traces with likelihood values larger than $T_{lik\psi}$ The threshold $T_{lik\psi}$ is selected so that the detection rates on the remaining labeled traces are maximized and both the false negative rate and the false positive rate on the remaining labeled traces are below a tolerance threshold T_{tol}.
3. The adversary makes a detection decision by evaluating a given trace with Alice's HMM. If the calculated likelihood is larger than T_{lik}, the given trace is declared as Alice's trace. Otherwise, the trace is declared as a trace made by another speaker.

6.2 Performance Evaluation

We evaluate the detection performance with four metrics: detection rate, false negative rate, false positive rate, and percentage of traces which can be tested. The two metrics, the false negative rate and the false positive rate used in performance evaluation, are calculated on the test traces. The last metric, percentage of traces which can be tested, is needed because for certain group of labeled traces, it is impossible to find a threshold $T_{lik\psi}$ so that both the false negative rate and the false positive rate on the labeled traces are below a given tolerance $T_{tol\psi}$

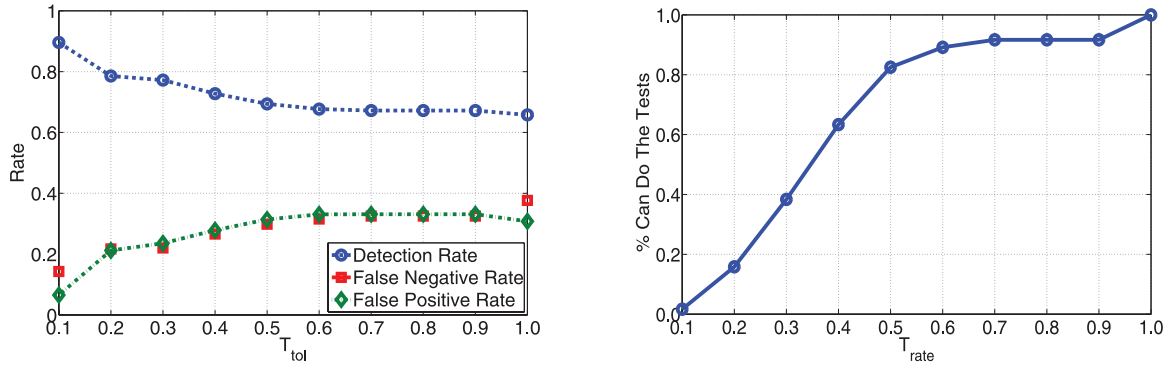
In this set of experiments, the average length of labeled traces and test traces are 14.7 minutes. In each detection attempt, there are 54 labeled traces and six Alice's traces. The experiment results are averaged over 120 tests.

Experiment results shown in Fig. 12 indicate that the detection rate decreases when the tolerance $T_{tol\psi}$ increases and in the mean time, the percentage of trace which can be tested increases. A smaller tolerance $T_{tol\psi}$ means better training, and in turn, better detection performance. A smaller tolerance $T_{tol\psi}$ also means stricter requirements so fewer traces can be tested. We can also observe that the detection rate can reach 0.89 when $T_{tol\psi}=0.1$ and only one percent traces can be tested. When $T_{tol\psi}=1$, i.e., all the traces can be tested, the detection rate is 0.63.

7 Discussion

From the experiment results shown above, it is apparent that the proposed traffic analysis attacks can greatly compromise the privacy of encrypted speech communications. Countermeasures are needed for privacy protection. In this section, we discuss possible countermeasures which can protect privacy with only marginal effect on the quality of service (QoS) of speech communications.

Simple countermeasures to the proposed traffic analysis attacks include padding speech traffic to constant rate traffic or randomly delaying speech packets to hide talk patterns. These simple approaches may render the proposed traffic



(a) Detection Rate

(b) Percentage of Test Traces Which Can Be Tested

Fig. 12. Detection performance.

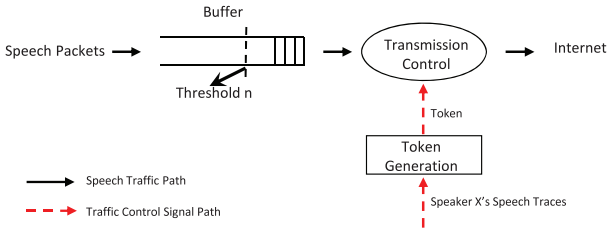


Fig. 13. Countermeasure: Camouflaging Alice's speech communication.

analysis attacks ineffective. But these approaches can cause significant waste of bandwidth or degrade the QoS of speech communications significantly.

A better approach is to camouflage the timing of Alice's speech packets according to another speaker's trace. As shown in Fig. 13, Alice's speech packets are first kept in a buffer. A token will be generated when it is time to send a packet according to Speaker X's trace. The transmissions of Alice's speech packets are controlled by these tokens. The transmission control in Fig. 13 functions as follows:

1. Each packet transmission consumes a token.
2. When a token is generated and the buffer is not empty, the transmission control will transmit the first packet in the buffer.
3. When a token is generated and the buffer is empty, a dummy packet is transmitted by the transmission control.
4. When $N_{buffψ}$ packets are held in the buffer and no token is available, the first packet in the buffer will be transmitted.

For the proposed countermeasure, dummy packets are sent only when necessary for camouflaging. The parameter $N_{buffψ}$ is used to control the queuing delay. This parameter should be carefully chosen to balance the QoS of speech communications and the privacy protection to defeat traffic analysis attacks.

The two metrics used in our initial analysis of the countermeasure are: 1) The detection rate defined in Section 5.2: It is used to measure the performance of the privacy protection of speech communications. 2) Additional packet delay caused by the countermeasure: It measures the degradation of the QoS of speech communications.

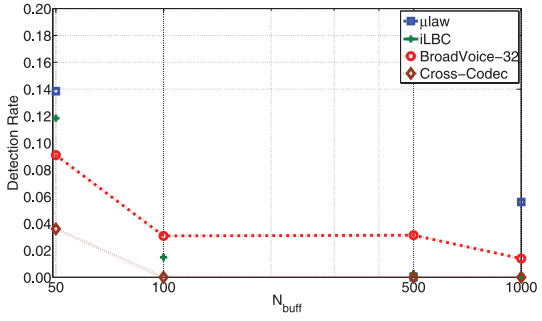
In this set of experiments, we use real speech communication traces collected from the experiment environment described in Section 5.1.

Fig. 14 shows the performance of the countermeasure. The threshold $T_{silenceψ}$ is set to 0.412 s in this series of experiments. Fig. 14a shows that the countermeasure can protect the privacy of speech communications since most detection rates are around the probability of a random guess. Fig. 14b shows the additional packet delays caused by the countermeasure. When $N_{buffψ}$ is 50 and 100, the additional delays caused by the countermeasure are less than 36 and 68 ms with a probability larger than 0.95, respectively. So the countermeasure will not cause any significant change in the quality of speech communications since the additional delays for $N_{buffψ}=50$ and $N_{buffψ}=100$ are still less than one third of and half of the delay budget for speech communications [31], respectively. The detection rates for small $N_{buffψ}$, such as $N_{buffψ}=1$ and $N_{buffψ}=10$, cannot be obtained from experiments, because $N_{buffψ}$ is too small and no silence gaps can be found in the speech communication traces.

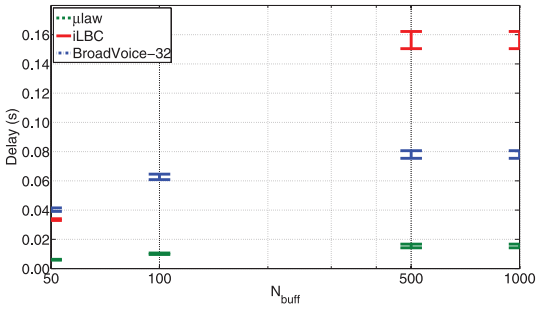
8 Related Work

In this section, we review privacy-related traffic analysis attacks at the application level and describe the relation between the proposed attack based on packet timing and side channel attacks on a cryptosystem to recover a cryptographic key.

The application-level traffic analysis attacks target at disclosing sensitive information at the application level. Song et al. [32] found that despite encryption and authentication mechanisms used in SSH, it is possible to obtain interkeystroke timing information from SSH packets since SSH sends out each keystroke in one separate packet during the interactive mode. Based on the interkeystroke timing information, they demonstrated that it was possible to reveal passwords used in SSH logins. Sun et al. [33] gave a quantitative analysis for identifying a webpage even if encryption and anonymizing proxies are used. They took advantage of the fact that a number of HTTP features such as the number and size of objects can be used as signatures to identify webpages with some accuracy. Unless the anonymizer addresses this, these signatures are visible to the adversary. Herrmann et al. [34] proposed to identify websites by applying common text mining techniques to the normalized frequency distribution of observable IP packet sizes. Lu et al. [35] showed the feasibility of website fingerprinting based on packet ordering information.



(a) Detection Rate



(b) Additional Packet Delays with 95% Confidence Interval

Fig. 14. Performance of the possible countermeasure.

Wright et al. [36] showed packet size information of VoIP packets can be used by an adversary to identify a spoken phrase in VoIP calls. In [37], it was shown that packet size information of VoIP packets could also be used to detect languages used in conversations even the conversations were encrypted.

The application-level traffic analysis attacks can be classified into two categories based on features of the network traffic used in these attacks. Most existing application-level traffic analysis attacks are based on packet size information [33], [34], [36], [35]. Wright et al. [38] proposed approaches to counter traffic analysis attacks on VoIP calls and their approaches are based on modifying packet sizes. Only a few application-level traffic analysis attacks are based on packet timing only. One example is the keystroke detection based on SSH packets [32].

The traffic analysis attacks proposed in this paper are based on packet timing information only since 1) CBR codecs generate voice packets of the same size and 2) encryption can easily pad voice packets to the same size during the encryption process. The countermeasure discussed in Section 7 protects communication privacy by modifying packet timing so that the original talk patterns can be camouflaged.

Concurrently with our research, Backes et al. [39] proposed an approach to detect speakers by measuring distance between distributions of silence gaps and talk spurts and the comparable detection performance is reported in [39]. The major differences between our traffic analysis attacks and the attacks proposed by Backes et al. [39] are: 1) Our attacks can filter out the noise caused by the random network delay so that the traces to be detected can be collected even from the last hop of a speech communication path. Extensive experiments show that the attacks are effective even for international speech communications routed through the commercial anonymity networks. 2) Our attacks are effective to compromise privacy of encrypted speech communications with different codecs and we evaluate the attacks with the cross-codec detection experiments. The capability of cross-codec detection is desired since encrypted speech communications prevent attackers from knowing codecs in use and different codecs are used in different network settings.

Conceptually the timing-based traffic analysis attacks proposed in this paper are similar as classical timing attacks [40], [41], [42], [43] on cryptosystems, one type of side-channel attacks. In Kocher's seminal paper [40], it was shown that the timing information of private key operations can be used to compromise secret keys. In this paper, similarly we show that timing information of encrypted speech communication systems can be used to compromise privacy of encrypted speech communications. More specifically, the proposed traffic analysis attacks use the packet timing information to detect speakers of encrypted speech communications. A slight difference between these two types of timing attacks is: Usually classical timing attacks on cryptosystems are implementation-specific, i.e., exploiting the implementation of a cipher instead of the internals of a cipher. The timing attacks proposed in this paper exploit silence suppression, an essential feature of speech communications to save bandwidth.

9 Conclusions

In this paper, we propose a class of passive traffic analysis attacks to compromise privacy of speech communications. The proposed attacks are based on application-level features extracted from speech communication traces. We evaluated the proposed attacks by extensive experiments over different types of networks including commercial anonymity networks and the campus network. The experiments show that the proposed traffic analysis attacks can detect speakers of encrypted speech communications with high detection rates based on speech communication traces of 15 minutes long on average.

Acknowledgments

The authors are grateful to the reviewers for their valuable comments that helped to improve the revised version. This work was supported in part by the US National Science Foundation (NSF) under grant No. 1144644. Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] S. Casner and S. Deering, "First ietf Internet Audiocast," *SIGCOMM Computer Comm. Rev.*, vol. 22, pp. 92-97, http://doi.acm.org/10.1145/142267.142338, July 1992.

[2] P. Zimmermann, A. Johnston, and J. Callas, "Zrtp: Media Path Key Agreement for Secure rtp Draft-Zimmermann-Avt-Zrtp-11," RFC, United States, 2008.

[3] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, "The Secure Real-Time Transport Protocol (srtp)," 2004.

[4] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," *Proc. 13th USENIX Security Symp.*, pp. 303-320, Aug. 2004.

[5] O. Berthold, H. Federrath, and S. Köpsell, "Web MIXes: A System for Anonymous and Unobservable Internet Access," *Proc. Designing Privacy Enhancing Technologies: Workshop Design Issues in Anonymity and Unobservability*, H. Federrath, ed., pp. 115-129, July 2000.

[6] J.M. Valin, "Speex: A Free Codec for Free Speech," *Proc. Australian Nat'l Linux Conf.*, 2006.

[7] P.T. Brady, "A Technique for Investigating on-off Patterns of Speech," *The Bell System Technical J.*, vol. 44, pp. 1-22, 1968.

[8] speex.org, "The Speex Projectpage,"http://www.speex.org, 2005.

[9] S. Henning, "Voice Communication Across the Internet: A Network Voice Terminal," COINS technical report, Dept. of Computer and Information Science, Univ. of Massachusetts at Amherst, 1992.

[10] ITU-T Study Group 15, *Coding of Speech at 8kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction Annex b: A Silence Compression Scheme for g.729 Optimized for Terminals Conforming to Recommendation v.70.Recommendation g.729b Telecomm. Standardization Sector of itu*, Int'l Telecomm. Union Std., 1996.

[11] cnn.com, "Police Reveal the Identity of Shooting Suspect," http://www.cnn.com/2006/US/09/29/school.shooting/index.html, 2011.

[12] "X-Lite 3.0 Free Softphone," http://www.xten.com/index.php?menu= Products&smenu=xlite, 2011.

[13] B.N. Levine, M.K. Reiter, C. Wang, and M.K. Wright, "Timing Attacks in Low-Latency Mix-Based Systems," *Proc. Eighth Int'l Financial Cryptography (FC '04) Conf.*, pp. 251-265, Feb. 2004.

[14] S.J. Murdoch and G. Danezis, "Low-Cost Traffic Analysis of Tor," *Proc. IEEE Symp. Security and Privacy*. May 2005.

[15] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, "Correlation-Based Traffic Analysis Attacks on Anonymity Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 21, no. 7, pp. 954 -967, July 2010.

[16] Y. Zhu and R. Bettati, "Compromising Anonymous Communication Systems Using Blind Source Separation," *ACM Trans. Information and System Security*, vol. 13, pp. 8:1-8:31, http://doi.acm.org/10.1145/1609956.1609964, Nov. 2009.

[17] X. Wang, S. Chen, and S. Jajodia, "Tracking Anonymous Peer-to-Peer Voip Calls on the Internet," *Proc. ACM Conf. Computer and Comm. Security*, pp. 81-91, Nov. 2005.

[18] Y.J. Pyun, Y.H. Park, X. Wang, D.S. Reeves, and P. Ning, "Tracing Traffic through Intermediate Hosts that Repacketize Flows," *Proc. IEEE INFOCOM '07,* May 2007.

[19] C. Rathinavelu and L. Deng, "Hmm-Based Speech Recognition Using State-Dependent, Linear Transforms on Mel-Warped dft Features," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP '96),* pp. 9-12, 1996.

[20] M.-P. Schambach, "Determination of the Number of Writing Variants with an HMM Based Cursive Word Recognition System," *Proc. Seventh Int'l Conf. Document Analysis and Recognition (ICDAR '03),* p. 119, 2003.

[21] J.W. Deng and H.T. Tsui, "An HMM-Based Approach for Gesture Segmentation and Recognition," *Proc. Int'l Conf. Pattern Recognition (ICPR '00),* pp. 679-682, 2000.

[22] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE,* vol. 77, no. 2, pp. 267-296, Feb. 1990.

[23] R. Bakis, "Continuous Speech Recognition via Centisecond Acousticstates," *J. the Acoustical Soc. of Am.,* vol. 59, p. S97, 1976.

[24] O. Berthold, A. Pfitzmann, and R. Standtke, "The Disadvantages of Free MIX Routes and How to Overcome Them," *Proc. Designing Privacy Enhancing Technologies Workshop Design Issues in Anonymity and Unobservability,* pp. 30-45, July 2000.

[25] G. Danezis and A. Serjantov, "Statistical Disclosure or Intersection Attacks on Anonymity Systems," *Proc. Sixth Information Hiding Workshop (IH '04),* pp. 293-308, May 2004.

[26] O. Berthold and H. Langos, "Dummy Traffic Against Long Term Intersection Attacks," *Proc. Privacy Enhancing Technologies Workshop (PET '02),* pp. 110-128, Apr. 2002.

[27] X. Wang, S. Chen, and S. Jajodia, "Tracking Anonymous Peer-to-Peer Voip Calls on the Internet," *Proc. 12th ACM Conf. Computer and Comm. Security (CCS '05),* pp. 81-91, 2005.

[28] FindnotProxyList, http://www.findnot.com/servers.html, 2011.

[29] ResearchChannels, http://www.researchchannel.org, 2011.

[30] "Audio Signals Used for Experiments,"http://academic.csuohio.edu/zhu_y/isc2010/instruction.txt, 2011.

[31] T. Szigeti and C. Hattingh, *End-to-End Qos Network Design: Quality of Service in Lans, Wans, and vpns (Networking Technology).* Cisco Press, 2005.

[32] D.X. Song, D. Wagner, and X. Tian, "Timing Analysis of Keystrokes and Timing Attacks on ssh," *Proc. 10th Conf. USENIX Security Symp. (SSYM '01),* pp. 25-25, 2001.

[33] Q. Sun, D.R. Simon, Y.-M. Wang, W. Russell, V.N. Padmanabhan, and L. Qiu, "Statistical Identification of Encrypted Web Browsing Traffic," *Proc. IEEE Symp. Security and Privacy (SP '02),* pp. 19-30, 2002.

[34] D. Herrmann, R. Wendolsky, and H. Federrath, "Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier," *Proc. ACM Workshop Cloud Computing Security (CCSW '09),* pp. 31-42, 2009.

[35] L. Lu, E.-C. Chang, and M. Chan, "Website Fingerprinting and Identification Using Ordered Feature Sequences," *Proc. 15th European Conf. Research in Computer Security (ESORICS),* D. Gritzalis, B. Preneel, and M. Theoharidou, eds. pp. 199-214. 2010.

[36] 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," *Proc. IEEE Symp. Security and Privacy (SP '08),* pp. 35-49, 2008.

[37] C.V. Wright, L. Ballard, F. Monrose, and G.M. Masson, "Language Identification of Encrypted Voip Traffic: Alejandra y Roberto or Alice and Bob?," *Proc. 16th USENIX Security Symp. USENIX Security Symp.,* pp. 4:1-4:12, http://portal.acm.org/citation.cfm?id=1362903.1362907, 2007.

[38] C. Wright, S. Coull, and F. Monrose, "Traffic Morphing: An Efficient Defense against Statistical Traffic Analysis," *Proc. Network and Distributed Security Symp. (NDSS '09),* Feb. 2009.

[39] M. Backes, G. Doychev, M. Dürmuth, and B. Köpf, "Speaker Recognition in Encrypted Voice Streams," *Proc. 15th European Symp. Research in Computer Security (ESORICS '10),* pp. 508-523, Sept. 2010.

[40] P.C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, rsa, dss, and Other Systems," *Proc. 16th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO '96)* pp. 104-113, http://portal.acm.org/citation.cfm?id=646761.706156, 1996.

[41] J.-F. Dhem, F. Koeune, P.-A. Leroux, P. Mestré, J.-J. Quisquater, and J.-L. Willems, "A Practical Implementation of the Timing Attack," *Proc. Int'l Conf. Smart Card Research and Applications,* pp. 167-182, http://portal.acm.org/citation.cfm?id=646692.703439, 2000.

[42] W.H. Wong, "Timing Attacks on Rsa: Revealing Your Secrets through the Fourth Dimension," *Crossroads,* vol. 11, p. 5, http://doi.acm.org/10.1145/1144396.1144401, May 2005.

[43] D. Brumley and D. Boneh, "Remote Timing Attacks are Practical," *Computer Networks,* vol. 48, pp. 701-716, http://portal.acm.org/citation.cfm?id=1090583.1090585, Aug. 2005.