2009

On Traffic Analysis Attacks to Encrypted VOIP Calls

Yuanchao Lu
Cleveland State University

ON TRAFFIC ANALYSIS ATTACKS TO ENCRYPTED VOIP CALLS

YUANCHAO LU

Bachelor of Science in Electrical Engineering

Beijing University of Posts and Telecommunications

July, 2007

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

November, 2009

This thesis has been approved for the

Department of **ELECTRICAL AND COMPUTER ENGINEERING**

and the College of Graduate Studies by

Thesis Committee Chairperson, Dr. Ye Zhu

Department/Date

Dr. Vijaya K. Konangi

Department/Date

Dr. Pong P. Chu

Department/Date

To my loved husband Dawei and my entire family...

ACKNOWLEDGMENTS

I would like to thank the following people:

Dr. Ye Zhu for giving me such a challenging opportunity, for his inspiring direction for the research and for his great support in all aspects of my study.

Dr. Vijaya K. Konangi, and Dr. Pong P. Chu, who are on my committee, for their time in reviewing and evaluating this dissertation.

Dr. Chansu Yu, Dr. Wenbing Zhao, and Dr. Zhiqiang Gao for their elaborately prepared lectures which imparted knowledge to me, and their perspectives of different things that enriched my view.

Dr. Dan Simon for my improvement in English writing.

Thank you to my friends for their kind help and friendship.

ON TRAFFIC ANALYSIS ATTACKS TO ENCRYPTED VOIP CALLS

YUANCHAO LU

ABSTRACT

The increasing popularity of VoIP telephony has brought a lot of attention and concern over security and privacy issues of VoIP communication. This thesis proposes a new class of traffic analysis attacks to encrypted VoIP calls. The goal of these attacks is to detect speaker or speech of encrypted VoIP calls. The proposed traffic analysis attacks exploit silent suppression, an essential feature of VoIP telephony. These attacks are based on application-level features so that the attacks can detect the same speech or the same speaker of different VoIP calls made with different VoIP codecs. We evaluate the proposed attacks by extensive experiments over different type of networks including commercialized anonymity networks and campus networks. The experiments show that the proposed traffic analysis attacks can detect speaker and speech of encrypted VoIP calls with a high detection rate which is a great improvement comparing with random guess. With the help of intersection attacks, the detection rate for speaker detection can be increased. In order to shield the detrimental effect of this proposed attacks, a countermeasure is proposed to mitigate the proposed traffic analysis attacks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYM

VoIP Voice over Internet Protocol

PSTN Public Switched Telephone Networks

HMM Hidden Markov Model

QoS Quality of Service

RTP Real-time Transport Protocol

CBR Constant Bit Rate

VBR Variable Bit Rate

CHAPTER I

INTRODUCTION

This thesis addresses on privacy issues of encrypted Voice over Internet Protocol (VoIP) calls. A class of traffic analysis attacks is proposed to compromise the privacy of encrypted VoIP calls. In this thesis, a countermeasure is proposed and studied to the class of traffic analysis attacks.

With the rapid growth of broadband Internet access services, the popularity of VoIP calls has grown significantly. As a competitor with traditional phone services provided over Public Switched Telephone Networks (PSTN), VoIP services are known for their lower cost and richer features.

The increasing popularity of VoIP telephony has brought a lot of attention and concern over security and privacy issues of VoIP communication. To protect confidentiality of VoIP calls, advanced users encrypt VoIP packets with Zfone [1] or SRTP [2], the secure version of real time transport protocol.

To further protect privacy of VoIP calls, advanced users are using anonymity networks to anonymize VoIP calls. For this purpose, low-latency anonymity networks such as Tor [3] and JAP [4] can be used. The common anonymizing technique used

in anonymity networks is rerouting which usually routes packets through a random-selected and usually longer path instead of the shortest path.

A class of passive traffic analysis attacks is proposed to compromise privacy of encrypted VoIP calls in this thesis. The procedure of proposed attacks is as follows: First, the adversary collects traces of encrypted VoIP calls made by a victim, say Alice. The adversary then extracts application-level features of Alice's VoIP calls and trains a Hidden Markov Model (HMM) with the extracted features. To test whether a call of interest is made by Alice, the adversary can extract features from the traces of the call and calculate likelihood of the calls made by Alice. The proposed attacks can detect speeches or speakers of encrypted calls with a high probability. In this thesis, two kinds of VoIP calls are involved: Skype and SIP-Based VoIP calls. Because these two kinds of VoIP calls are two primary VoIP calls.

In comparison with traditional traffic analysis attacks, the proposed traffic analysis attacks are different in the following aspects: (a) The proposed traffic analysis attacks are at application-level, and traditional traffic analysis attacks are at transport-level or network-level: The proposed traffic analysis attack aim to detect speeches or speakers of VoIP calls, and these detections are at application-level. Traffic analysis used in these proposed attacks is based on application-level features. Most traditional traffic analysis attacks [5, 6, 7, 8] aim to identify traffic flows, and these identifications are at transport-level or network-level. These traditional traffic analyses are based on transport-level or network-level features such as correlation between traffic flows at sending end and receiving end. (b) Because of the previous difference, the proposed traffic analysis attacks are more practical: (a) The proposed traffic analysis attacks do not require *simultaneous* access to traffic flows at both ends. But the *simultaneous* access is usually required for traditional traffic analysis attacks. For international VoIP calls, simultaneous access at both ends of calls may not be possi-

ble in practice. (b) The attacks can detect the same speech or the same speaker of different VoIP calls made with different VoIP codecs and protocols.

The contributions made in this thesis are summarized as follows:

- A class of traffic analysis attacks to compromise privacy of encrypted VoIP calls is proposed. The attacks are passive and based on the HMM, a powerful tool to model temporal data. It also proposes a method to extract application level features from traffic flows for application-level traffic analysis attacks.

- The proposed traffic analysis attacks are evaluated through extensive experiments over the Internet and commercial anonymity networks. For most of VoIP calls made in the experiments, the two parties are at least 20 hops away, and end-to-end delay between the two parties is at least 80ms. The experiments show that the traffic analysis attacks are able to detect speeches or speakers of encrypted VoIP calls with a high probability.

- Intersection attacks are proposed to improve the effectiveness of the proposed attacks.

- A countermeasure is proposed to mitigate the proposed traffic analysis attacks and analyzed the effect of the countermeasure on quality of VoIP calls.

The organization of this thesis is organized as follows: Chapter 2 reviews related work. In Chapter 4, the details of proposed traffic analysis attacks on Skype VoIP calls are descried. Chapter 3 introduces the traffic analysis attacks on SIP-Based VoIP Calls. Finally, the conclusion is offered in Chapter 5.

CHAPTER II

RELATED WORK

In this chapter, we review related work on low-latency anonymity networks and related traffic analysis attacks.

2.1 Low-Latency Anonymity Networks

After Chaum proposed the anonymous communication for email in his seminal paper [9], many low-latency anonymity networks have been proposed or even implemented for different applications. The examples are *ISDN-mixes* [10] for telephony, *Web Mix* [4] for web browsing, *MorphMix* [11] for peer-to-peer applications, *GAP* base *GNUnet* [12] for file sharing. TARZAN [13], *Onion Router* [14], and *Tor* [3], the second-generation onion router, are designed for general usage by low-latency applications. Especially *Tor* has some desirable features for low-latency applications such as perfect forward secrecy and congestion control. In this thesis, we used the anonymity network managed by findnot.com to anonymize VoIP calls instead of the Tor network, because UDP traffic is not natively supported by Tor. The commercialized

anonymous communication services provided by findnot.com can allow us to route VoIP packets through entry points located in different countries into the anonymity network.

Common techniques used in low-latency anonymity networks are encryption and re-routing. Encryption prevents packet content access by adversaries. To confuse adversaries, anonymity networks using re-routing techniques forward encrypted packets in a usually longer and random path instead of using the shortest path between the sender and the receiver. To attack an anonymity network using the re-routing technique, the attacker usually needs to be more powerful, for example, to be a global attacker.

2.2 Traffic Analysis Attacks

Traffic analysis attacks can be classified into two categories, network-level traffic analysis attacks and application-level traffic analysis attacks. Network-level traffic analysis attacks target at disclosing network-level or transport-level information. Most privacy-related network-level traffic analysis attacks focus on traffic flow identification or traffic flow tracking. The examples are Levine *et al.* [5] on anonymity networks, the active attack proposed by Murdoch and Danezis [8] on Tor network, and the flow correlation attacks [6]. Application-level traffic analysis attacks target at disclosing application-level information. The examples are keystrokes detection based on packet timing [15], web page identification [16], spoken phrase identification [17] with variable bit rate codecs.

The traffic analysis attacks proposed in this thesis are at application-level. These attacks can detect speeches or speakers of encrypted VoIP calls based on talk patterns, the application-level features which do not vary from call to call.

CHAPTER III

TRAFFIC ANALYSIS ATTACKS ON SIP-BASED ENCRYPTED VOIP CALLS

In this chapter, we address on privacy issues of SIP-Based encrypted VoIP calls. A class of traffic analysis attacks is proposed to compromise the privacy of encrypted SIP-Based VoIP calls. We propose and study countermeasures to the class of traffic analysis attacks in this chapter.

The increasing popularity of VoIP telephony has brought a lot of attention and concern over security and privacy issues of VoIP communication. To protect confidentiality of VoIP calls, advanced users encrypt VoIP packets with Zfone [1] or SRTP [2], the secure version of Real-time Transport Protocol (RTP). To further protect privacy of VoIP calls, advanced users are using anonymity networks to anonymize VoIP calls. For this purpose, low-latency anonymity networks such as Tor [3] and JAP [4] 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 chapter, we propose a class of passive traffic analysis attacks to compromise privacy of encrypted VoIP calls. The procedure of proposed attacks is as follows: First an adversary collects traces of encrypted VoIP calls made by a victim, say Alice. The adversary then extracts application-level features of Alice's VoIP calls and trains a Hidden Markov Model (HMM) with the extracted features. To test whether a call of interest is made by Alice, the adversary can extract features from the trace of interest and calculate likelihood of the call being made by Alice. The proposed attacks can detect speeches or speakers of encrypted calls with high probabilities. In comparison with traditional traffic analysis attacks, the proposed traffic analysis attacks are different in the following aspects: (a) The proposed traffic analysis attacks do not require *simultaneous* access to one traffic flow of interest at both sides. (b) The attacks can detect the same speech or the same speaker of different VoIP calls made with different VoIP codecs.

3.1 Background

In this section, we first review protocols used in current VoIP communications and then proceed with review of key principles in speech coding and silence suppression related to VoIP communication and the proposed traffic analysis attacks.

In general, VoIP protocols can be classified into two categories: (a) Signaling protocols: These protocols are designed for call setup and termination. SIP [18] and H.323 [19] are two of most widely-used signaling protocols for VoIP. (b) Transport protocols: These protocols are designed to transfer voice packets. A typical example is Real-time Transport Protocol (RTP) [20]. Most current mass-market VoIP services such as Vonage [21] and AT&T CallVantage [22] use SIP and RTP as the signaling protocol and the transport protocol respectively. One exception is Skype VoIP service that uses a proprietary protocol. In this chapter, we focus on mass-market VoIP

services based on SIP and RTP protocols.

3.1.1 Speech Coding

In VoIP telephony, 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 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.

We focus on Constant Bit Rate (CBR) codecs in this project since most codecs used in current VoIP telephony are CBR codecs[1]. In this thesis, we evaluate the proposed traffic analysis attacks against codecs of various bit rates.

3.1.2 Silence Suppression

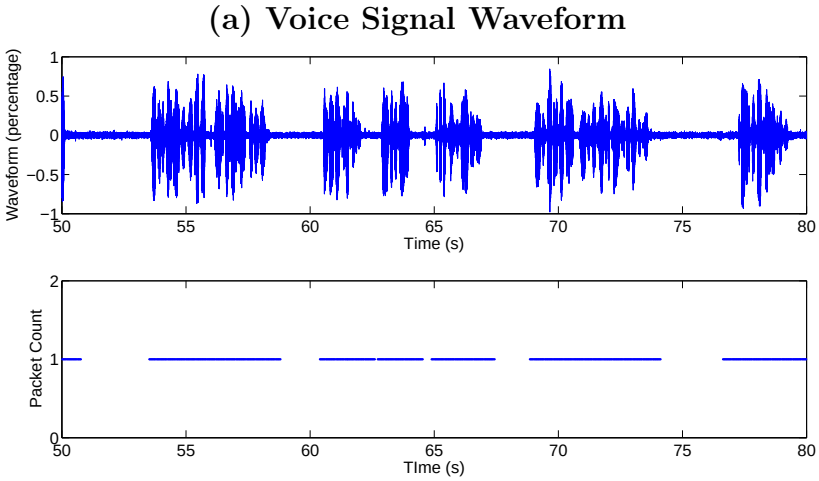
To further save bandwidth, VoIP telephony employs functionality known as silence suppression or voice activity detection (VAD). The main idea of silence suppression is to disable voice packet transmission when one of the parties involved in a call is silence. To prevent the other party from suspecting that the call is dead and then dropping the call, comfort noise is generated at the receiving side. Silence suppression is a general feature supported in codecs, VoIP softwares, and RTP.

A silence detector makes voice-activity decision based on voice frame energy, equivalent as average voice sample energy of a VoIP packet. If the frame energy is below a threshold, the voice detector declares silence. Traditional silence detectors [23] 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

[1]Variable Bit Rate (VBR) codecs are primarily used for coding audio files instead of voice communication [23]. Recently there are interests in using VBR codec such as Speex [24] for VoIP telephony. But no implementation is publicly available according to our knowledge. We believe proposed traffic analysis attacks can also be launched against VoIP services using VBR codecs since silence suppression is a general feature of VoIP codecs.

Table I: Major Parameters of G.729B Silence Detector

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



(a) Voice Signal Waveform

(b) Packet Train

Figure 1: An Example of Silence Suppression

[25] and G.729B [26]. Major parameters of G.729B silence detector, one of the most popular silence detectors, are listed in Table I.

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.

Figure 1 shows an example of silence suppression. Figure 1.(a) shows the waveform of a sheriff's voice signal extracted from a video published at cnn.com [27]. Figure 1.(b) shows the packet train generated by feeding the voice signal to X-Lite

[28], a popular VoIP client. From Figure 1, we can easily observe the correspondence between silence periods in the voice signal and gaps in the packet train. The length of a silence period will be different from the length of the corresponding gap in a packet train because of the hangover technique.

The proposed traffic analysis attacks exploit silence suppression. Different people have different talk patterns in terms of talk spurts and silence gaps. For example, some persons speak very fast with only a couple of short silence gaps while some speak with long silence gaps. As shown in Figure 1, an eavesdropper can learn a speaker's talk pattern from packet timing. Based on talk patterns learned from packet timing, the proposed traffic analysis attacks can detect speeches or speakers of encrypted VoIP calls with high accuracy.

3.2 Problem Definition

In this chapter, we are interested in analyzing the traffic of encrypted VoIP calls through anonymity networks. We focus on detecting speeches and speakers of encrypted VoIP calls by analyzing the sensitive information revealed from the traffic pattern at application-level.

The typical attack scenario focused in this chapter is as follows: An adversary may want to detect whether the target speaker, say Alice, is communicating with Bob now or not based on the previous encrypted VoIP calls made by Alice. The previous calls may use different codecs than the one Alice using now. The adversary may collect VoIP packets at any point on the path from Alice to Bob and may also want to detect the content of the conversation, such as a partial speech in previous calls.

Comparing with previous researches, the proposed attacks do not require simultaneous access to both sides of the links connected to Alice and Bob. Traces

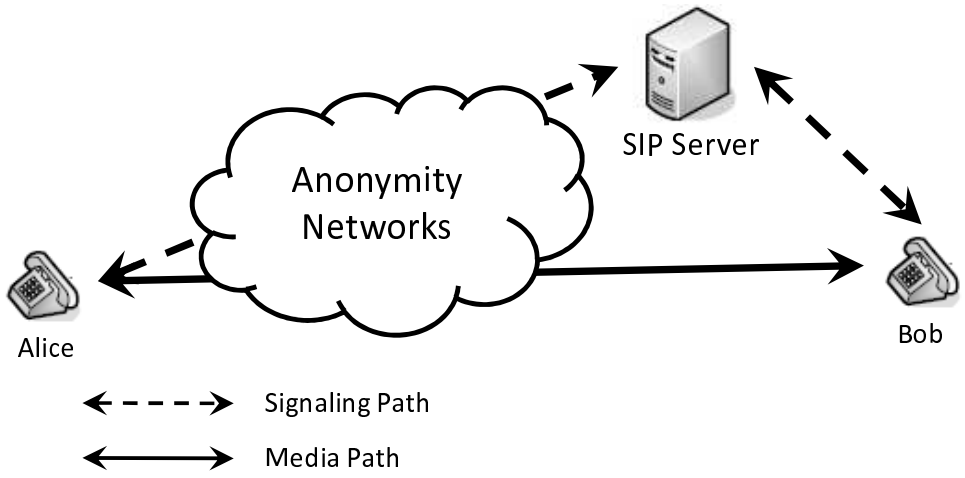


Figure 2: Network Model

of calls used in detection can be collected at different time and in different network environment and these calls possibly made with different codecs.

3.2.1 Network Model

In this chapter, we also assume Alice uses mass-market VoIP services to communicate with Bob as shown in Figure 2. In other words, we assume SIP and RTP are used as the signaling protocol and the transport protocol respectively. To protect confidentiality of her VoIP calls, we assume Alice encrypts her VoIP packets by using secure versions of the RTP protocol such as SRTP [2] and ZRTP used in Zphone [1].

To better protect privacy of her calls, we assume Alice routes these encrypted VoIP calls through anonymity networks as shown in Figure 2. For better voice quality, Alice can use low-latency anonymity networks such as Tor [3] and JAP [4].

3.2.2 Threat Model

We focus on passive attacks in this thesis. 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 detect. We assume that the adversary

only has access to the links directly connected to participants of VoIP calls. This assumption is widely used in traffic analysis attacks such as attacks on anonymity networks and tracing VoIP calls [5, 8, 29, 30]. We do not assume the adversary as a global attacker because rerouting techniques used in anonymity networks make global attacks too costly to be practical. The threat model is weaker than 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 a VoIP call which may not be feasible for international VoIP calls. Instead we assume the adversary can collect traces of VoIP calls made by Alice in advance and use these collected traces to detect whether Alice is a participant in the VoIP conversation of interest. Our model is similar as the model for identifying a human being by fingerprints: Fingerprints of human beings are collected in advance through 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.

The threat model assumes the detections are based on different VoIP calls. So the speaker identification should also be independent of the voice content of VoIP calls.

3.3 Detecting Speeches and Speakers of SIP-Based VoIP Calls

In this section, we describe traffic analysis attacks to detect speeches or speakers of SIP-Based encrypted VoIP calls. We begin the section with an overview of the proposed traffic analysis attacks and details of each step in our algorithm are described after the overview.

3.3.1 Overview

The proposed traffic analysis attacks are based on packet timing information. As described in Section 3.1.2, silence suppression enables adversaries to recover talk patterns in terms of talk spurts and silence gaps from packet timing. Adversaries can create a Hidden Markov Model (HMM) to model Alice's talk pattern recovered from SIP-Based encrypted VoIP calls made by her. When adversaries want to determine which SIP-Based encrypted VoIP call is made by Alice, adversaries can check talk patterns recovered from the call of interest against Alice's model.

The proposed attacks can be divided into two phases: the training phase and the detection phase as shown in Figure 3. The two steps in the training phase are feature extraction and HMMs training. The detection phase consists of three steps: feature extraction, speech detection or speaker detection, and intersection attack. The last step, intersection attack, is optional. We describe the details of each step bel

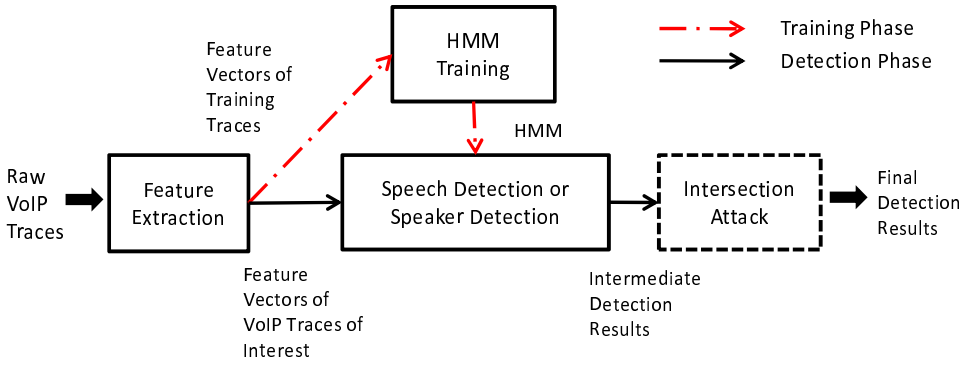


Figure 3: The Proposed Attack

3.3.2 Feature Extraction

The input and output of the feature extraction step are raw traces of VoIP calls and feature vectors respectively. The feature vector used in the proposed attacks is

shown below:

$$\begin{bmatrix} ts_1 & ts_2 & \cdots & ts_n \\ sg_1 & sg_2 & \cdots & sg_n \end{bmatrix}$$

where n is the length of a feature vector, ts_i and sg_j denote the length of the ith talk spurt and the jth silence gap respectively.

Talk spurts and silence gaps are differentiated by a silence threshold $T_{silence}$: If an inter-packet time is larger than the threshold, the inter-packet time is declared as a silence gap. Otherwise the inter-packet time is declared as a part of one talk spurt.

Obviously the threshold $T_{silence}$ is critical to overall detection performance. We did initial experiments to investigate the suitable range of the threshold for detection: We feed voice signals to VoIP clients and collect VoIP packets generated by VoIP clients. Different values of the threshold $T_{silence}$ are used to determine silence gaps. Actual silence gaps can be found by checking marker bits in RTP packets which indicate the start of talk spurts[2]. We evaluate a value of the threshold by two metrics: false positive rate and false negative rate. False positive rate is the fraction of talk spurts that were erroneously declared as silence gaps. False negative rate is the fraction of silences gaps that are erroneously declared as talk spurts. The experiment results with different codecs[3] are shown in Figure 4.

We can observe that for a wide range of the threshold $T_{silence}$, both the false positive rate and the false negative rate are low: When $T_{silence}$ is larger than $70ms$, the false positive rate is below 10% for all the codecs. The false negative rate is below 20% when $T_{silence}$ is less than $100ms$. The wide range is because of the big difference between inter-packet time of silence gaps and inter-packet time of talk spurts: Silence gaps are in order of seconds. Inter-packet time during talk spurts is usually around

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

[3]Details of these codecs can be found in Table 3.1.

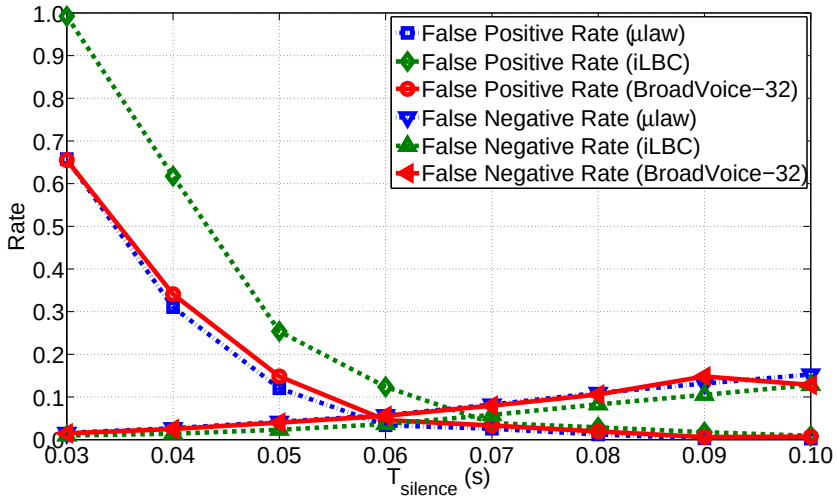


Figure 4: Experiment Results on the Threshold $T_{silence}$

packetization delay of $20ms$ or $30ms$ for most codecs.

We can also observe that increasing the threshold $T_{silence}$ decreases the false positive rate and increases the false negative rate. The changes in these two rates are again because the inter-packet time during silence gaps is larger than inter-packet time during talk spurts.

A big challenge in feature extraction is to filter out noise caused by random network delays in silence tests. Random network delays can cause errors in silence tests especially at receiving side: Because of random packet delays, inter-packet time during talk spurts can become larger than the threshold $T_{silence}$. The main idea of filtering noise in silence tests is to determine a silence gap based on N *successive* inter-packet intervals instead of *one* inter-packet interval. The silence test with filtering techniques works as follows: If one inter-packet interval is larger than the threshold $T_{silence}$, we declare a new silence gap only when none of the following $\lfloor \frac{T_{silence}}{packetization\ delay} \rfloor - 1$[4] inter-packet intervals are shorter than a threshold T_{spurt}, used to filter out long inter-packet intervals caused by network delays. The rationale behind the new silence tests method is that: If an inter-packet interval is erroneously declared as a silence gap

[4]We use $\lfloor \rfloor$ to denote floor operation.

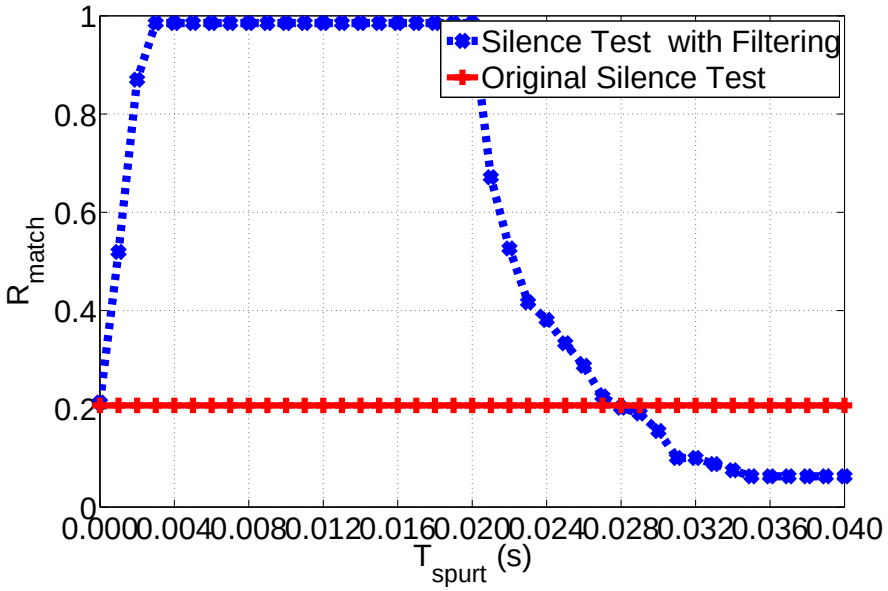


Figure 5: Match Rate R_{match} vs. Threshold T_{spurt} (μlaw Codec)

because network delays increase the length of the inter-packet interval, then following inter-packet intervals must likely be shorter than normal inter-packet intervals during talk spurts. The new silence tests can improve silence detection performance in terms of the false positive rate. The filtering does not focus on false negative errors because: (a) The false negative rate changes very little when $T_{silence}$ changes. (b) We take into account false negative errors in choices of HMM structures.

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

$$R_{match} = \frac{\text{The number of the gaps found in both sending side and receiving side}}{\text{The number of the gaps found in sending side}}$$

Ideally, the match rate R_{match} should be 1 meaning that silence gaps detected at sending side can match silence gaps detected at receiving side exactly. The experiment results are shown in Figure 5.

Figure 5 shows that the filtering technique can significantly increase the match

rate R_{match}: The original silence test can only achieve match rate of 0.21. The silence test with filtering techniques can achieve a match rate of 0.99 when T_{spurt} is between $3ms$ and $20ms$. The match rate is low when T_{spurt} is less than $0.3ms$ because the threshold T_{spurt} is too low to filter out large inter-packet intervals caused by network delays. The match rate is low when T_{spurt} is larger than $20ms$. It is because a tight threshold T_{spurt} filters out most of silence gaps since normal inter-packet intervals during talk spurts are of $20ms$ for μlaw codec. In following experiments, we set the threshold T_{spurt} to be $10ms$.

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

3.3.3 HMMs Training

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

A Hidden Markov Model (HMM) based classifier is used to detect speeches or speakers of VoIP calls. 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 [31], handwriting recognition [32], and gesture recognition [33]. In the proposed attacks, HMMs are trained to model talk patterns.

In this chapter, we consider each talk period including one talk spurt and one silence gap as a hidden (invisible) state. The output observation from one state is the length of a talk spurt and the following silence gap. Since each state corresponds to a talk period, a VoIP speech 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 speeches and language modeling. Even when the assumption does not hold strictly, the extended

HMM can still work well [34].

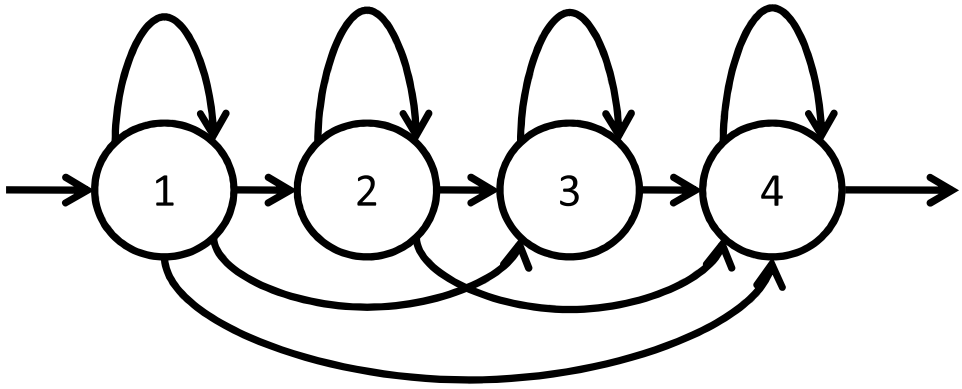


Figure 6: HMM

The HMM used in traffic analysis attacks is the modified left-right HMM [34] as shown in Figure 6. It is based on left-right models because of the nonergodic nature of speech signals [34], 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 is less than i. Additional constraints are placed on the state transitions in the left-right model to make sure that large changes in state diversion do not occur, i.e., $a_{ij} = 0$, when $j > i + \Delta$. For the well-known banded left-right model [34] and Bakis model [35], Δ is 1 and 2 respectively.

We extended classical left-right models to allow transition from the ith state to the $(i+3)$th state, i.e., $\Delta = 3$, as hown in Figure 6. Our modification on the left-right model is because of possible false negative errors made in the feature extraction step and adaptive silence thresholds used in silence detectors as described in Section 3.1: **False negative errors** made when some silence gaps are not detected in feature extraction. The false negative errors can be caused by a large threshold $T_{silence}$ or hangover time as described in Section 3.1. Hangover time reduces length of silence gaps recovered from the packet timing since VoIP packets still being sent during the

beginning and the end of 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 VoIP calls. In modern codecs, the threshold used in a silence detector dynamically changes to adapt to changes in background noise. Because of the dynamically changing threshold, silence duration in the same speech can be detected as silence in one call or as a part of a talk spurt in another call. Although the inconsistent detection because of adaptive silence threshold does not happen very often, it affects our overall speech detection and speaker detection performance.

To take into account the possible false negative errors made in the feature extraction and inconsistency of VoIP silence detectors, we allow state transition 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 VoIP call traces. Our experiments with different left-right models also show that the modified left-right model can achieve better detection performance than other left-right models.

In the modified HMM, the number of states are heuristically set to be 10 according to length of feature vectors. Following the principle of Occam's razor, the number of states should be small enough to avoid over-fitting and large enough to model the ergodic nature of VoIP calls. We get similar detection performance for different number of states when the number of states is larger than five, as shown in Figure 7. When the number of state is too large, the training of HMMs fails to converge to an optimal solution.

In this step, two kinds of HMMs can be trained: (a) A speaker-specific model can be obtained by training the HMM with traces of VoIP calls made by Alice. (b) For speeches detection, we focus on detecting speeches made by one specific speaker, say

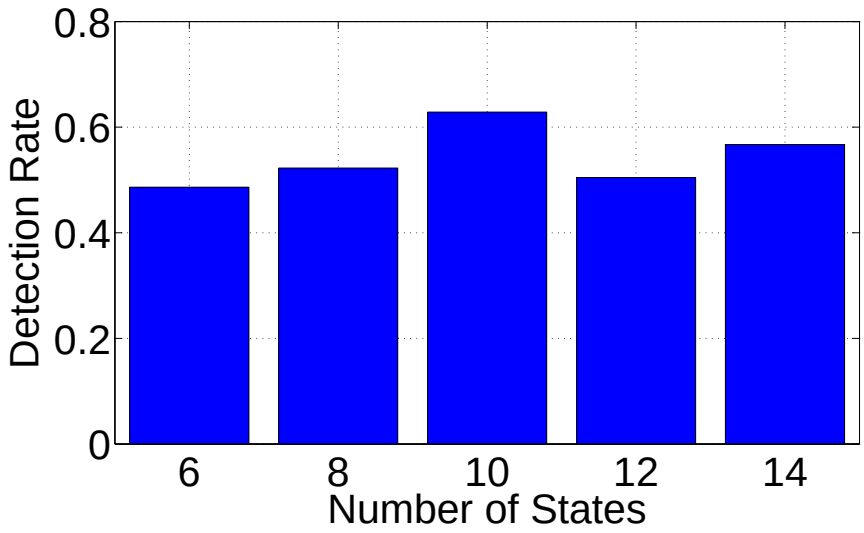


Figure 7: Detection Rate with Different Number of States

Alice. So a speech-specific model can be obtained by training the model with traces of the same speech made by Alice. The trained HMMs are used in the following speech detection or speaker detection step.

3.3.4 Speech Detection and Speaker Detection

The inputs to this step are the Alice's speech-specific or the Alice's speaker-specific HMM trained in the previous step and feature vectors generated from a pool of raw VoIP traces of interest. The output of this step is the intermediate detection result. For the speaker detection, the intermediate detection result is K_{top} speakers from the candidate pool with talk patterns closest to Alice's talk pattern. For the speech detection, the intermediate detection result is K_{top} speeches from the candidate pool with speech patterns closest to speech patterns in training traces.

The detection step can be divided into two phases: (a) First, the likelihood of each feature vector is calculated with the trained HMM. (b) The trace with the highest likelihood is declared as the trace generated from one specific speech or by Alice if the intersection step is not used. To improve detection accuracy, the intermediate

detection results can be fed into the optional intersection attack step.

3.3.5 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 [36, 37, 38]: Instead of making a detection decision result based on one trial, we can improve detection accuracy by 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 VoIP call traces made by the same speaker, the adversary can have m trails as described in Section 3.3.4. From each detection, the adversary can obtain K_{top} traces with K_{top} highest likelihoods. The overall rank for each speaker is calculated by adding ranks in m trails. The speaker with highest rank is determined to be Alice. Tie can be broken by comparing the sum of likelihood in m trails.

In summary, the proposed traffic analysis attacks can be divided into two phases: the training phase and the detection phase. Since application-level features are extracted from VoIP calls, the traffic analysis attacks are independent from codecs used for VoIP communication. In other words, it is possible to train the HMM with traces of VoIP calls made with one codec and detect speeches or speakers of VoIP calls made with another codec. We evaluate the proposed traffic analysis attacks with empirical experiments as described below.

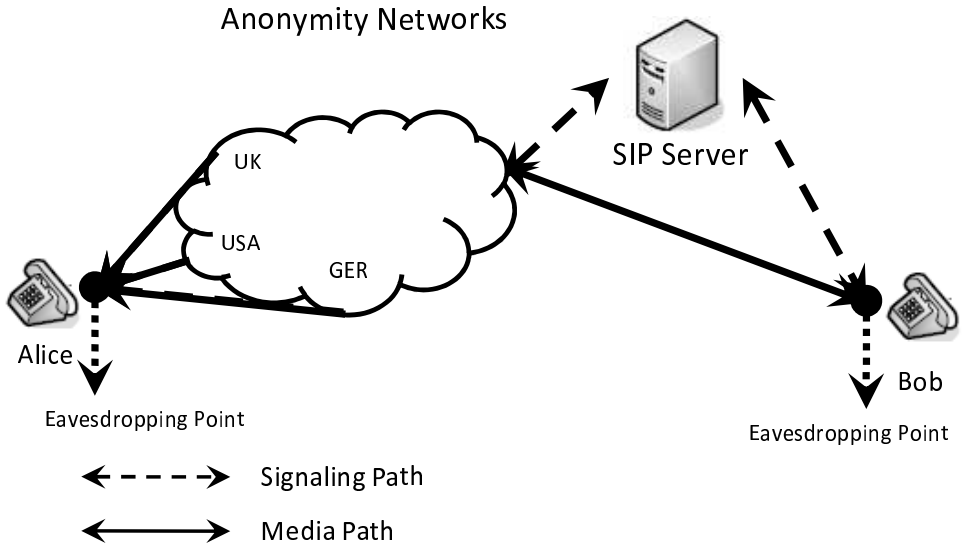


Figure 8: Experiment Setup

Table II: 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

3.4 Empirical Evaluation

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

3.4.1 Experiment Setup

The experiment setup is as shown in Figure 8. VoIP packets are first directed to the anonymity network managed by findnot.com before arriving at the other side of the call. We use the commercial anonymous communication services provided by findnot.com[5] mainly because it is possible to select entry points into the anonymity network [39]. In our experiments, VoIP packets were directed through entry points

[5]We did not use Tor [3] to anonymize VoIP calls because Tor has no direct support of anonymizing UDP packets and VoIP packets generally are UDP packets.

in England, Germany, and United States as shown in Figure 8. For these VoIP calls made through anonymity networks, the end-to-end delay is at least $80ms$ and the two communication parties are at least 20 hops away from each other. About a quarter of calls are made through the campus network so that traces of VoIP calls over a wide range of networks are available for our experiments.

The audio signals are extracted from videos hosted on Research Channels [40]. Traces used in both training and detection are 14.7 minutes long on average if not specified[6]. For the purpose of intersection attacks, at least three different speeches are available for most speakers and each speech was sent through at least four different network entry points[7]. Totally 360 VoIP calls were made through different entry points of the anonymity network managed by findnot.com and through the campus network.

In our experiments, VoIP calls were made with X-Lite [28], one of the most popular SIP-based VoIP clients. X-Lite supports a wide range of codecs for different voice quality. 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 II.

3.4.2 Metrics

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

For both speech detection and speaker detection with traces generated by the same codec, the detection rate for random guess is about $\frac{1}{109}$, because in each trial,

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

[7]The campus network entry point is one of the choices.

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., the trace generated by a specific speech. In each trial of speech detection, three traces of the same speech are used for training and one trace of the same speech is one of the candidate traces. In each trial of speaker detection, one trace of Alice's speeches is used as one of the candidate traces and Alice's other traces are used for training.

For *cross-codec detection*, i.e., detection with traces generated from all codecs used in experiments, the detection rate for random guess is about $\frac{1}{325}$, because in each trial, there are around 325 candidate traces in the pool including the "right" trace. In each trial of speech detection, eleven traces of the same speech from three different codecs are used for training and one trace of the same speech is one of the candidate traces. In each trial of speaker detection, one trace of Alice's speeches is used as one of the candidate traces and Alice's other traces are used for training.

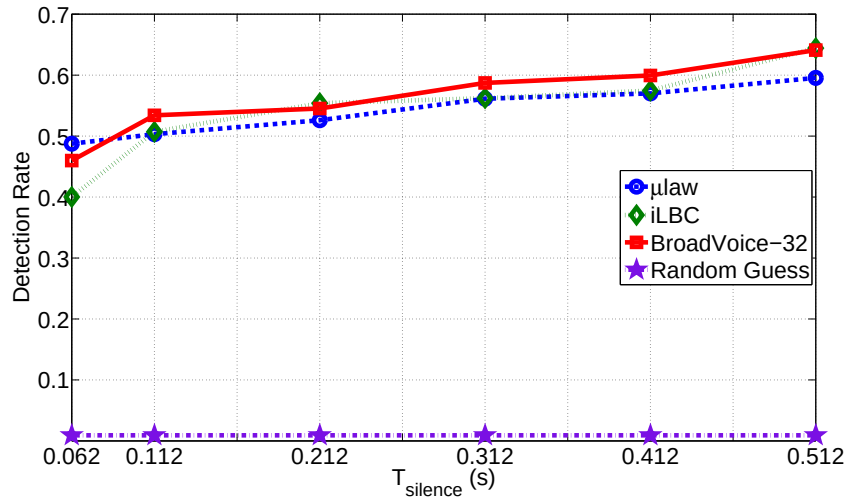
In all the experiments below, the training traces and candidate traces are all collected from *different* VoIP calls. For better training, all the traces used in training are collected from sending side, i.e., from the link connected to Alice's computer.

3.4.3 Threshold $T_{silence}$

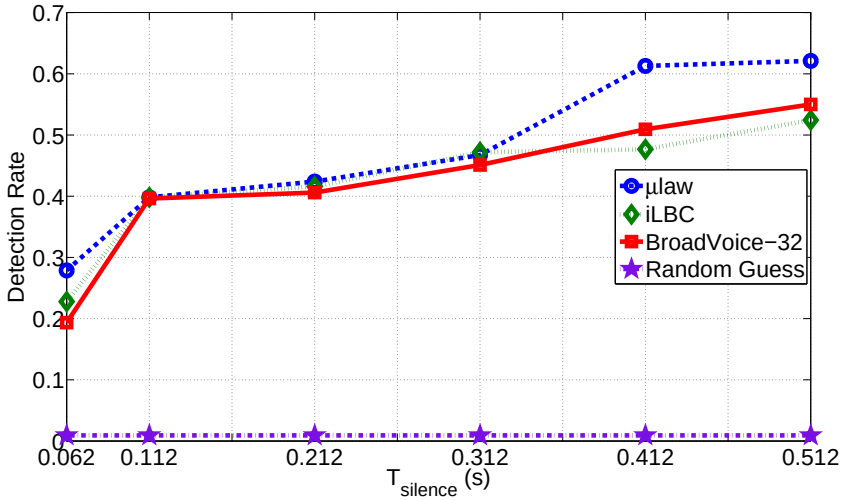
This series of experiments are designed to test the effect of the parameter $T_{silence}$, the threshold used in silence tests.

Figure 9 shows speech detection performance with different threshold $T_{silence}$. In both Figure 9 and Figure 10, each detection rate is obtained based on 120 trials.

From Figure 9, we can observe: (a) For a wide range of $T_{silence}$, the detection rate is larger than 0.5, about 55-fold over random guess. The detection rate can be higher than 0.55 for all the three codecs, so more than 61-fold improvement over random guess. For BoradVoice-32 codec, the detection rate can reach 0.66. (b) In

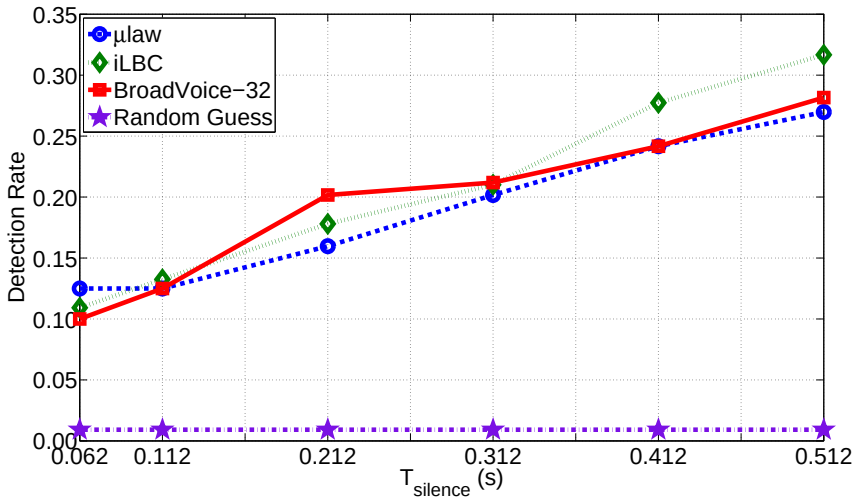


(a) Candidate Traces Collected from Sending Side

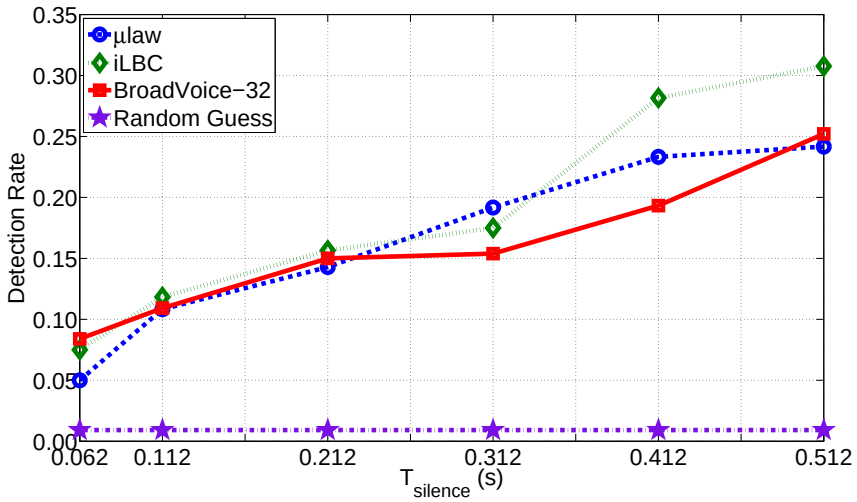


(b) Candidate Traces Collected from Receiving Side

Figure 9: Speech Detection Performance with Different Threshold $T_{silence}$



(a) Candidate Traces Collected from Sending Side



(b) Candidate Traces Collected from Receiving Side

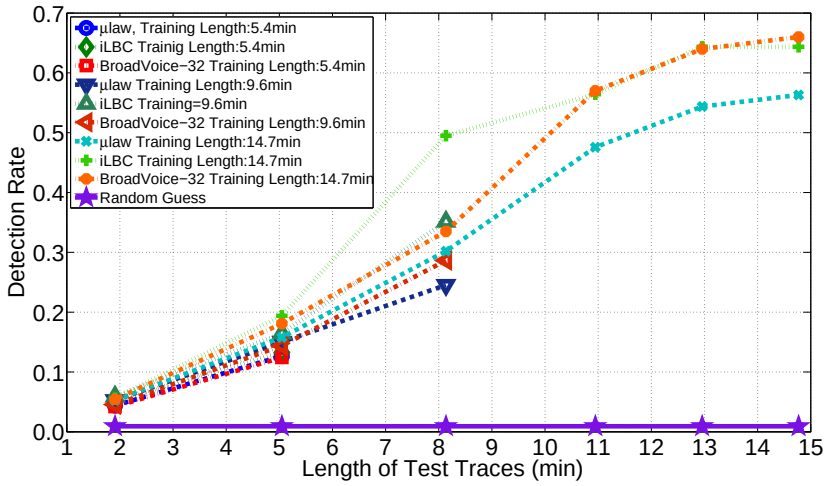
Figure 10: Speaker Detection Performance with Different Threshold $T_{silence}$

general, the detection rate increases when the threshold $T_{silence}$ increases. When $T_{silence}$ becomes large, the detection rate may drop simply because shorter feature vectors are used for training and detection. When $T_{silence}$ is larger than $0.512s$, feature vectors are too short for detection so that the HMM training cannot converge for the certain traces. (c) The detection rate for candidate traces collected from sending side is comparable with the detection rate for candidate traces collected from receiving side. It is because filtering techniques used in silence tests can largely filter out noise caused by random network delays at receiving side which can vary from call to call. Similar observations can be made from Figure 10. The detection rate for speaker detection can reach 0.32, about 35-fold improvement over random guess. In the following experiments, we set $T_{silence}$ to be 0.412 second.

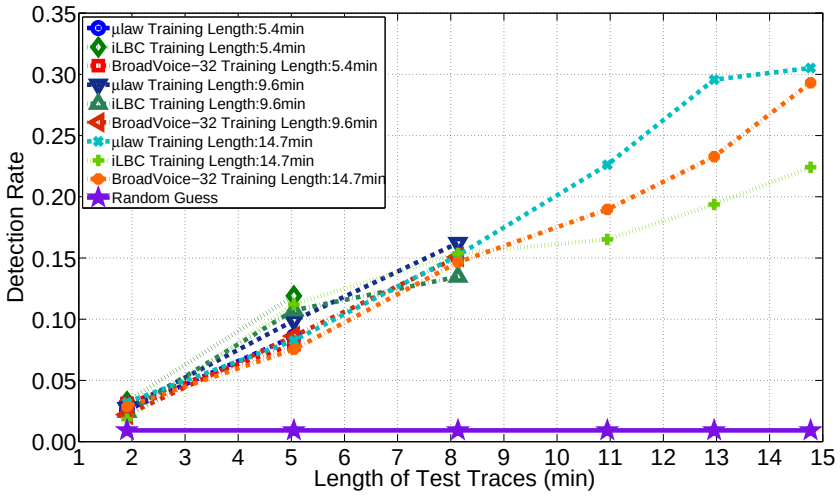
3.4.4 Length of Training and Test Traces

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

From experiment results shown in Figure 11, we can observe that even for five-minute-long training and test traces, the detection rate for speech detection and speaker detection can achieve 0.20 and 0.12, about 22-fold and 13-fold improvement over random guess respectively. Figure 11 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 training traces and test traces to be 14.7 minutes.



(a) Speech Test



(b) Speaker Test

Figure 11: Detection Performance with Different Length of Training Traces and Test Traces

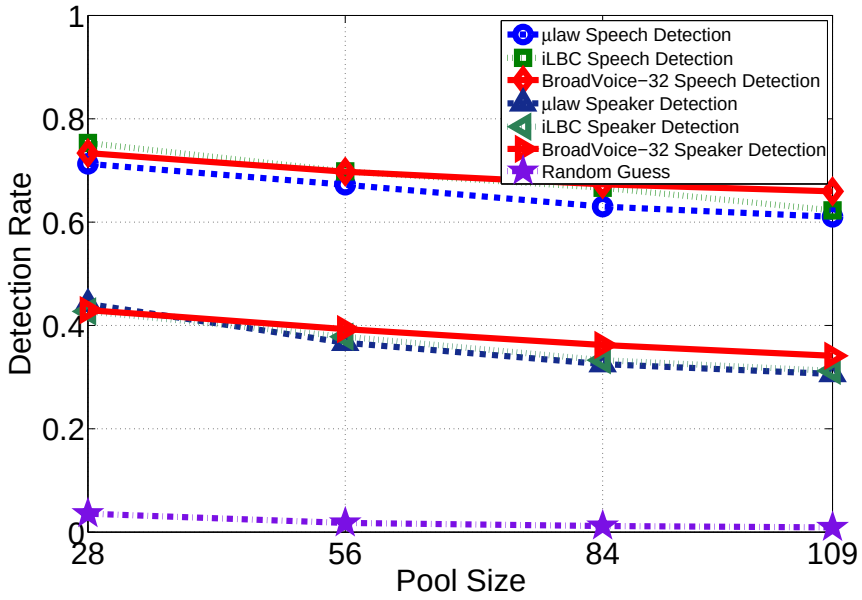


Figure 12: Detection Performance with Different Pool Size

3.4.5 Pool Size

In this set of experiments, we investigate detection performance with different sizes of candidate pools. From the experiment results shown in Figure 12, we can observe that when the pool size increases, the detection rate slightly decreases for all the codecs, since it is harder to find the right one from a larger pool. But the ratio between the speech detection rate and random guess rate changes from 20 to 70, when the pool size changes from 28 to 109, meaning the traffic analysis attacks are more effective when the pool size is large. We can also observe that for the μlaw codec, one of the most frequently used codec in VoIP telephony, the speech detection and the speaker detection can achieve detection rate of 0.72 and 0.65 when the pool size is 28, approximately 20-fold and 70-fold improvement over random guess respectively.

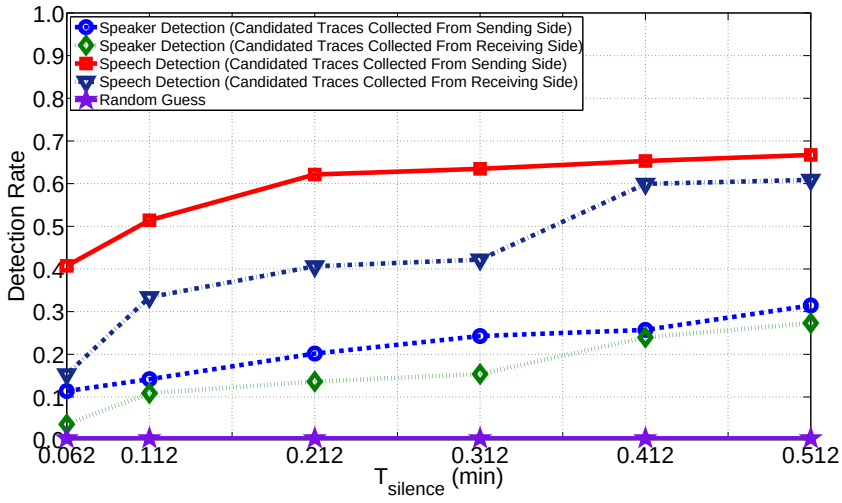


Figure 13: Cross-Codec Detection Performance with Different Threshold $T_{silence}$

3.4.6 Cross-Codec Detection

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

Figure 13 shows performance of cross-codec detection with different threshold $T_{silence}$. Each detection rate in Figure 13 is obtained based on about 360 trials. We can observe that the detection rates for speech detection and speaker detection can reach 0.61 and 0.31 respectively, about 203-fold and 103-fold improvement over random guess.

Figure 14 shows the detection performance with different length of training traces and test traces. We can again observe the detection rate increases with the length of training traces and test traces. The speech detection and speaker detection with only five minutes of training traces and test traces can achieve 0.12 and 0.21,

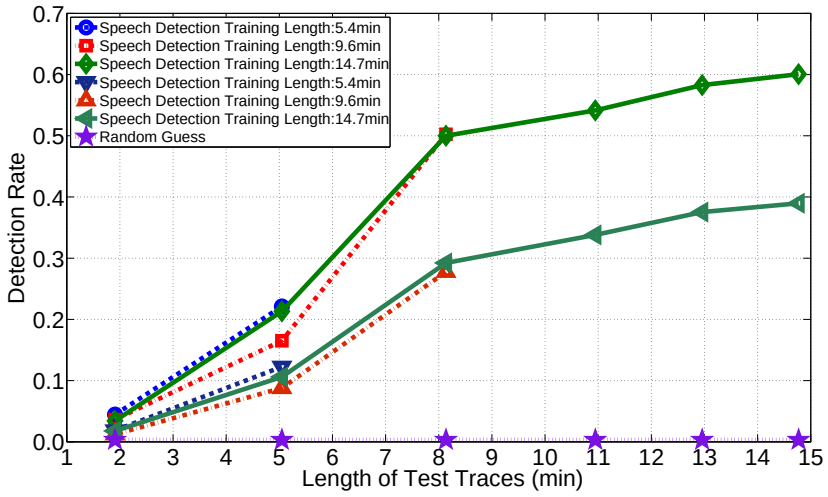


Figure 14: Cross-Codec Detection Performance with Different Length of Training Trace and Test Traces

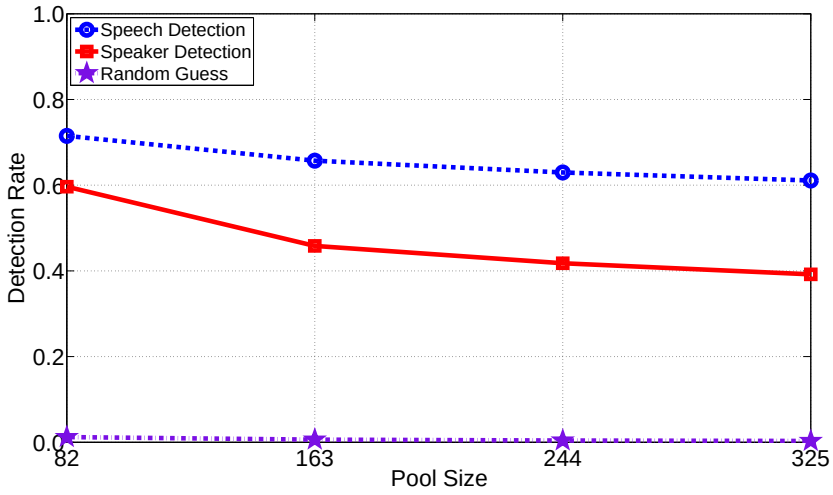


Figure 15: Cross-Codec Detection Performance with Different Pool Size

Table III: 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	0.3500	0.3625	0.3750	0.4000
412	0.4325	0.4500	0.6000	0.6250

40-fold and 70-fold improvement over random guess respectively.

Figure 15 shows the detection performance with different sizes of candidate pools. We can again observe that the detection rate decreases slightly with the increase of pool size. When the pool size is 82, the speech detection and speaker detection can achieve 0.72 and 0.60 respectively.

3.4.7 Intersection Attack

In this set of experiments, we evaluate the effectiveness of intersection attacks on cross-codec speaker detection. On average, there are 37 candidate speakers in each trial. So the detection rate for random guess is about $\frac{1}{37}$. Each candidate speaker has 9 VoIP traces available for detections. So the final detection result is obtained by combining the intermediate detection results of 9 trials.

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

In summary, the proposed traffic analysis attacks can significantly improve the detection rate over random guess. We believe that given more training traces, higher detection rate can be achieved.

3.5 Detecting Speaker without Candidate Pools

The initial threat model assumes that the "right" speaker is in a 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 candidate pools.

Instead we assume that the adversary possesses traces of VoIP calls made by Alice and other persons. We call these traces as labeled traces since these traces are collected in advance and the adversary knows the identities of callers. The goal of the adversary is to detect, whether Alice is the speaker of a VoIP call of interest.

3.5.1 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 calls 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} is determined based on remaining labeled traces including the second half of traces of Alice's calls. The adversary evaluates each of these traces against Alice's model and calculates its likelihood. Given a threshold T_{lik}, the false positive rate and the false negative rate on the remaining labeled traces can be calculated as follows: (a) False negative rate is defined as the proportion of Alice's calls detected as calls made by other speakers, i.e., the proportion of Alice's calls with likelihood values less than T_{lik}. (b) False positive rate is defined as the proportion of calls made by other speakers detected as Alice's calls, i.e., the proportion of other speakers' calls with likelihood values larger than T_{lik}. The threshold T_{lik} is selected so that the detection rates on the remaining 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 a trace of Alice's call. Otherwise, the trace is declared as a trace made by other speakers.

3.5.2 Performance Evaluation

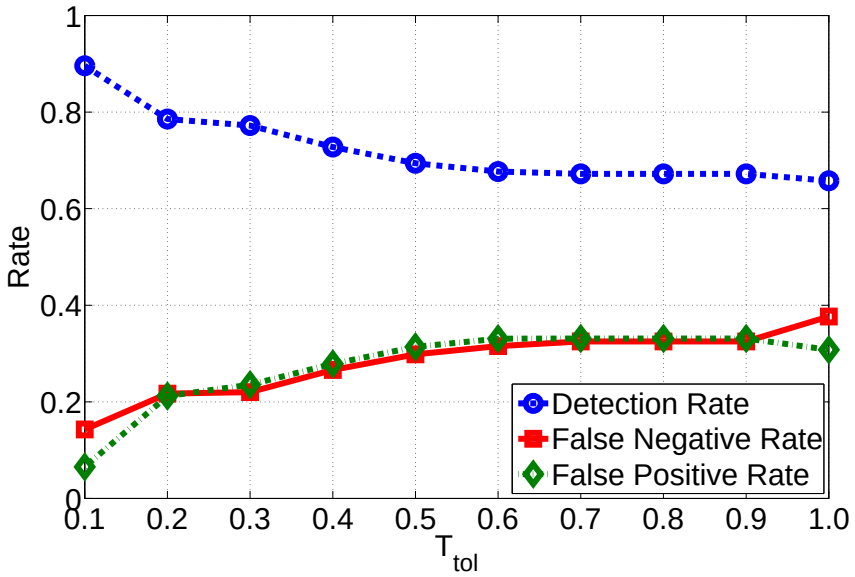
We evaluate the detection performance with four metrics: detection rate, false negative rate, false positive rate, and percentage of traces which can be tested. The two metrics, the false negative rate and the false positive rate used in performance evaluation, are calculated on the test traces. The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

In this set of experiments, the average length of labeled traces and test traces are 14.7 minutes. In each detection, there are 54 labeled traces and 6 traces of Alice's calls. The experiment results are averaged over 120 tests.

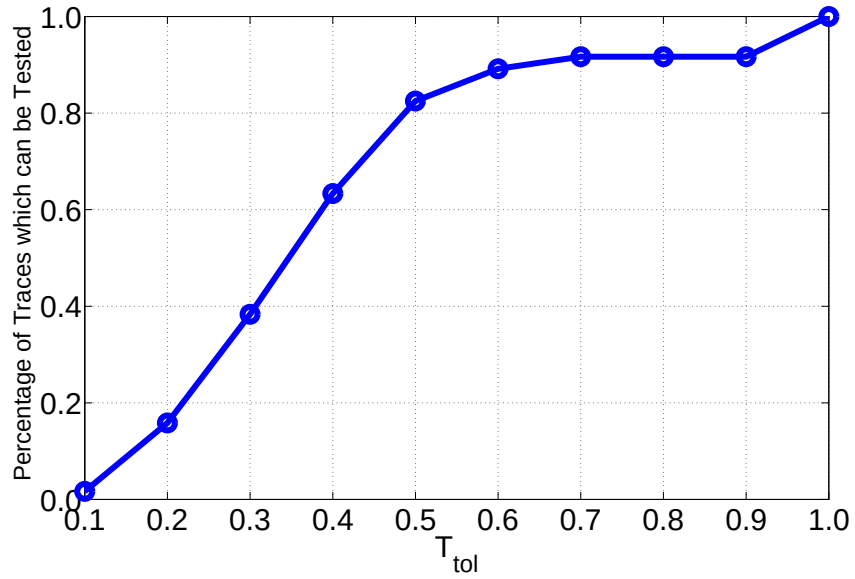
Experiment results shown in Figure 16 indicate that detection rate decreases when the tolerance T_{tol} increases and in the mean time, the percentage of trace which can be tested increases. A smaller tolerance T_{tol} means better training, and in turn, better detection performance. A smaller tolerance T_{tol} also means stricter requirements so fewer traces can be tested. We can also observe that the detection rate can reach 0.89 when $T_{tol} = 0.1$ and only 1% traces can be tested. When $T_{tol} = 1$, i.e., all the traces can be tested, the detection rate is 0.63.

3.6 A Countermeasure and Its Performance

From the discussion above, it is apparent that the proposed traffic analysis attacks can greatly compromise the privacy of encrypted VoIP calls. Countermeasures



(a) Detection Rate



(b) Percentage of Test Traces that can be Tested

Figure 16: Detection Performance

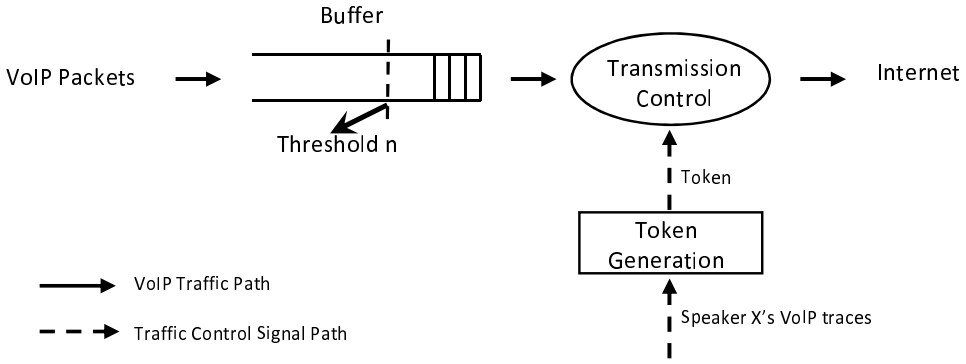


Figure 17: Countermeasure: Camouflaging Alice's VoIP Packets

are needed for privacy protection. In this section, we introduce a countermeasure which can both protect privacy with only marginally effect on quality of service (QoS) of VoIP calls.

3.6.1 Overview

Simple countermeasures to the proposed traffic analysis attacks include padding VoIP traffic to constant rate traffic or randomly delaying VoIP packets to hide talk patterns. These simple approaches may render the proposed traffic analysis attacks ineffective. But these approaches can cause significant waste of bandwidth or degrade the QoS of VoIP calls significantly.

The main idea of our countermeasure is to camouflage the timing of Alice's VoIP packets according to another speaker's traces. As shown in Figure 17, Alice's VoIP 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 VoIP traces. The transmissions of Alice's VoIP packets are controlled by these tokens. The transmission control in Figure 32 functions as follows: (a) Each packet transmission consumes a token. (b) When a token is generated and the buffer is not empty, transmission control will transmit the first packet in the buffer. (c) When a token is generated and the buffer is empty, a dummy packet is transmitted by the transmission control. (d) When N_{buff} packets

are held in the buffer and no token is available, the first packet in the buffer will be transmitted.

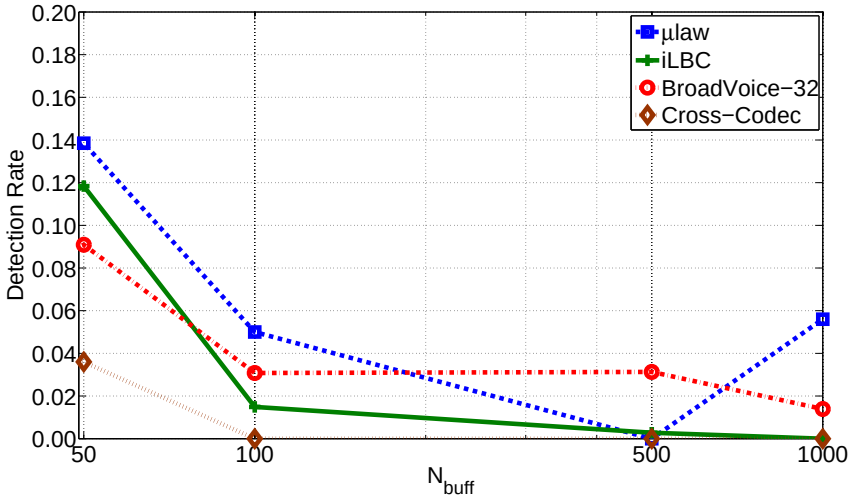
For the proposed countermeasure, dummy packets are sent only when necessary for camouflaging. The parameter N_{buff} is used to control queuing delays. This parameter should be carefully chosen to balance the QoS of VoIP calls and privacy protection to defeat traffic analysis attacks.

3.6.2 Performance Evaluation of the Countermeasure

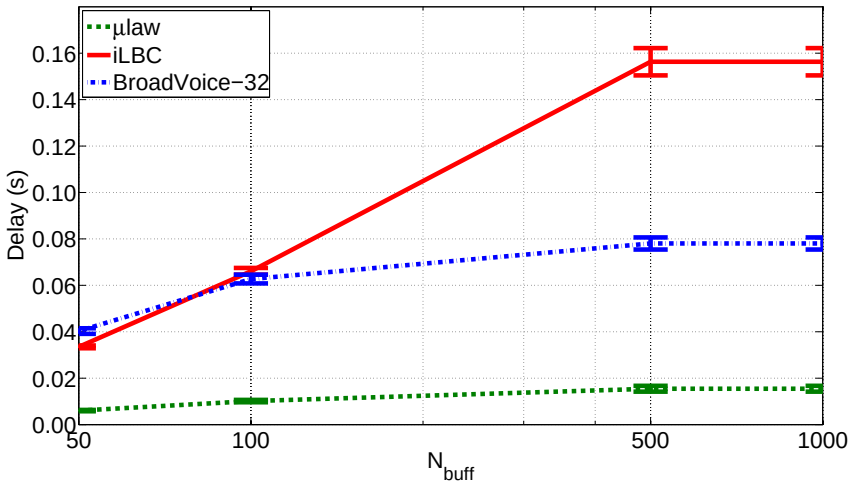
We evaluate the countermeasure with two metrics: (a) The detection rate defined in Section 3.4.2: It is used to measure the performance of privacy protection of VoIP calls. (b) Additional packet delays caused by the countermeasure: It measures the degradation of the QoS of VoIP calls.

In this set of experiments, we use real traces collected from the experiment environment described in Section 3.4.1.

Figure 18 shows the performance of the countermeasure. The threshold $T_{silence}$ is set to $0.412s$ and the corresponding detection rate is 0.6 a threshold with as shown in Figure 9. Figure 18(a) shows that the countermeasure can protect the privacy of VoIP calls since most detection rates are around the probability of random guess. Figure 18(b) shows additional packet delays caused by the countermeasure. When N_{buff} is 50 and 100, the additional delays caused by the countermeasure is less than $36ms$ and $68ms$ with a probability larger than 0.95 respectively. So the countermeasure will not cause any significant change in the quality of VoIP calls since the additional delays for $N_{buff} = 50$ and $N_{buff} = 100$ are still less than one third of and half of the delay budget for VoIP calls [41] respectively. The detection rates for small N_{buff}, such as $N_{buff} = 1$ and $N_{buff} = 10$, cannot be obtained from experiments, because N_{buff} is too small and no silence gaps can be found in privacy VoIP traces.



(a) Detection Rate



(b) Additional Packet Delay with 95% Confidence Interval

Figure 18: Performance of the Proposed Countermeasure

3.7 Discussion and Future Work

Our experiments clearly show that the proposed traffic analysis attacks can greatly compromise privacy of VoIP calls. The detection rates for speech detection and speaker detection are 70-fold and 35-fold improvement over random guess. Higher detection rate can be achieved with more training traces.

Comparable detection performances are achieved for both traces collected by sending side and receiving side. It is an indication that when the threshold is large enough, feature extracted in the proposed attacks are largely independent of network dynamics.

The framework proposed in this chapter, including extracting application-level features from network traffic traces and statistical analysis of extracted application-level feature by HMMs, can be potentially used to infer other sensitive information at application level. For example, the framework can be potentially used to detect speaker's emotion during a call suppose the speaker's talk behavior can change significantly when the speaker's mood changes. The framework may also be used to detect different types of speeches such as seminar talk, conversation between two parties, and classroom discussion. One of our future works is to explore the potential of the framework experimentally and theoretically.

CHAPTER IV

TRAFFIC ANALYSIS ATTACKS ON SKYPE VOIP CALLS

In this chapter, we address on privacy issues of Skype calls. Skype is one of the most popular VoIP service providers. Skype VoIP services are provided on a peer-to-peer structure. Skype peers form an overlay network. A Skype call may be dynamically routed through Skype peers during the call for better Quality of Service (QoS) [42]. One of the main reasons for the popularity of Skype VoIP services is its unique set of features to protect privacy of VoIP calls such as strong encryption [43], proprietary protocols [43], unknown codecs [44], dynamic path selection [42], and the constant packet rate [45].

In this chapter, we propose a class of passive traffic analysis attacks to compromise privacy of Skype calls. The procedure of proposed attacks is as follows: First the adversary collected Skype call traces made by a victim, say Alice. The adversary then extracts application-level features of Alice's VoIP calls and trains a Hidden Markov Model (HMM) with the extracted features. To test whether a call of interest is made

by Alice, the adversary can extract features from the trace of the call and calculate likelihood of the call being made by Alice. The proposed attacks can identify speeches or speakers of Skype calls with high probabilities.

4.1 Problem Definition

In this chapter, we focus on traffic analysis on Skype VoIP calls through anonymity networks to disclose sensitive information at application-level. More specifically, we are interested in detecting speeches and speakers of Skype VoIP calls by analyzing traffic patterns at the application-level.

A typical attack scenario focused in this chapter is as follows: An adversary who has possession of traces of *previous* Skype VoIP calls made by a victim, say Alice, may want to detect whether Alice is talking to Bob *now* by collecting Skype packets on the link to Bob. The adversary may also want to detect the speech content, such as the repetition of a partial speech in previous Skype calls.

In this chapter, we assume that traffic traces used in analysis can be collected at different time. This is the major difference between our research and the previous researches. Most of the previous researches assume that the adversary has *simultaneous* access to *both* links connected to Alice and Bob *during the Skype call* between Alice and Bob. By passively correlating VoIP flows at both ends or actively watermarking VoIP flows, the adversary can detect whether Alice is communicating with Bob. But for the typical attack scenario described above, both flow correlation and watermarking techniques do not work because traces to be compared are collected from different VoIP calls: (a) Correlation between different calls is low. (b) Watermarks used to mark traffic flows of Alice's VoIP calls can be different for different calls because of recycling watermarks or simply because Alice is making a call from a different computer.

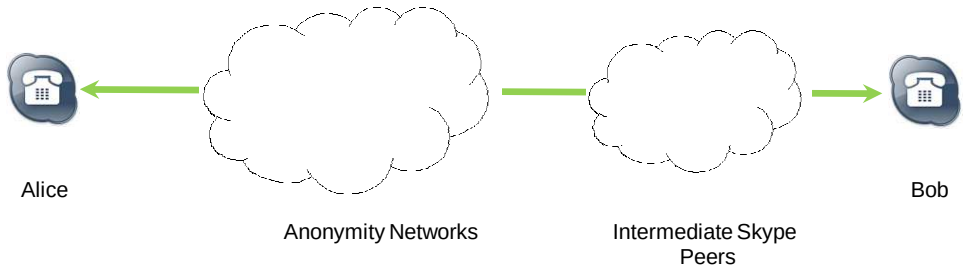


Figure 19: Network Model

4.1.1 Network Model

In the chapter, we assume Alice makes VoIP calls by Skype. We are particularly interested in Skype VoIP calls because: (a) Skype is based on peer-to-peer structure. During a Skype call, VoIP packets may follow more than one path through different Skype peers or Skype supernodes [42]. The peer-to-peer structure and dynamic path selection make security attacks or eavesdropping on Skype calls more difficult. (b) Skype uses proprietary protocols so that attackers cannot differentiate media packets from signaling packets. (c) Skype uses unknown codecs that renders traffic analysis exploiting characteristics of voice codecs nearly impossible [44]. (d) Skype calls are encrypted and hard to decipher [43]. (e) Skype sends packets at the constant rate of 33 packet/second [45]. Due to the unique set of features listed above, Skype is known as secure voice communication [43] which can protect privacy of communication parties.

As shown in Figure 19, we assume Alice routes Skype calls through anonymity networks to further protect privacy of her Skype calls. For better voice quality, Alice can use low-latency anonymity networks such as Tor and JAP.

4.1.2 Threat Model

We focus on passive attacks in this thesis. In other words, the attacks launched by the adversary do not disturb the existing network traffic. In comparison with active traffic analysis attacks [8, 46], the proposed attacks are harder to detect.

We assume that the adversary only has access to the links directly connected to participants of VoIP calls. This assumption is widely used in traffic analysis attacks such as attacks on anonymity networks [8]. We do not assume the adversary as a global attacker because re-routing techniques used in anonymity networks and dynamic path selection employed by Skype make global attacks too costly to be practical.

Our threat model does not require *simultaneous* access to the links connected to participants of a VoIP call since it may not be feasible for long-distance calls, such as international calls. Instead we assume the adversary can collect traces of VoIP calls made by Alice in advance and use these collected traces to detect whether Alice is a participant in the VoIP conversation of interest. Our model is similar as the model for identifying a human being by fingerprints: Fingerprints of human beings are collected in advance through 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.

The threat model assumes the detections are based on different Skype calls. So the speaker identification should also be independent of the voice content of Skype calls.

4.2 Detecting Speech and Speaker of Skype-Based VoIP Calls

In this section, we describe traffic analysis attacks to detect speeches or speakers of encrypted VoIP calls. We begin the section with an overview of the proposed traffic analysis attack and details of each step in our algorithm are described after the overview.

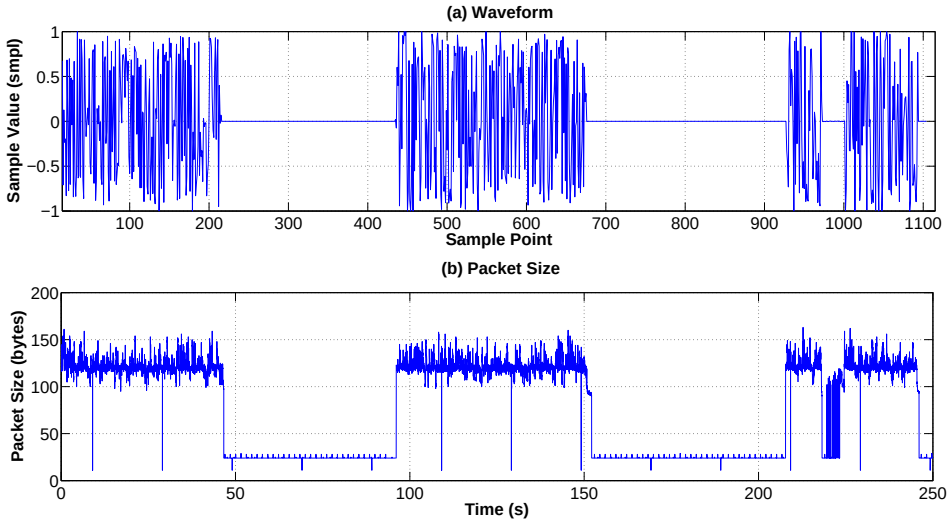


Figure 20: An Example

4.2.1 Overview

The proposed traffic analysis attacks are based on packet size information. A simple experiment shown in Figure 20 indicates that packet size information can disclose speech-level information. Figure 20(a) shows an audio signal with three silence periods. Figure 20(b) shows the packet sequence generated by feeding the audio signal into Skype clients. From the packet sequence plotted in Figure 20(b), we can observe: (a) Even during silent periods, Skype clients still generate packets at a constant rate. (b) During silent periods, VoIP packets generated by Skype are small in comparison with packets generated during talk periods. We do not focus on packet timing information in this chapter mainly because Skype clients send VoIP packets at a constant rate [45].

One of the challenges in this chapter is to extract application-level features from collected VoIP packet traces, i.e., features existing in different VoIP calls. Based on the features existing in different VoIP calls, traffic analysis attacks can possibly detect speeches or speakers of VoIP calls. The feature used in the proposed attacks is the throughput vector $[s_1, s_2, \cdots, s_n]$, where n is the length of the vector. The element

s_i in the throughput vector is calculated as follows:

$$s_i = \frac{\text{sum of bytes received or sent during the } i^{th} \text{ sample interval}}{T} \tag{4.1}$$

where T is the length of sample intervals.

The length of sample interval T should be selected in the order of seconds for the following two reasons: (a) Because of re-routing techniques used in anonymity networks and dynamic path selection employed in Skype, VoIP packets can arrive at destination in an order different from the order at sending end. A larger sample interval can largely absorb the difference. This is also the reason why we do not use per-packet size as the feature vector. (b) Talk patterns are of low frequency while network dynamics is of higher frequency. Network dynamics is usually in the order of millisecond while the patterns such as silent periods are in the order of seconds [47, 48]. The averaging effect of sample intervals is equivalent as low-pass filtering. A larger sample interval in the order of seconds can filter out network dynamics information which can vary from call to call and keep the low-frequency talk patterns.

The Hidden Markov Model (HMM) has been introduced in 3.3.3. In the proposed attacks, HMMs are trained to model talk patterns used for speech detection or speaker detection.

The proposed attacks can be divided into two phases: the training phase and the detection phase as shown in Figure 21. The two steps in the training phase are feature extraction and HMMs training. The detection phase consists of three steps: feature extraction, speech detection or speaker detection, and intersection attack. The last step, intersection attack, is optional. We describe the details of each step below.

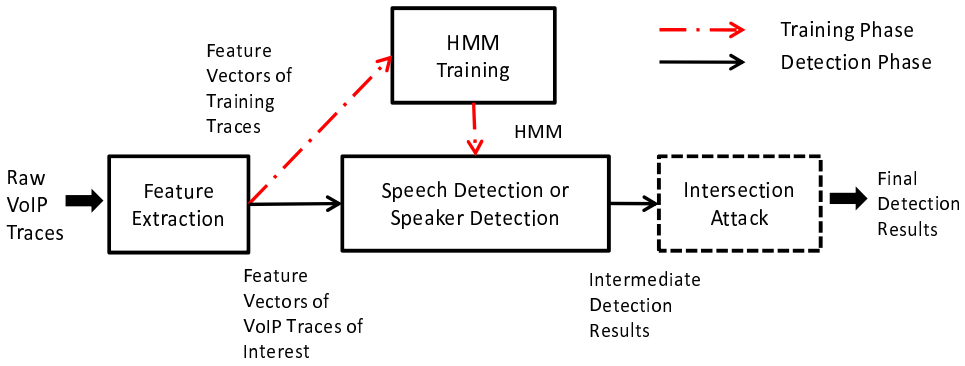


Figure 21: Steps of The Proposed Attacks

4.2.2 Feature Extraction

The input and output of the feature extraction step are raw traces of VoIP calls and throughput vectors respectively.

Two parameters are used in this step to control the generation of throughput vectors: (a) Length of sample interval T: As described in Section 4.2.1, the length of sample interval should be large enough to filter out network dynamics different from call to call and keep talk patterns. At the same time, it is desired to select a sample interval small enough so that throughput vectors are long enough for the training purpose. (b) Threshold on packet size H_{packet}: The threshold is used to filter out signaling packets and excluding signaling packets can lead to better trained HMMs of talk patterns. Since Skype uses proprietary protocols, unknown codecs, and encryption, it is impossible to separate signaling packets based on protocol headers. We heuristically differentiate signaling packets from media packets by the threshold H_{packet}: Signaling packets are usually smaller than media packets. In raw VoIP traces, we also find that packets of small and fixed sizes are sent or received periodically and independent of speech activities. The guidelines on the choice of these two parameters are given in Section 4.3.

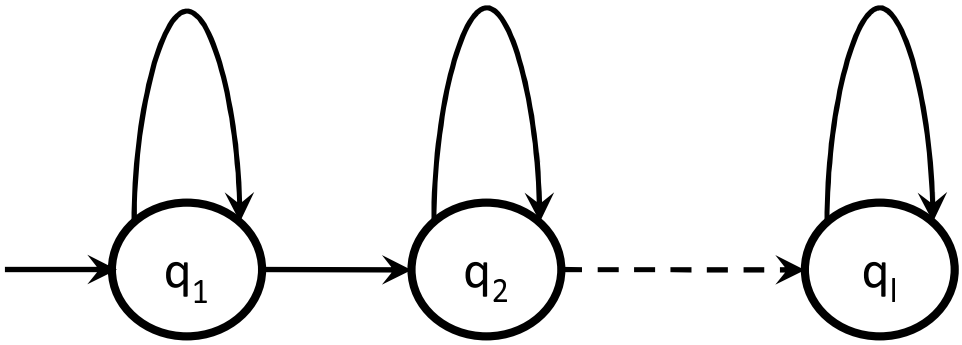


Figure 22: Left-right Hidden Markov Model

4.2.3 HMM Training

The input and output of this step are throughput 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, i.e., independent from 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 outputs. Therefore the sequence of outputs generated by an HMM gives some information about the sequence of states.

In the proposed attacks, HMMs are trained to model talk patterns used for speech detection or speaker detection. More specifically, the attacks are based on on-off patterns of silence in speeches which have been used as one feature for speaker detection [49]. As shown in Figure 20, the on-off patterns in speeches can be possibly recovered from packet size. But the pattern recovery is noisy because: (a) It is impossible to differentiate voice packets from signaling packets. (b) A sample interval may contain several on-off periods or may be a part of a long silent gap or talk spurt. Ideally only two states, talk and silence, are enough to model talk patterns with a voice silence detector as used in [49]. Because of the noise in pattern recovery, more states of different combinations of on-off periods are used in the HMM. The number

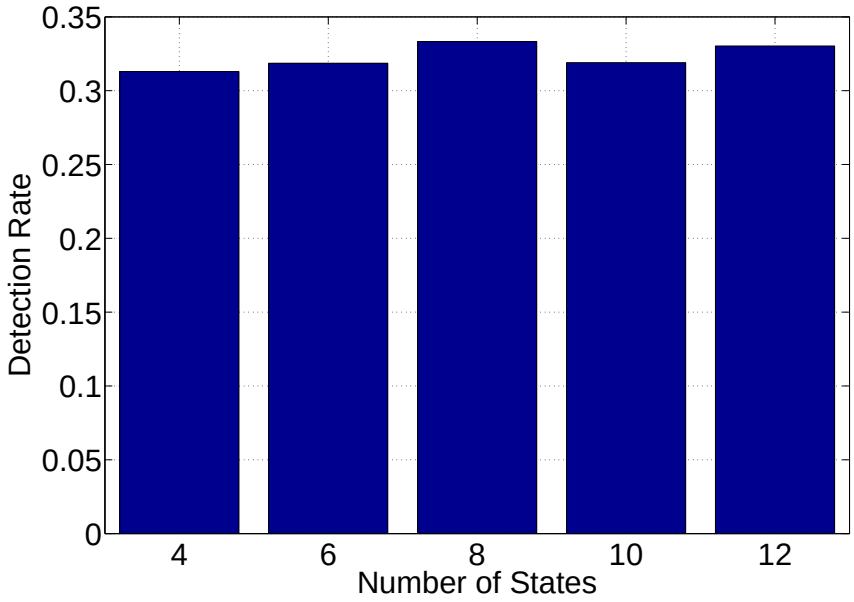


Figure 23: Detection Rate with Different Number of States

of states in HMM is heuristically set to be eight according to the length of through vectors. Following the principle of Occam's razor, the number of states should be small enough to avoid over-fitting and large enough to model the ergodic nature of VoIP calls. We get similar detection performance for different number of states when the number of states is larger than five, as shown in Figure 23. When the number of state is too large, the training of HMMs fails to converge to an optimal solution.

The HMM used in traffic analysis attacks is the left-right HMM [34] as shown in Figure 22. We choose the left-right model because of the nonergodic nature of speech signals [34], i.e., the attribute of signals whose properties change over time. Each node in Figure 22 represents a state in one sample interval. The observable variable is the throughput of each sample interval.

Two kinds of HMMs can be trained: (a) For the speech detection, we focus on detecting speeches made by one specific speaker, say Alice. So a speech-specific model can be obtained by training the model with traces of the same speeches made by Alice. (b) A speaker-specific model can be obtained by training the HMM with

traces of VoIP calls made by a specific speaker. The trained HMMs are used in the following speech detection or speaker detection.

4.2.4 Speech Detection and Speaker Detection

The inputs to this step are the Alice's speech-specific or the Alice's speaker-specific HMM trained in the previous step and throughput vectors generated from a candidate pool of raw VoIP traces of interest. The output of this step is the intermediate detection result. For the speaker detection, the intermediate detection result is K_{top} speakers from the candidate pool with talk pattens closest to Alice's talk pattern. For the speech detection, the intermediate detection result is K_{top} speeches from the candidate pool with speech pattens closest to talk patterns in training traces.

The detection step can be divided into two phases: (a) First, the likelihood of each throughput vector is calculated with the trained HMM. (b) The trace with the highest likelihood is declared as the trace generated from a specific speech by Alice if intersection attack is not used. To improve detection accuracy, the intermediate detection results can be fed into the optional step, intersection attack.

4.2.5 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 is similar as described in [36, 37, 38]: Instead of deciding the detection result based on one trial, we can improve detection accuracy by a number of trials and the final detection result is determined by combining (or intersecting) the results from each trial.

More specifically, for the proposed attacks, suppose it is possible to get m Skype

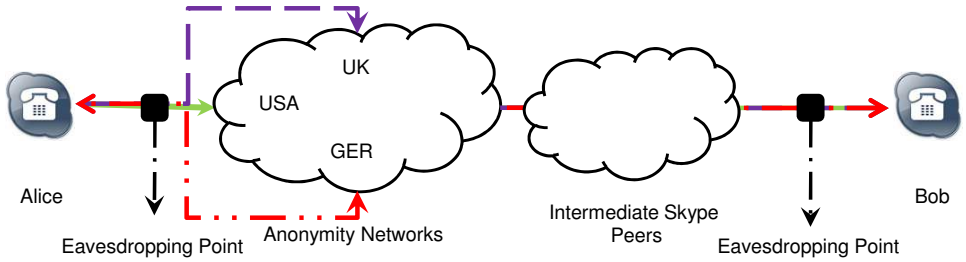


Figure 24: Experiment Setup

call traces made by the same speaker, the adversary can do m trials as described in Section 4.2.4. From each detection, the adversary can obtain k traces with the K_{top} highest likelihoods. The overall rank for each speaker is calculated by adding ranks in m trials. The speaker with the highest rank is determined to be Alice. Tie can be broken by comparing the sum of likelihood in m trials.

4.3 Empirical Evaluation

In this section, we evaluate the effectiveness of the proposed detections.

4.3.1 Experiment Setup

The experiment setup is as shown in Figure 24. Skype packets are first directed to the anonymity network managed by findnot.com and then relayed by Skype peers or supernodes before arriving at the other end of the call. We use the commercial anonymous communication services provided by findnot.com mainly because it is possible to select entry points into the anonymity network [39]. In our experiments, Skype packets are directed through entry points in England, Germany, and United States as shown in Figure 24. For these Skype calls made through anonymity networks, the end-to-end delay is at least $80ms$ and the two communication parties are at least 20 hops away from each other. About a quarter of calls are made through the campus network so that traces of VoIP calls over a wide range of networks are

available for our experiments.

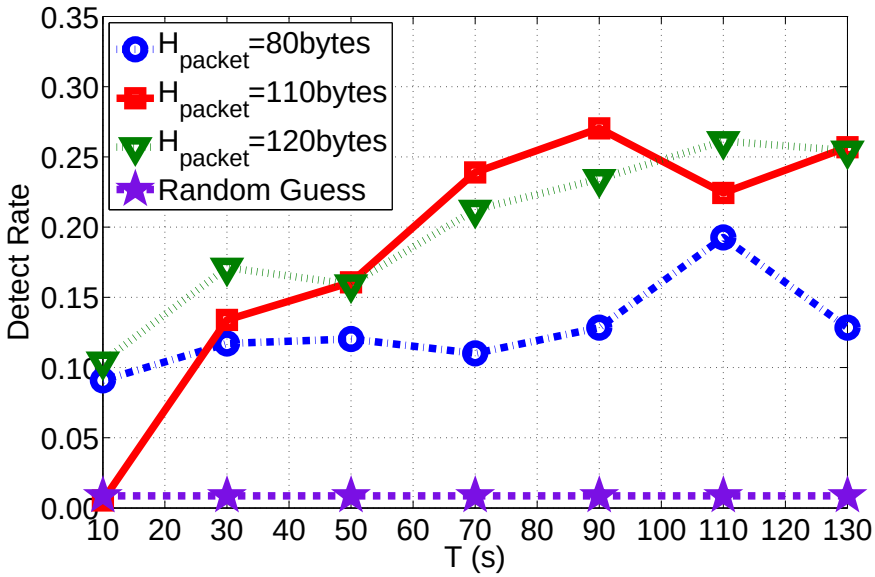
The audio signals are extracted from videos posted on Research Channels [40] for consistent sound quality. The length of extracted audio signals is about 38.5 minutes. At least three different speeches are available for most speakers and each speech is sent through at least four different network entry points[1]. In total 116 Skype calls are made through different entry points of the anonymity network managed by findnot.com and through the campus network.

4.3.2 Metrics

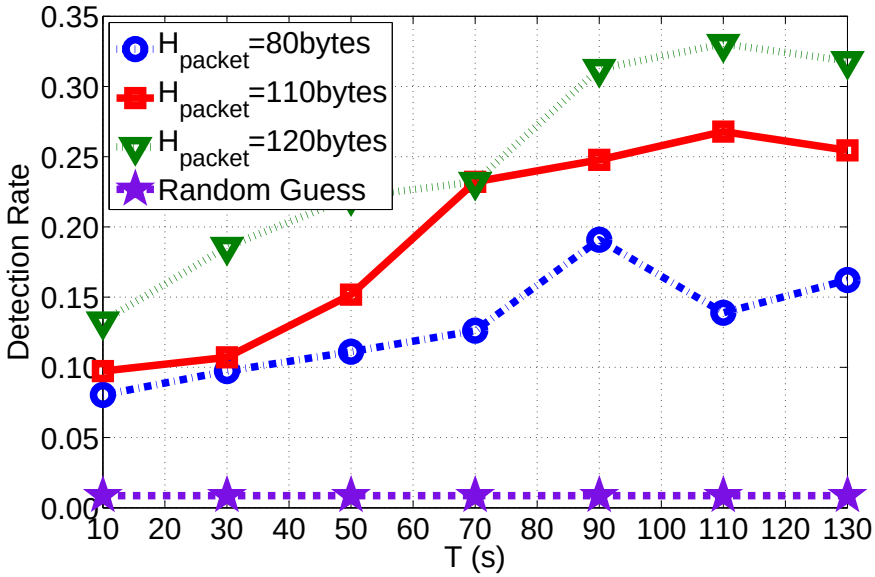
We use detection rate to measure effectiveness of the proposed attacks. In this chapter, the detection rate is defined as the ratio of the number of successful detections to the number of attempts.

For both speech detection and speaker detection, the detection rate for random guess is about $\frac{1}{105}$, because in each trial, there are 105 candidate traces in the pool on average. One of the traces in the pool is the "right" trace, i.e., the trace generated by a specific speech or speaker. In each trial of the speech detection, three traces of the same speech are used for training and one trace of the same speech is one of the candidate traces. In each trial of the speaker detection, one trace of Alice's speech is used as one of the candidate traces and Alice's other traces are used for training.

In all the experiments below, the training traces and candidate traces are all collected from *different* Skype calls. For better training, all the traces used in training are collected from the sending end, i.e., from the link connected to Alice's computer.

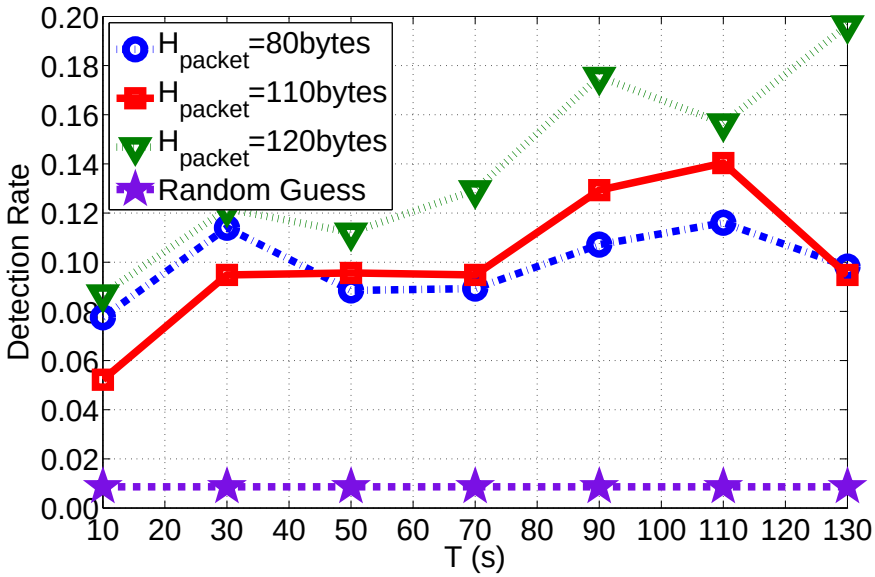


(a) Candidate Traces Collected from Sending Side

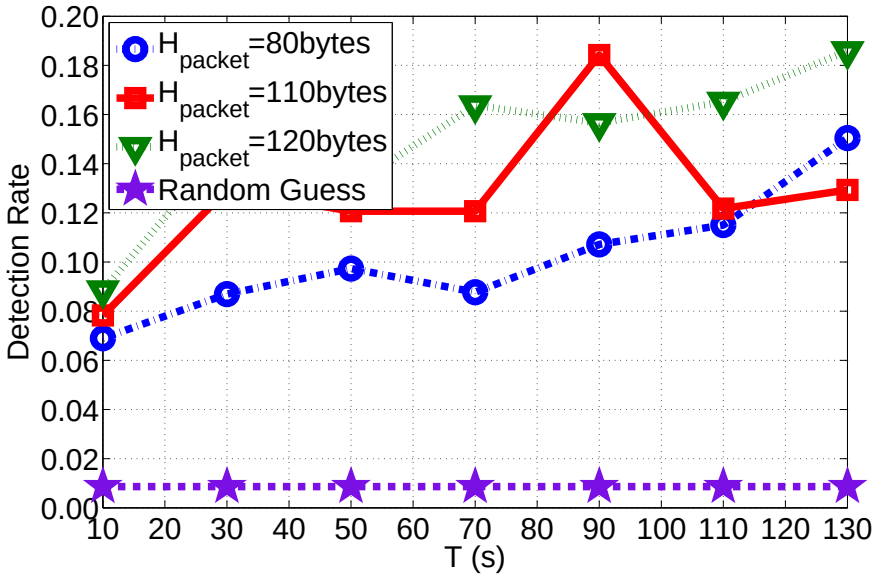


(b) Candidate Traces Collected from Receiving Side

Figure 25: Effect of Parameter T on Speech Detection



(a) Candidate Traces Collected from Sending Side



(b) Candidate Traces Collected from Receiving Side

Figure 26: Effect of Parameter T on Speaker Detection

4.3.3 Effect of Parameter T (Length of Sample Interval)

This series of experiments are designed to test the effect of the parameter T, the length of sample intervals.

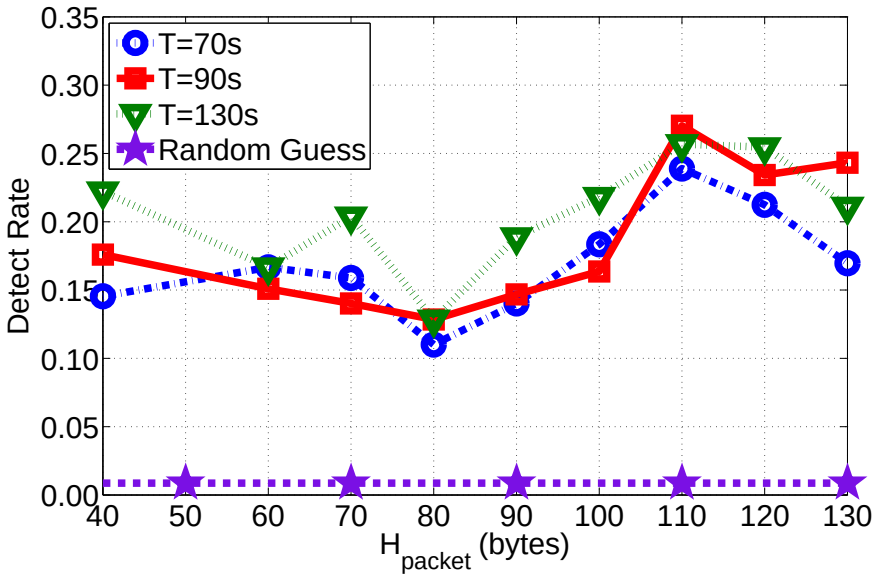
Figure 25 shows the effect of the parameter T on speech detection. From these two figures, we can observe: (a) For a wide range of T, the detection rate is larger than 0.1, about 10-fold improvement over random guess. (b) When T is small, the detection rate is relatively low. It is because a small T cannot be used to extract talk patterns usually in the order of second as discussed in Section 4.2.1. (c) When T becomes large, the detection rate may drop simply because shorter throughput vectors are used for training and detection. (d) The detection rate can be as high as 0.33, about 35-fold improvement over random guess. (e) 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 T is big enough to filter out network dynamics at receiving end which can vary from call to call. Similar observations can be made from Figure 26. The detection rate for speaker detection can reach 0.18, about 18-fold improvement over random guess.

4.3.4 Effect of Parameter H_{packet} (Threshold on Packet Size)

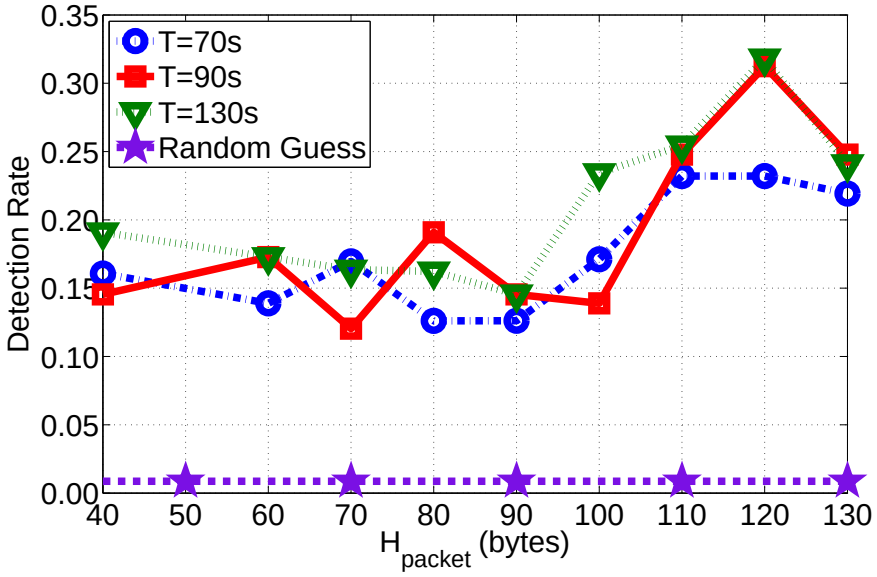
This series of experiments are designed to test the effect of the parameter H_{packet}, the threshold on packet size.

Figure 27 shows the effect of the parameter H_{packet} on speech detection. From Figure 27, we can observe: (a) When H_{packet} is less than 100 bytes, the detection rate is low. We believe it is because small H_{packet} cannot be used to remove all signaling packets. (b) When H_{packet} is larger than 130 bytes, the detection rate may decrease. The reason is too few packets left because of the larger threshold. (c) The detection

[1]The campus network entry point is one of the choices.

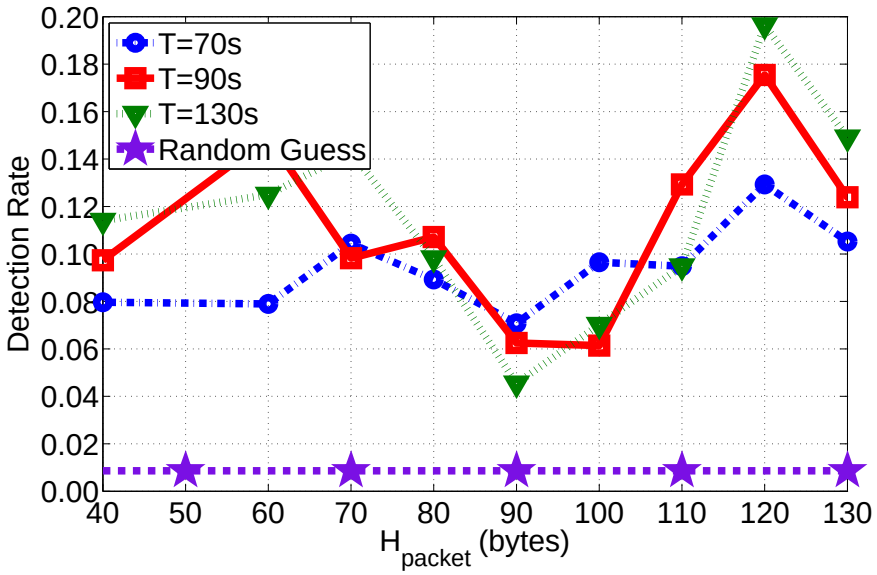


(a) Candidate Traces Collected from Sending Side

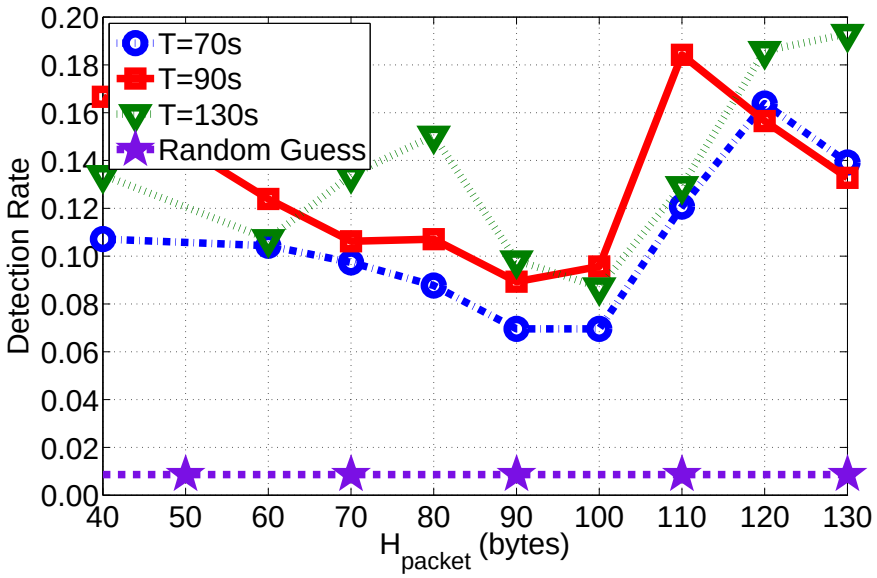


(b) Candidate Traces Collected from Receiving Side

Figure 27: Effect of Parameter H_{packet} on Speech Detection



(a) Candidate Traces Collected from Sending Side



(b) Candidate Traces Collected from Receiving Side

Figure 28: Effect of Parameter H_{packet} on Speaker Detection

rate for speech detection can achieve 0.33, about 35-fold improvement over random guess.

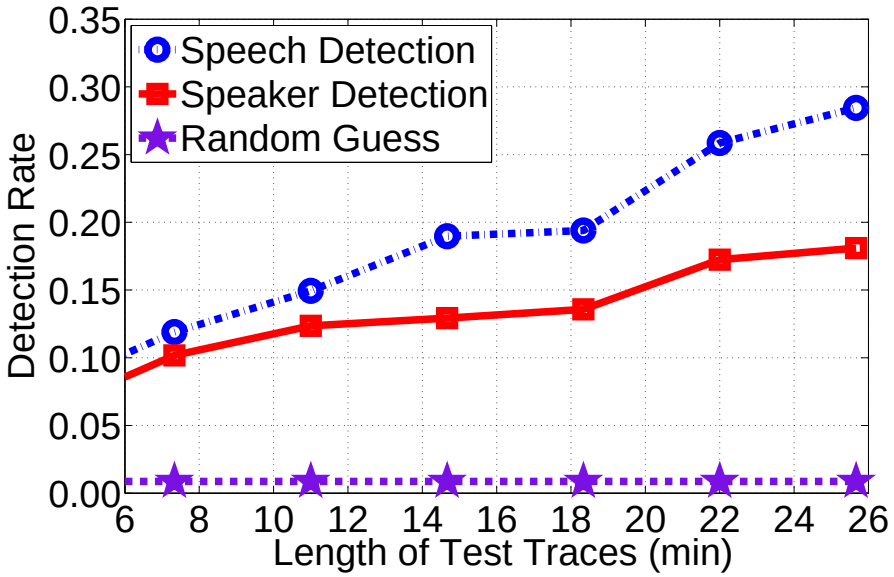
Figure 28 shows the effect of the parameter H_{packet} on speaker detection. We can observe: (a) The best range of H_{packet} for speaker detection is from 110 bytes to 130 bytes. (b) The detection rate can reach to 0.2, about 20-fold improvement over random guess.

4.3.5 Length of Training Traces and Test Traces

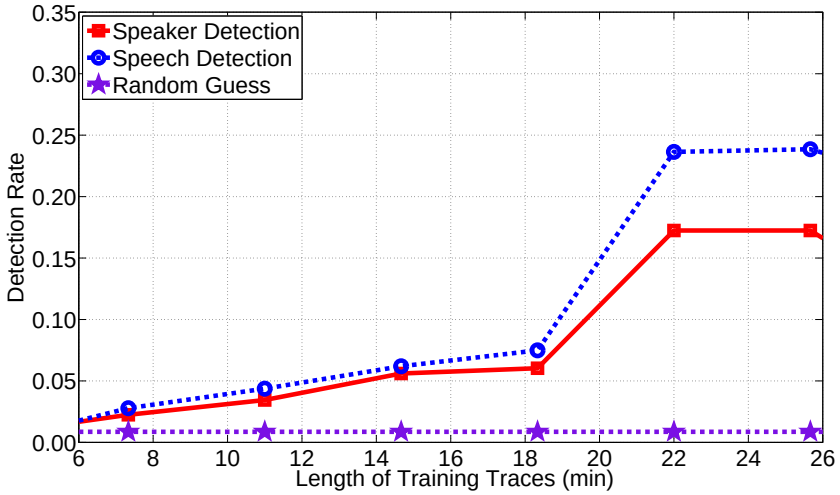
The length of training traces and test traces available for traffic analysis largely determines the effectiveness of proposed traffic analysis. In this set of experiments, we evaluate performance of the proposed attacks with different length of training traces and test traces. Figure 29 shows the experiment results on length of test traces. The results are obtained with training traces of length 38.5 minutes, $T = 110s$, and $H_{packet} = 120bytes$. We can observe that the detection rates for both speech detection and speaker detection increase with the length of test traces. When test traces are 25 minutes long, the detection rates for speech detection and speaker detection are 0.28 and 0.18, about 28-fold and 18-fold improvement over random guess respectively.

4.3.6 Pool Size

In this set of experiments, we investigate the performance of traffic analysis attacks with different size of candidate pools. From the experiment results shown in Figure 30, we can observe that when the pool size increases, the detection rate slightly decreases for both speech detection and speaker detection, since it is harder to find the right one from a larger candidate pool. But the ratio between the speech detection rate and the corresponding random guess rate changes from 12.59, when pool size is 27, to 35 when pool size is 105, meaning the traffic analysis attacks are



(a) Test Time



(b) Training Time

Figure 29: Detection Rate vs Test Time and Training Time on Speech Detection and Speaker Detection

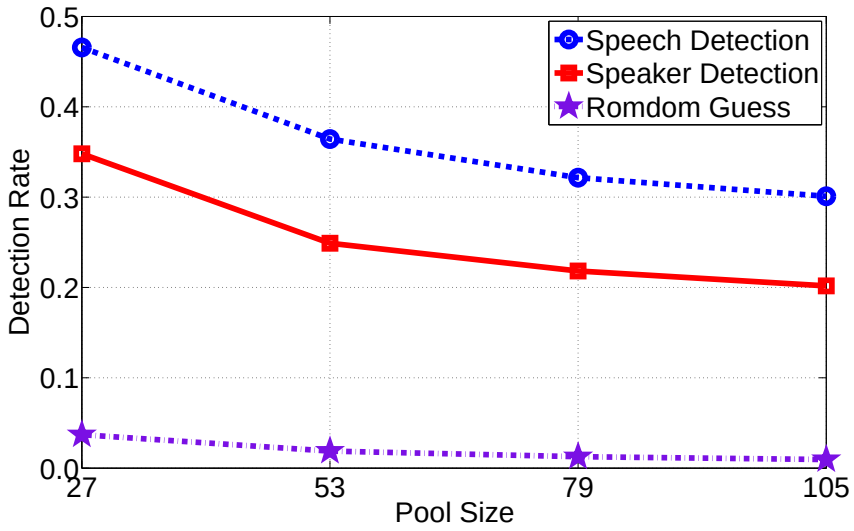


Figure 30: Detection Performance with Different Pool Size

more effective when the pool size is large.

4.3.7 Intersection Attack

In this set of experiments, we evaluate the effectiveness of intersection attacks on speaker detection. On average, there are 33 candidate speakers. So the detection rate for random guess is about $\frac{1}{33}$. Each candidate speaker has 3 Skype traces available for detection so the final detection result is obtained by combining the intermediate detection results of three trials.

From previous experiments, we learned specify explicitly suitable ranges for parameters T and H_{packet} to achieve higher detection rate. We use parameters in these ranges in the intersection attacks described below and parameters used in the following experiments are given in figures.

Figure 31 shows the performance of the intersection attack. From Figure 31, we can observe: (a) In general, when K_{top}, the number of most likely candidates selected from each trial, increases, the detection rate increases because more high-

likelihood traces are considered in the intersection attack step. (b) The detection rate can reach 0.44, about 15-fold improvement over random guess. (c) The detection rate for candidate traces collected from the sending end is again comparable with the detection rate for candidate traces collected from the receiving end.

In summary, the proposed traffic analysis attacks can significantly improve the detection rate over random guess. We believe that given more training traces, higher detection rates can be achieved.

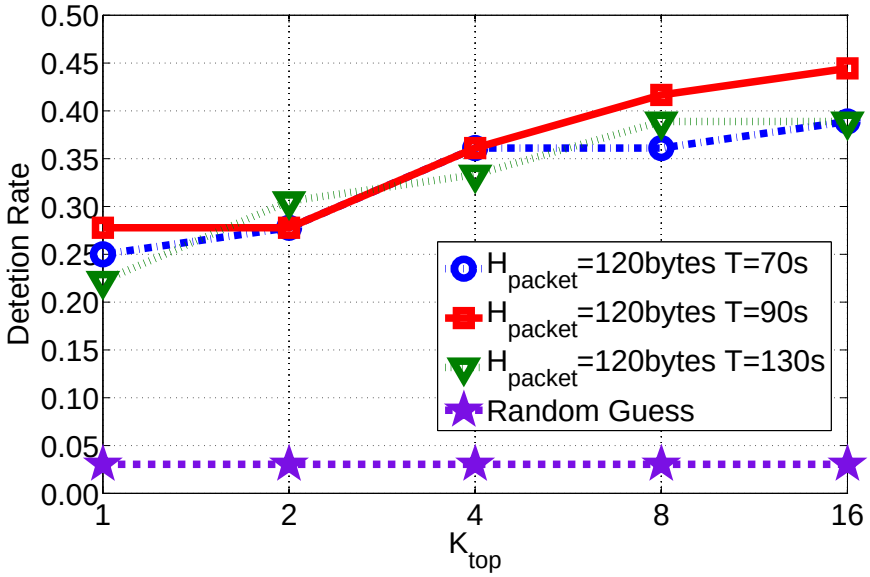
4.4 A Countermeasure and Its Performance

From the discussion above, it is apparent that the proposed traffic analysis attacks can greatly compromise the privacy of Skype calls. Countermeasures are needed to protect privacy of Skype calls. In this section, we introduce a countermeasure which can protect privacy at the cost of marginally effect on quality of VoIP calls.

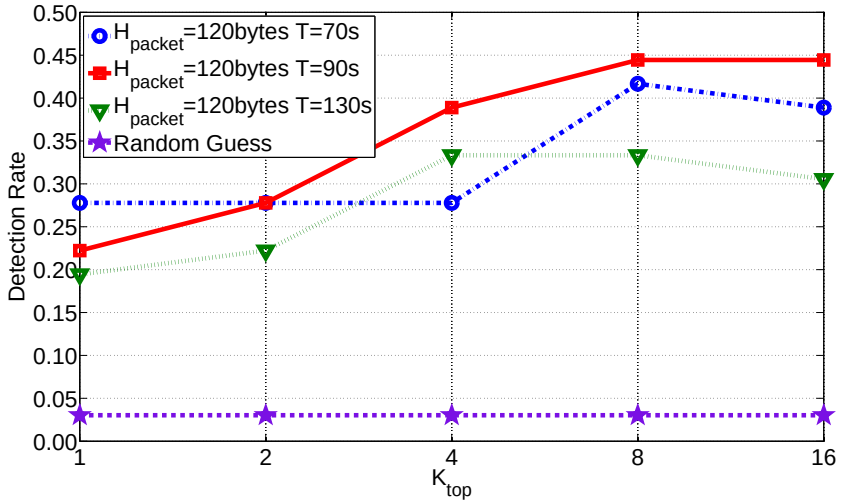
4.4.1 Overview

The main idea of the countermeasure is to camouflage Alice's Skype packets according to another speaker's traces. As shown in Figure 32, Alice's Skype packets are re-packetized according to packet sizes of another speaker's Skype packets. The re-packetization is controlled by the byte tokens generated according to packet size of Speaker X's Skype packets: when it is time to send Speaker X's Skype packet of size v-byte, a v-byte token is generated to signal the re-packetization module to allow v-byte Skype payload stored in buffer to be transmitted.

Another possible countermeasure is to pad all the packets to the same size. We do not propose this countermeasure because: (a) A significant amount of bandwidth



(a) Candidate Traces Collected from Sending Side



(b) Candidate Traces Collected from Receiving Side

Figure 31: Performance of Intersection Attack

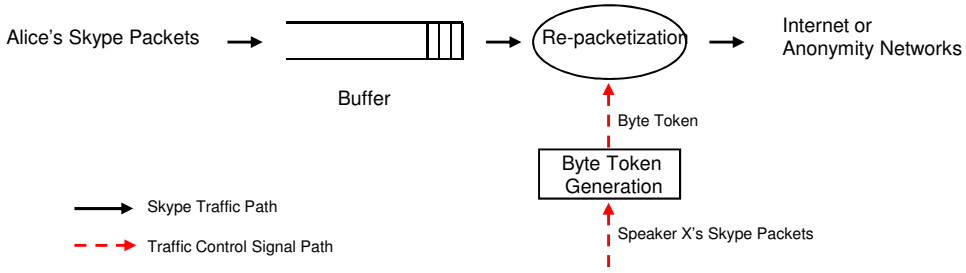


Figure 32: Countermeasure: Camouflaging Alice's Skype Packets

can be wasted to send padding bits. (b) Skype flows of constant packet sizes may catch special interest from adversaries.

4.4.2 Performance Evaluation of The Countermeasure

We evaluate the countermeasure with two metrics: (a) The detection rate defined in Section 4.3.2: It is used to measure the performance of preserving privacy of Skype calls. (b) Packet delay caused by the countermeasure: We use it to measure the degradation of quality of VoIP calls.

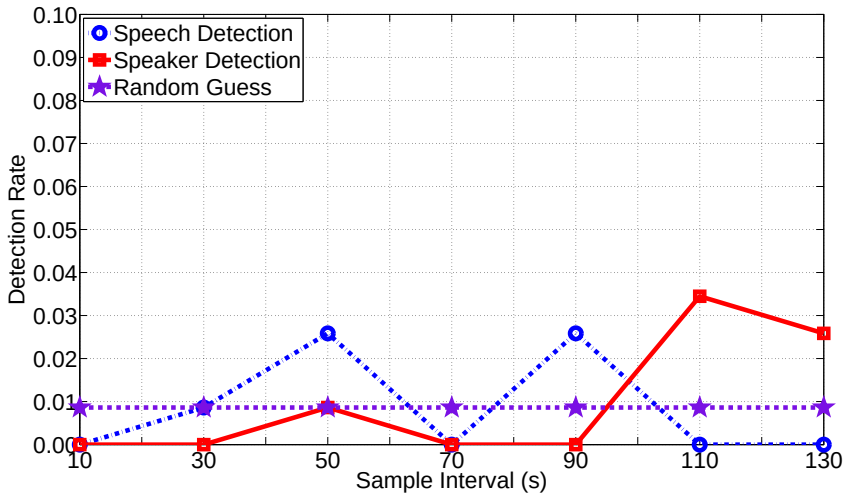
In this set of experiments, we use real traces collected from the experiment environment described in Section 4.3.1.

Figure 33 shows the performance of the countermeasure. Figure 33(a) shows that the countermeasure can preserve the privacy of Skype calls since the detection rate is around the probability of random guess. Figure 33(b) shows the distribution of packet delay caused by the countermeasure. The mean of the delay caused by the countermeasure is $0.10ms$. The delay is less than $0.102ms$ with a probability larger than 0.95. So the delay caused by the countermeasure is negligible. In other words, the countermeasure will not cause any significant change in the quality of Skype calls since it is much less than the delay budget for VoIP calls [41].

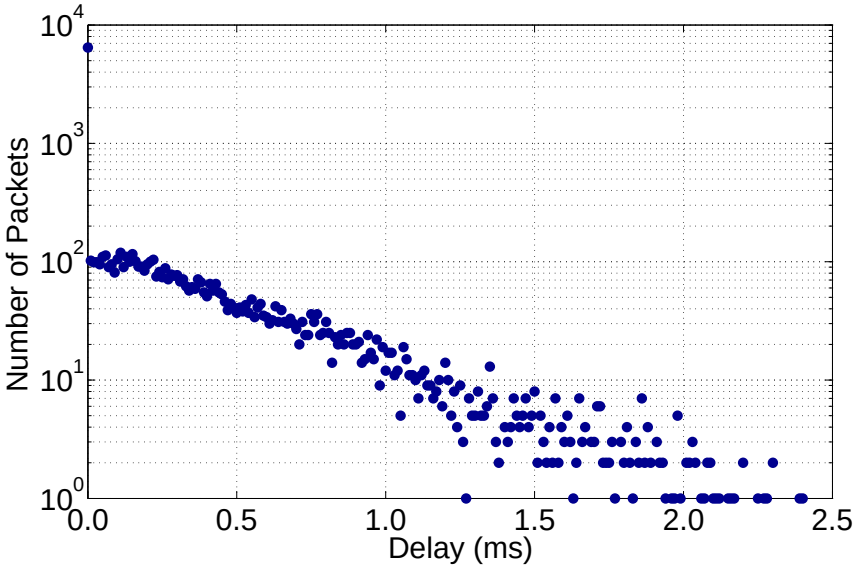
4.5 Discussion and Future Work

Our experiments clearly show that the proposed traffic analysis attacks can greatly compromise privacy of Skype calls. The detection rates for speech detection and speaker detection are 35-fold and 15-fold improvement over random guess. Higher detection rate can be achieved by adding more training traces.

The traditional speaker detection problem assuming access to speech signals has been well studied [50]. In this chapter only noisy talk patterns recovered from



(a) Detection Rate



(b) Distribution of Packet Delay Caused by Countermeasure

Figure 33: Performance of the Proposed Countermeasure

packet sizes are available for traffic analysis. In [49], a speaker detection approach based on face, mouth motion and silence detection is proposed. In comparison with the 90% high detection rate achieved in [49], our detection rate is relatively low since fewer features are available for traffic analysis.

The framework proposed in this chapter, including extracting application-level features from network traffic traces and statistical analysis of extracted application-level feature by the HMM, can be potentially used to infer other sensitive information at application-level. For example, the framework can be potentially used to detect speaker's emotion during a call suppose the speaker's talk behavior can change significantly when the speaker's mood changes. One of our future works is to explore the potential of the framework.

CHAPTER V

CONCLUSIONS

In this thesis, we propose a class of passive traffic analysis attacks to compromise privacy of Skype calls and SIP-Based VoIP calls. The proposed attacks are based on application-level features extracted from VoIP call traces. The proposed attacks are evaluated by extensive experiments over different types of networks including commercialized anonymity networks and our campus network. The experiments show that the proposed traffic analysis attacks can detect speeches and speakers of SIP based VoIP calls with 0.65 and 0.32 detection rate respectively, about 70-fold and 35-fold improvement over random guess. For Skype calls, the speech detection rate and speaker detection rate are 0.33 and 0.44, about 30-fold and 15-fold improvement over random guess. Countermeasures are proposed to mitigate the proposed traffic analysis attacks by camouflaging. The proposed countermeasures can largely mitigate the traffic analysis attacks and does not cause significant degradation on quality of VoIP calls.

BIBLIOGRAPHY

[1] Zimmermann P., A.J.E., Callas, J.: ZRTP: Media path key agreement for secure rtp draft-zimmermann-avt-zrtp-11. RFC (2008)

[2] Baugher, M., McGrew, D., Naslund, M., Carrara, E., Norrman, K.: The secure real-time transport protocol. RFC (2004)

[3] Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceeding of the 13th USENIX Security Symposium. (August 2004) 303–320

[4] Berthold, O., Federrath, H., Köpsell, S.: Web MIXes: A system for anonymous and unobservable Internet access. In Federrath, H., ed.: Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, Springer-Verlag, LNCS 2009 (July 2000) 115–129

[5] Levine, B.N., Reiter, M.K., Wang, C., Wright, M.K.: Timing attacks in low-latency mix-based systems. In Juels, A., ed.: Proceeding of Financial Cryptography (FC '04), Springer-Verlag, LNCS 3110 (February 2004) 251–265

[6] Zhu, Y., Fu, X., Graham, B., Bettati, R., Zhao, W.: On flow correlation attacks and countermeasures in mix networks. In: Proceedings of Privacy Enhancing Technologies workshop (PET 2004). Volume 3424 of LNCS. (May 2004) 207–225

[7] Danezis, G.: The traffic analysis of continuous-time mixes. In: Proceeding of Privacy Enhancing Technologies Workshop (PET 2004). Volume 3424 of LNCS. (May 2004) 35–50

[8] Murdoch, S.J., Danezis, G.: Low-cost traffic analysis of Tor. In: Proceedings of the 2005 IEEE Symposium on Security and Privacy, IEEE CS (May 2005) 183–195

[9] Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2) (1981) 84–90

[10] Pfitzmann, A., Pfitzmann, B., Waidner, M.: ISDN-mixes: Untraceable communication with very small bandwidth overhead. In: Proceedings of the GI/ITG Conference on Communication in Distributed Systems. (February 1991) 451–463

[11] Rennhard, M., Plattner, B.: Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In: Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2002). (November 2002) 91–102

[12] Bennett, K., Grothoff, C.: GAP – practical anonymous networking. In Dingledine, R., ed.: Proceedings of Privacy Enhancing Technologies workshop (PET 2003), Springer-Verlag, LNCS 2760 (March 2003) 141–160

[13] Freedman, M.J., Morris, R.: Tarzan: A peer-to-peer anonymizing network layer. In: Proceeding of the 9th ACM conference on Computer and communications security, New York, NY, USA, ACM Press (2002) 193–206

[14] Goldschlag, D.M., Reed, M.G., Syverson, P.F.: Hiding routing information. Information Hiding (May 1996) 137–150

[15] Xiaodong, D., David, S., Tian, W.X.: Timing analysis of keystrokes and timing attacks on ssh. In: Proceedings of The Tenth USENIX Security Symposium. (2001) 337–352

[16] Sun, Q., Simon, D.R., Wang, Y.M., Russell, W., Padmanabhan, V.N., Qiu, L.:

Statistical identification of encrypted web browsing traffic. In: IEEE Symposium on Security and Privacy, Society Press (2002) 19–30

[17] Saponas, T.S., Lester, J., Hartung, C., Agarwal, S., Kohno, T.: Devices that tell on you: privacy trends in consumer ubiquitous computing. In: SS'07: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, USENIX Association (2007) 1–16

[18] R. J. P. de Figueiredo: Signal and image processing (sip 2007). In: Proceedings of the IASTED International Conference, IASTED/ACTA Press (2007)

[19] Wu, W., Bulut, H., Fox, G.C., Uyar, A.: Adapting h.323 terminals in a service-oriented collaboration system. IEEE Internet Computing **9**(4) (July 2005) 43–50

[20] Audio-Video Transport Working Group, Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications. RFC 1889 (Proposed Standard) (jan 1996) Obsoleted by RFC 3550.

[21] Vonage. Available:`http://www.vonage.com`

[22] AT&T CallVantage. Available:`http://www.usa.att.com/callvantage/consumer_redirect.jsp`

[23] Brady, P.T.: A technique for investigating on-off patterns of speech. The Bell System Technical Journal **44** (1965) 1–22

[24] The Speex projectpage. http://www.speex.org (2005)

[25] Henning, S.: Voice communication across the internet: a network voice terminal. Coins technical report, University of Massachusetts at Amherst, Dept. of Computer and Information Science (1992)

[26] Union., I.T.: Coding of speech at 8 kbit/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 telecommunication standardization sector of itu. Technical report (November 1996)

[27] Police reveal the identity of shooting suspect. Available:`http://www.cnn.com/2006/US/09/29/school.shooting/index.html` (2006)

[28] X-Lite 3.0 FREE Softphone. Available:`http://www.xten.com/index.php?menu=Products\&smenu=xlite`

[29] Wang, X., Chen, S., Jajodia, S.: Tracking anonymous peer-to-peer voip calls on the internet. In: Proceedings of the ACM Conference on Computer and Communications Security. (November 2005) 81–91

[30] Pyun, Y.J., Park, Y.H., Wang, X., Reeves, D.S., Ning, P.: Tracing traffic through intermediate hosts that repacketize flows. In: Proceedings of IEEE INFOCOM '07. (May 2007) 634–642

[31] Rathinavelu, C., Deng, L.: Hmm-based speech recognition using state-dependent, linear transforms on mel-warped dft features. In: ICASSP '96: Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference, Washington, DC, USA, IEEE Computer Society (1996) 9–12

[32] Schambach, M.P.: Determination of the number of writing variants with an hmm based cursive word recognition system. In: ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition, IEEE Computer Society (2003) 119

[33] Deng, J., Tsui, H.: An hmm-based approach for gesture segmentation and recognition. Volume 3., Los Alamitos, CA, USA, IEEE Computer Society (2000) 3683

[34] Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. (1990) 267–296

[35] Bakis, R.: Continuous speech recognition via centisecond acoustic states. Acoustical Society of America Journal **59** (1976)

[36] Berthold, O., Pfitzmann, A., Standtke, R.: The disadvantages of free MIX routes and how to overcome them. In Federrath, H., ed.: Proceeding of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, Springer-Verlag, LNCS 2009 (July 2000) 30–45

[37] Danezis, G., Serjantov, A.: Statistical disclosure or intersection attacks on anonymity systems. In: Proceeding of 6th Information Hiding Workshop (IH 2004). LNCS (May 2004) 293–308

[38] Berthold, O., Langos, H.: Dummy traffic against long term intersection attacks. In Dingledine, R., Syverson, P., eds.: Proceeding of Privacy Enhancing Technologies Workshop (PET 2002), Springer-Verlag, LNCS 2482 (April 2002) 110–128

[39] FindnotProxyList: Available:`http://www.findnot.com/servers.html`.

[40] ResearchChannels: Available:`http://www.researchchannel.org`.

[41] Szigeti, T., Hattingh, C.: End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs (Networking Technology). Cisco Press (2004)

[42] Baset, S., Schulzrinne, H.: An analysis of the skype peer-to-peer internet telephony protocol. In: Proceedings of 25th IEEE International Conference on Computer Communications. (INFOCOM 2006). (April 2006) 1–11

[43] Berson, T.: Skype security evaluation. ALR-2005-031, Anagram Laboratories (18 October 2005)

[44] Chen, K.T., Huang, C.Y., Huang, P., Lei, C.L.: Quantifying skype user satisfaction. (2006) 399–410

[45] Perényi, M., Molnár, S.: Enhanced skype traffic identification. In: ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools, ICST, Brussels, Belgium, Belgium, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2007) 1–9

[46] Wang, X., Chen, S., Jajodia, S.: Network flow watermarking attack on low-latency anonymous communication systems. In: SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy, IEEE Computer Society (2007) 116–130

[47] John, T., Halina, R.: Effect of pre-utterance pause length on perceptions of communicative competence in aac-aided social conversations. (2003) 222–234

[48] Jianfen, C.: Speaking rate and its variations. (2003) 222–234

[49] Pavlovic, V., Rehg, J.M., Garg, A., Huang, T.S.: Multimodal speaker detection using error feedback dynamic bayesian networks. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **2** (2000) 2034

[50] Campbell, J.P.: Speaker recognition: a tutorial. Proceedings of the IEEE **85**(9) (1997) 1437–1462