7-1-2011

Traffic Analysis Attacks on Skype VoIP Calls

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

Huirong Fu
Oakland University

Publisher's Statement

NOTICE: this is the author's version of a work that was accepted for publication in Computer Communications. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Computer Communications, 34, 10, (07-01-2011); 10.1016/j.comcom.2010.12.007

Traffic analysis attacks on Skype VoIP calls

Ye Zhu [a,*], Huirong Fu [b]

[a] Cleveland State University, 2121 Euclid Avenue, Cleveland, OH 44115, USA
[b] Oakland University, Rochester, MI 48309-4478, USA

1. Introduction

In this paper, we address on privacy issues of Skype calls. 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. 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 routed through Skype peers during the call for better Quality of Service (QoS) [1,2]. 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 [3], proprietary protocols [3], unknown codecs [4], and dynamic path selection[1] [1,2], and the constant packet rate [5]. To further protect privacy of Skype VoIP calls, advanced users are using anonymity networks to anonymize VoIP calls. For this purpose, low-latency anonymity networks such as Tor [6] and JAP [7] can be used.

In this paper, we propose a class of passive traffic analysis attacks to compromise privacy of Skype calls. The procedure of the proposed attacks is as follows: First an adversary collects 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.

The contributions made in this paper are summarized as follows:

- We propose a class of traffic analysis attacks to compromise privacy of Skype calls. The attacks are passive and based on the HMM, a powerful tool to model temporal data. We also propose a method to extract application-level features from traffic flows for application-level traffic analysis attacks.
- We evaluate the proposed traffic analysis attacks through extensive experiments over the Internet and commercial anonymity networks. For most of Skype calls made in the experiments, the two parties are at least 20 hops away and the end-to-end delay between two parties is at least 80 ms. Our experiments show that the traffic analysis attacks are able to detect speeches or speakers of Skype calls with high probabilities.
- We propose intersection attacks to improve the effectiveness of the proposed attacks.
- We propose a countermeasure to mitigate the proposed traffic analysis attacks and analyze the effect of the countermeasure on quality of Skype calls.

The rest of the paper is organized as follows: Section 2 reviews related work. In Section 3, we formally define the problem. The details of proposed traffic analysis attacks are described in Section 4. In Section 5, we evaluate the effectiveness of the proposed traffic analysis attacks with experiments on commercialized anonymity

networks and our campus network. Section 7 presents a countermeasure to mitigate the proposed traffic analysis attacks. Discussion and the outline of future work are given in Section 8. We conclude the paper in Section 9.

2. Related work

In this section, 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 [8], many low-latency anonymity networks have been proposed or even implemented for different applications. The examples are *ISDN-mixes* [9] for telephony, *Web Mix* [7] for web browsing, *MorphMix* [10] for peer-to-peer applications, *GAP* base *GNUnet* [11] for file sharing. TARZAN [12], *Onion Router* [13], and *Tor* [6], 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 our experiments, 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 attacks by Levine et al. [14] on anonymity networks, the active attack proposed by Murdoch and Danezis [15] on the Tor network, our flow correlation [16], and our flow separation [17] attacks.

Application-level traffic analysis attacks target at disclosing application-level information. The examples are keystroke detection based on packet timing [18], web page identification [19], spoken phrase identification [20] with variable bit rate codecs.

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

There are a number of research efforts focusing on traffic analysis of VoIP. Wang et al. [24] proposed to watermark VoIP traffic flows to trace VoIP calls through the Internet. In [21], Wright et al. showed that it was possible to recover spoken phrases from VoIP packet size information. Wright et al. [22] also showed the feasibility to detect languages used in VoIP conversations based on VoIP packet size information.

Similar as [21,22], our research in this paper focuses on disclosing application-level information from traffic analysis of VoIP. The traffic analysis attacks proposed in this paper aim to identify speakers of VoIP calls. Another difference is on the type of VoIP codecs and protocols. The researches in [21,22] focus on a variable bit rate (VBR) codec, more specifically the open-source Speex codec [23], and standardized VoIP protocols. We focus on the Skype VoIP service which uses codecs unknown to the public and its own proprietary protocols. Skype is also known for its strong encryption preventing packet content access. These privacy protection measures taken by Skype render traffic analysis on Skype VoIP traffic more difficult since (1) we have to treat the Skype software as a black box and (2) we are not even able to identify signaling packets so that these signaling packets can be completely removed before traffic analysis.[2]

3. Problem definition

In this paper, 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 paper 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 paper, 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 location or with a different computer.

3.1. Network model

In the paper, 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 [1]. 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 [4]. (d) Skype calls are encrypted and hard to decipher [3]. (e) Skype sends packets at the constant rate of 33 packet/s [5]. Due to the unique set of features listed above, Skype is known as secure voice

[2] In general, signaling packets are not affected by talk patterns so signaling packets are essentially "noise" in recovering talk patterns from Skype traffic. Signaling packets are not considered in [21,22] since these packets can be filtered out easily for standardized VoIP calls. In other words, patterns recovered from VoIP traffic in [21,22] are noise-free.

communication [3] which can protect privacy of communication parties.

As shown in Fig. 1, 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.

3.2. Threat model

We focus on passive attacks in this paper. In other words, the attacks launched by the adversary do not disturb the existing network traffic. In comparison with active traffic analysis attacks [15,24], 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 [15]. 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. 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.

4.1. Overview

The proposed traffic analysis attacks are based on packet size information. A simple experiment shown in Fig. 2 indicates that packet size information can disclose speech-level information. Fig. 2(a) shows an audio signal with three silence periods. Fig. 2(b) shows the packet sequence generated by feeding the audio signal into Skype clients. From the packet sequence plotted in Fig. 2(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 paper mainly because Skype clients send VoIP packets at a constant rate [5].

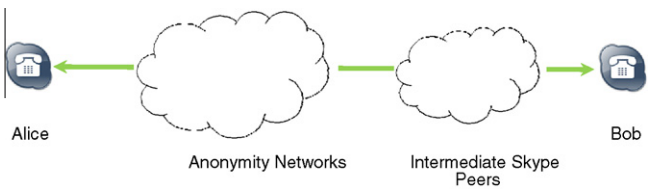


Fig. 1. Network model.

One of the challenges in this paper 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 $[s_1, s_2, \ldots, s_n]$, where n is the length of the vector. The element s_i in the throughput vector is calculated as follows:

$$s_i = \frac{\text{sum of bytes received or sent during the } i\text{th sample interval}}{T} \quad (1)$$

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 [25]. 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.

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 [26], handwriting recognition [27], and gesture recognition [28]. In the proposed attacks, HMMs are trained to model talk patterns.

The proposed attacks can be divided into two phases: The training phase and the detection phase as shown in Fig. 3. The two steps in the training phase are feature extraction and HMM 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.

4.2. Feature extraction

The input and output of the feature extraction step are raw traces of Skype 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 4.1, 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 5.

4.3. HMM training

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

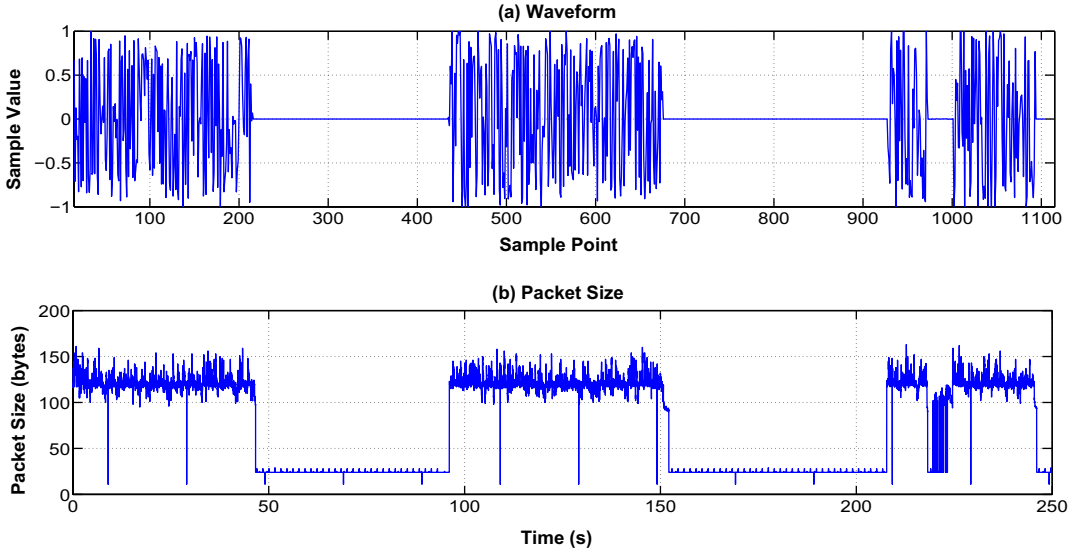


Fig. 2. An example.

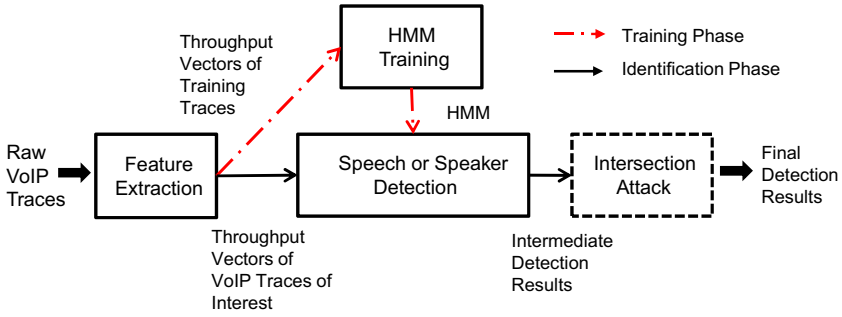


Fig. 3. Steps of the proposed attacks.

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. A nice introduction of Hidden Markov Model can be found in [29].

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 [30]. As shown in Fig. 2, 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 [30]. Because of the noise in pattern recovery, more states of different combinations of on–off periods are used in the HMM. We heuristically set the number of states in the HMM to be eight according to the length of throughput vectors. The HMM used in traffic analysis attacks is the left–right HMM [29] as shown in Fig. 4. The left–right model, also called as a Bakis model [31,32], has the property that the stat index is non-decreasing with the time. In other words,

$$a_{i,j} = 0, \quad \text{when } j < i \tag{2}$$

where $a_{i,j}$ denotes the state transition probability from the ith state to the jth state and the zero transition probability means that the transition from the ith state to the jth state is prohibited if $j < i$ as shown in Fig. 4. The left–right model also requires that

$$\pi_i = \begin{cases} 1, & i = 1 \\ 0, & i \neq 1 \end{cases} \tag{3}$$

where π_i denotes the initial state probability for the ith state. In other words, the left–right model mandates that the state sequence starts from the first state. We choose the left–right model because of the nonergodic nature of speech signals [29], i.e., the attribute of signals whose properties change over time. Each node in Fig. 4 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

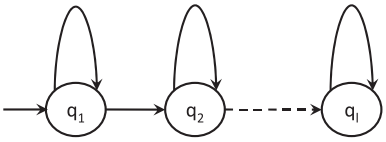


Fig. 4. A left–right Hidden Markov Model.

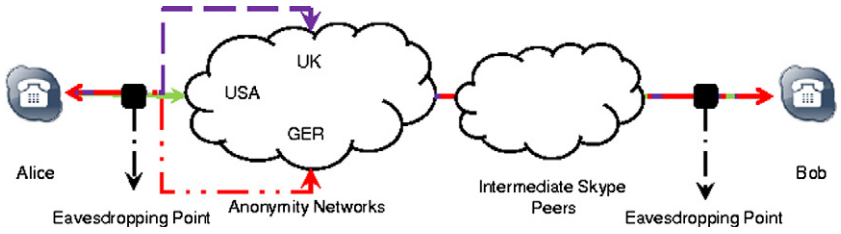


Fig. 5. Experiment setup.

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

4.4. Speech detection and speaker detection

The inputs to this step are the Alice's speech-specific or 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 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 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.

4.5. 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 [33–35]: 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 call traces made by the same speaker, the adversary can do m trials as described in Section 4.4. 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.

5. Empirical evaluation

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

5.1. Experiment setup

The experiment setup is as shown in Fig. 5. 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 [36]. In our experiments, Skype packets were directed through entry points in England, Germany, and United States as shown in Fig. 5. For these Skype calls made through anonymity networks, the end-to-end delay is at least 80 ms and the two communication parties are at least 20 hops away from each other. About a quarter of calls are made through 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 [37] for consistent sound quality. The length of extracted audio signals about 38 min. At least three different speeches are available for most speakers and each speech was sent through at least four different network entry points.[3] In total 180 Skype calls were made through different entry points of the anonymity network managed by findnot.com and through the campus network.

5.2. Metrics

We use detection rate to measure effectiveness of the proposed attacks. In this paper, 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 $\frac{1}{169}$, because in each trial, there are 169 candidate traces in the pool on average. One of the traces in the pool is the correct 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 speech is used as one of the candidate traces and Alice's other traces are used for training.

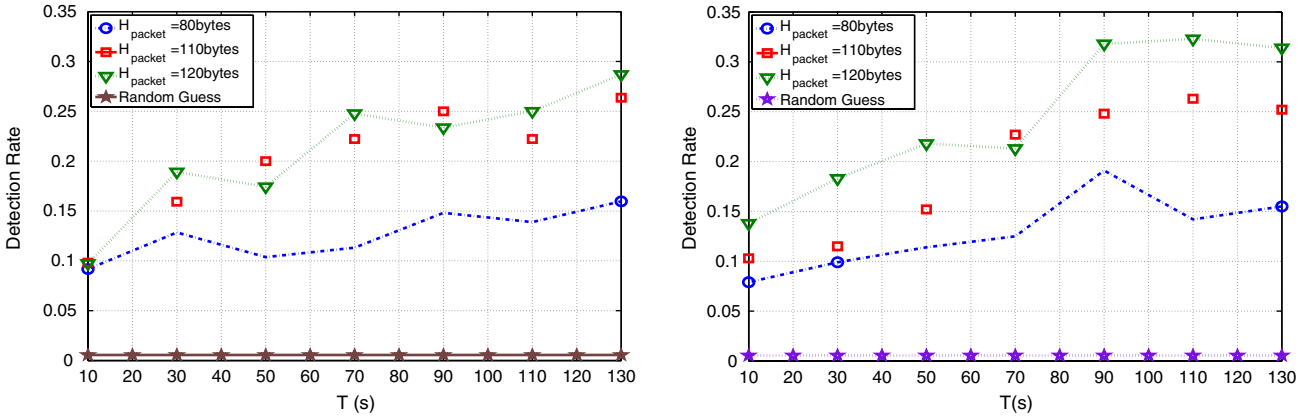
In all the experiments below, the training traces and candidate traces are all collected from *different* Skype calls. For better training, all the traces used in training are collected from sending end, i.e., from the link connected to Alice's computer.

5.3. Effect of parameter T (length of sample interval)

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

Fig. 6 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 pattern usually in the order of second as discussed in Section 4.1. (c) When T becomes large, the detection rate may drop simply because of shorter throughput vector used for training and detection. (d) The detection rate can be as high as 0.3, about 50-fold improvement over random guess. (e) The detection rate for candidate traces collected from sending end is comparable with the detection rate for candidate traces collected from receiving end. It is because T is big enough to filter out network dynamics

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



(a) Candidate Traces Collected from Sending End (b) Candidate Traces Collected from Receiving End

Fig. 6. Effect of parameter T on speech detection.

at receiving end which can vary from call to call. Similar observations can be made from Fig. 7. The detection rate for speaker detection can reach 0.18, about 30-fold improvement over random guess.

5.4. Effect of parameter H_{packet} (threshold on packet size)

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

Fig. 8 shows the effect of the parameter H_{packet} on speech detection. From Fig. 8, 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 are left because of the larger threshold. (c) The detection rate for speech detection can achieve 0.32, about 55-fold improvement over random guess.

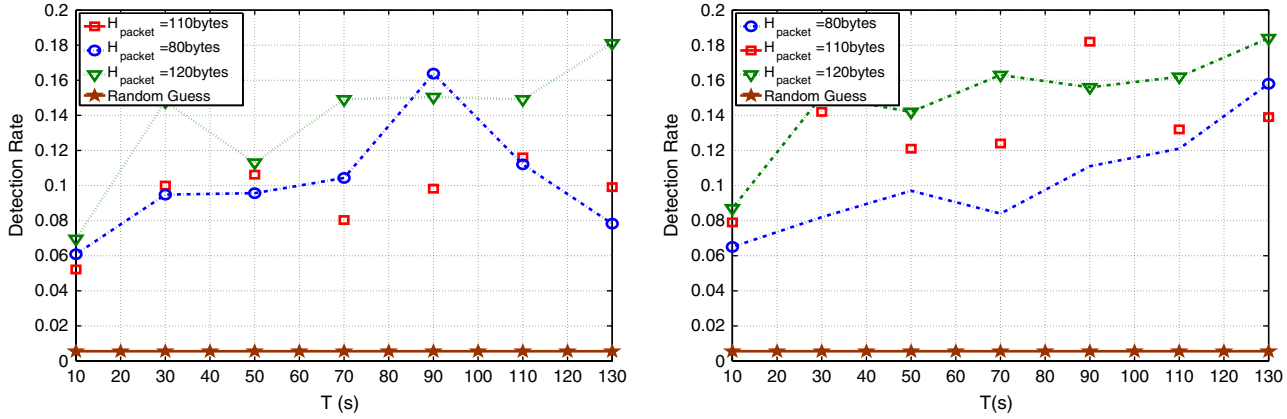
Fig. 9 shows the effect of the parameter H_{packet} on speaker detection. We can observe: (a) The best range of H_{packet} for speaker detection is from 110 bytes to 130 bytes. (b) The detection rate can reach to 0.2, about 34-fold improvement over random guess.

5.5. 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 lengths of training traces and test traces. Fig. 10 shows the experiment results on length of test traces. The results are obtained with training traces of length 38.5 min, $T = 110$ s, and $H_{packet} = 120$ bytes. We can observe that detection rates for both speaker detection and speech detection increase with length of test traces. When test traces are 25 min long, the detection rates for speaker detection and speech detection are 0.30 and 0.23, 50-fold and 39-fold improvement over random guess respectively. In this set of experiments, we also observe that speakers with similar talk patterns such as multiple similar segments of throughput vectors are misidentified as each other with higher probabilities than other speakers. When the length of training and test traces increases, the probability distribution of the misidentification is more close to the uniform distribution. We believe it is because (1) talk patterns from different speaker are different albeit of some similarities and (2) longer traces can better train the models to capture the difference.

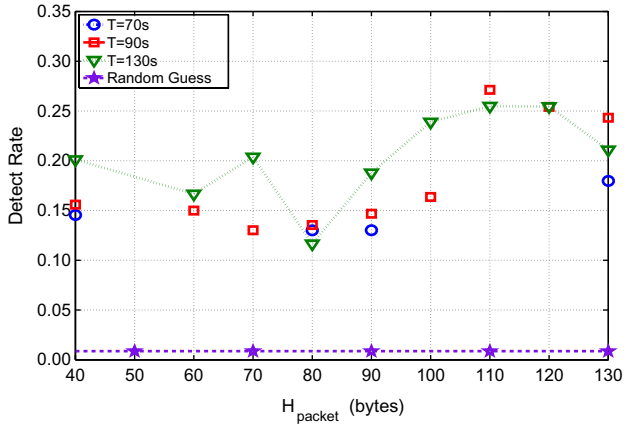
5.6. Pool size

In this set of experiments, we investigate the performance of traffic analysis attacks with different size of candidate pool. From the experiments results shown in Fig. 11, we can observe that when pool size increases, the detection rate slightly decreases for both speech detection and speaker detection, since it is harder to

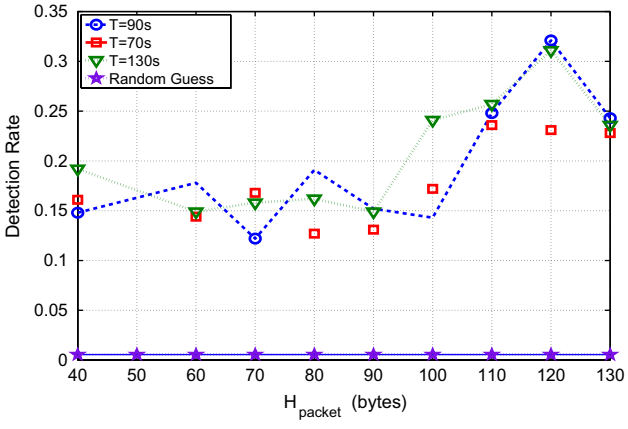


(a) Candidate Traces Collected from Sending End (b) Candidate Traces Collected from Receiving End

Fig. 7. Effect of parameter T on speaker detection.

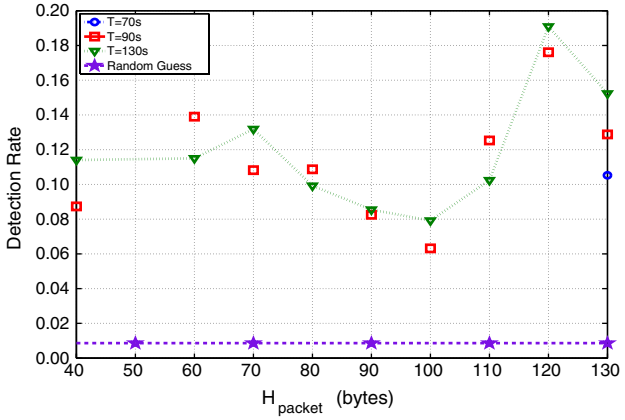


(a) Candidate Traces Collected from Sending End

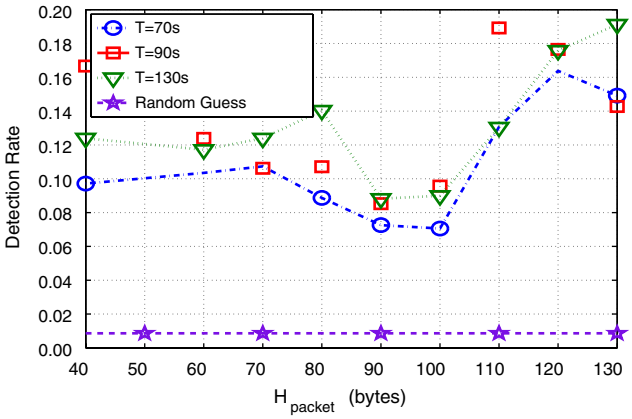


(b) Candidate Traces Collected from Receiving End

Fig. 8. Effect of parameter H_{packet} on speech detection.

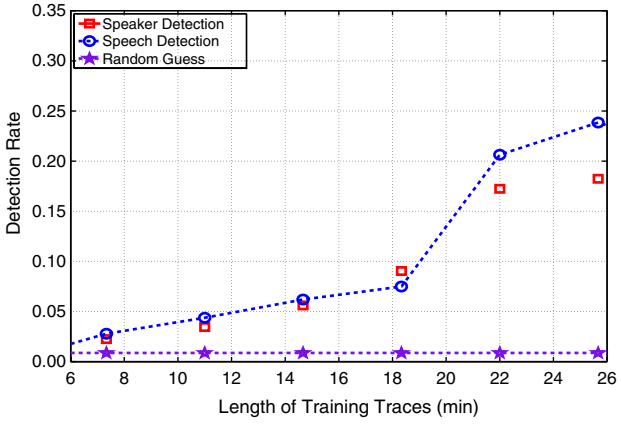


(a) Candidate Traces Collected from Sending End

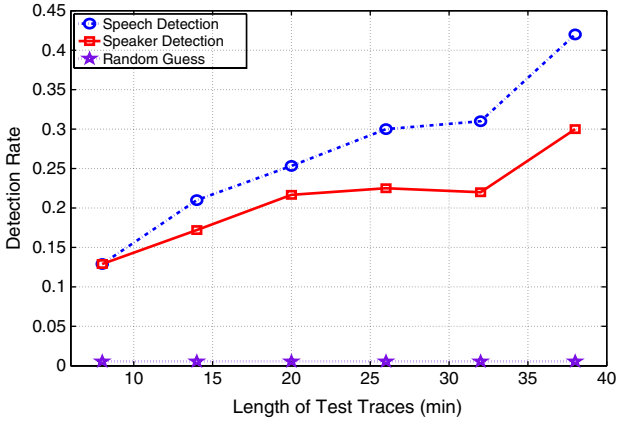


(b) Candidate Traces Collected from Receiving End

Fig. 9. Effect of parameter H_{packet} on speaker detection.



(a) Length of Training Traces



(b) Length of Test Traces

Fig. 10. Detection performance with different length of training traces and test traces.

find the right one from a larger candidate pool. But the ratio between the speech detection rate and random guess rate changes from 12.59 when pool size is 27–31.60 when pool size is 105, meaning the traffic analysis attacks are more effective when the pool size is large.

5.7. 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

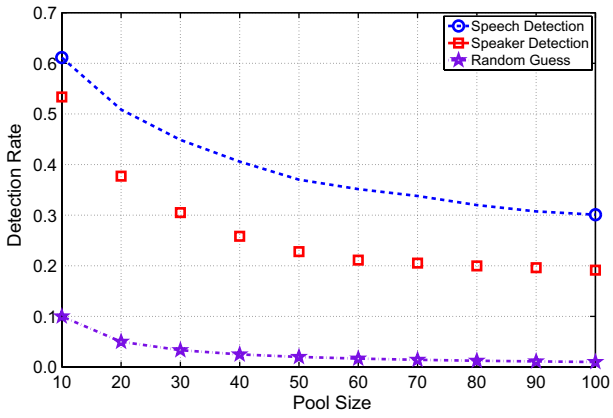


Fig. 11. Detection performance with different pool size.

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 3 trials.

From previous experiments, we learned 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.

Fig. 12 shows the performance of intersection attack. From Fig. 12, we can observe: (a) When K_{top}, the number of most likely candidates selected from each trial, increases, in general 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 sending end is again comparable with the detection rate for candidate traces collected from 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 rate can be achieved. Through cross-validation and averaging detection performance over various parameters, we believe the detection results are generally applicable.

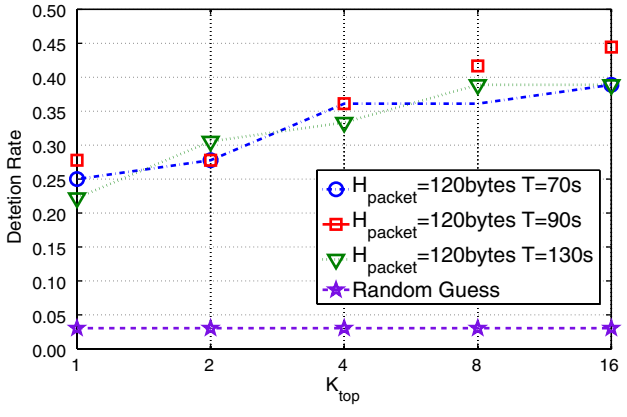
6. False alarm evaluation of speaker detection

In this section, we evaluate the proposed speaker detection with false alarm rates. For this purpose, we changed the threat model as follows: We assume that the adversary possesses traces of speech communications made by Alice and other speakers. We call these traces as labeled traces since these traces are collected in advance and the adversary knows the identities of speakers. The goal of the adversary is to detect, whether Alice is the speaker of a speech communication of interest. The major differences from the initial threat model are: (1) The initial threat model assumes that the adversary only possesses the Alice's traces in advance. In the new threat model, the adversary possesses both Alice's traces and other speakers' traces in advance. (2) In the initial threat model, the adversary aims to find Alice's trace from a pool of candidate traces based on Alice's traces collected in advance. The goal of the adversary in the new threat model is to detect whether a speech communication of interest is made by Alice.

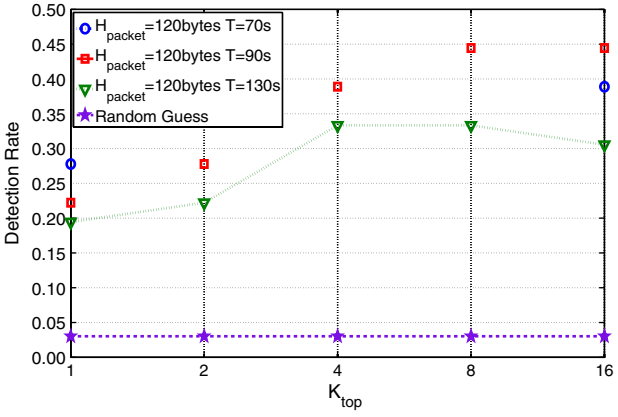
6.1. Detection approach

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

1. The adversary splits the labeled traces of Alice's speech communications into two halves. An HMM to model Alice's talk pattern is established based on the first half of the traces.
2. A detection threshold T_{tol} is determined based on remaining labeled traces including the second half of traces of Alice's speech communications. The adversary evaluates each of these traces against Alice's model and calculates its likelihood. Given a threshold T_{tol}, 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 speech communications detected as speech communications made by other speakers, i.e., the proportion of Alice's speech communications with likelihood values less than T_{tol}. (b) False positive rate is defined as the proportion of speech communications made by other speakers detected as Alice's speech communications, i.e., the proportion of other speakers' traces with likelihood values larger than T_{tol}. The threshold T_{tol} 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_{tol}, the given trace is declared as Alice's trace. Otherwise, the trace is declared as a trace made by other speakers.

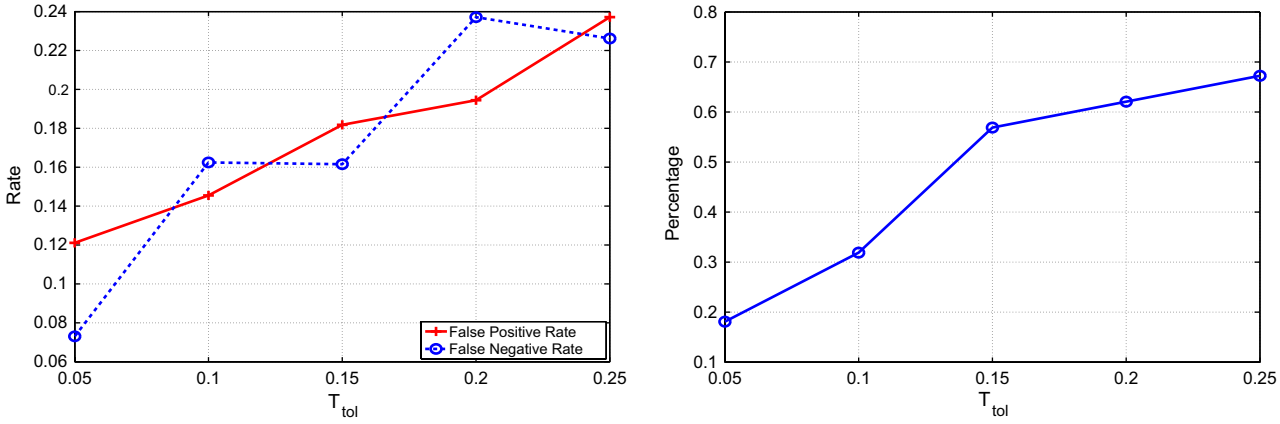


(a) Candidate Traces Collected from Sending End



(b) Candidate Traces Collected from Receiving End

Fig. 12. Performance of intersection attack.



(a) Detection Performance

(b) Percentage of Traces which can be Tested

Fig. 13. Performance of speaker detection without candidate pools.

6.2. Performance evaluation

We evaluate the detection performance with three metrics: False negative rate, false positive rate, and percentage of traces which can be tested. The two metrics, the false negative rate and the false positive rate used in performance evaluation, are calculated *on the test traces*. The last metric, percentage of traces which can be tested, is needed because for certain group of labeled traces, it is impossible to find a threshold T_{tol} 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 parameters are $T = 110$ s and $H_{packet} = 120$ - bytes and the average length of labeled traces and test traces are 39 min. The experiment results are averaged over all possible combinations of training traces and test traces.

Experiment results shown in Fig. 13 indicate that false positive rate and false negative rate both increase when the tolerance T_{tol} increases as expected 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 both false positive rate and false negative rate are below 0.2 when $T_{tol} = 0.15$ and around 60% traces can be tested.

From Fig. 13, we can observe that both false alarm rate and false negative rate are larger than 0.1 when $T_{total} \geqslant 0.1$. In other words, the proposed detection approach is not very effective in practice when $T_{total} \geqslant 0.1$. But still the detection results indicate the a serious vulnerability: The detection rates much higher than the random guess rate indicate that the detection approach can greatly reduce the anonymity of the Skype speech communications.

7. Possible countermeasures

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.

A naive countermeasure is to pad all the packets to the same size. We do not propose this countermeasure because: (a) A significant amount of bandwidth can be wasted to send padding bits. (b) Skype flows of constant packet sizes may catch special interest from adversaries.

In the rest of this section, we introduce a countermeasure which can protect privacy at the cost of marginal effect on quality of VoIP calls.

7.1. Skype camouflage

The main idea of the countermeasure is to camouflage Alice's Skype packets according to another speaker's traces. As shown in Fig. 14, 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: (1) 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. (2) If the buffer is empty, dummy packets will be sent to consume available byte tokens. During re-packetization, packet delimiters are added to the end of original packets and these re-packetized packets are encrypted with a session key shared between both parties of the Skype call.[4]

At the receiving end, the re-packetized packets are first decrypted with the session key and then converted to original Skype packets based on the packet delimiters. Recovered Skype packets are forwarded to the Skype client.

7.2. Performance evaluation of the countermeasure

We evaluate the countermeasure with two metrics: (a) The detection rate defined in Section 5-B: 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 5.1.

Fig. 15 shows the performance of the countermeasure. Fig. 15(a) shows that the countermeasure can preserve the privacy of Skype calls since the detection rate is around the probability of random guess. Fig. 15(b) shows the distribution of packet delay caused by the countermeasure. The mean of the delay caused by the countermeasure is 0.10 ms. The delay is less than 0.102 ms 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 [39].

In our experiments, we also find the delay caused by the countermeasure is smaller when Speaker X speaks more than Alice since more byte tokens are generated. So it is desired to

[4] The session key can be shared between both parties with Diffie–Hellman exchange as in Zphone [38].

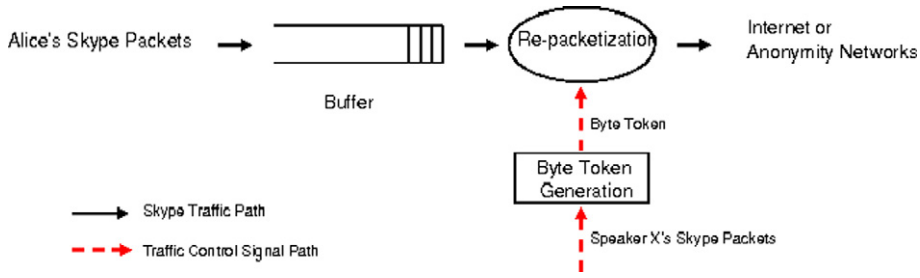
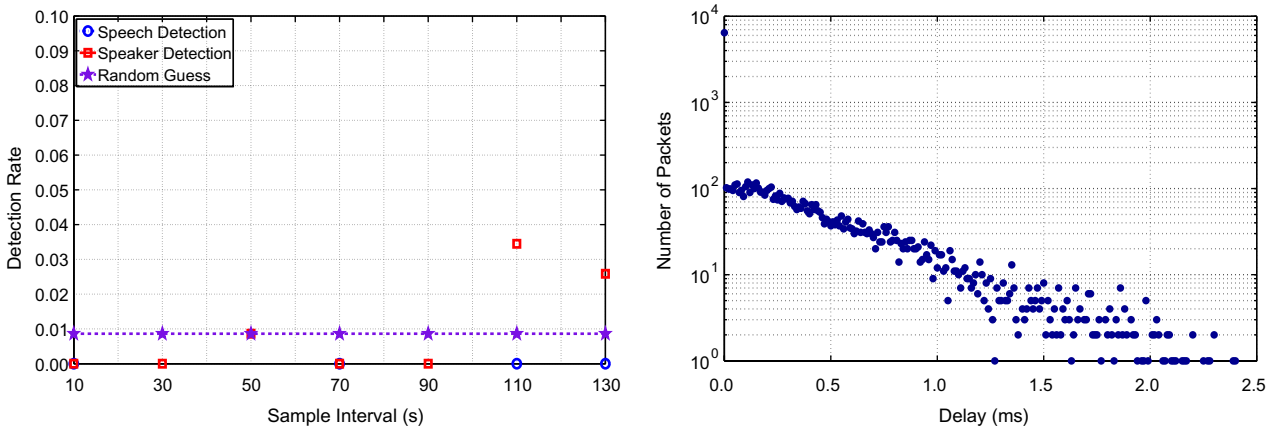


Fig. 14. Countermeasure: camouflaging Alice's Skype packets.



(a) Detection Rate

(b) Distribution of Packet Delay Caused by Countermeasure

Fig. 15. Performance of the proposed countermeasure.

camouflage Alice's Skype calls with a Skype trace from a speaker who speaks more than Alice in phone calls. Our future work will focus on modeling the delay caused by the difference between Alice's speech and Speaker X's speech and providing a guideline on choosing Skype traces of Speaker X.

8. 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. Aside from the intersection attack, a number of improvements could be made to for higher detection rates: (1) Adding more training traces or increasing the length of training traces will improve detection performance as shown in Fig. 10. Given the satisfactory results with only a small number of training traces, we leave the task of further improving performance of the traffic analysis attacks for future work. (2) We can also improve the detection performance by removing the noise in the talk pattern recovery. Since Skype uses proprietary protocols and strong encryption, it is not easy to differentiate speech packets from signaling packets. In turn, the recovered talk patterns are noisy. We plan to further analyze the Skype traffic and investigate approaches to separate out signaling packets to further improve detection performance.

The traditional speaker detection problem assuming access to speech signals has been well studied [40]. In [30], a speaker detection approach based on face, mouth motion, and silence detection is proposed. In comparison with the 90% high detection rate achieved in [30], our detection rate is relatively low simply because fewer features are available for traffic analysis and only noisy talk patterns recovered from packet sizes are available for traffic analysis. We plan to investigate the fundamental limits of the proposed attacks with only noisy talk patterns recovered from Skype traces in our future work.

The framework proposed in this paper, including extracting application-level features from network traffic traces and statistical analysis of extracted application-level feature by the HMM, can be potentially used for other applications. For example, it can be used to detect cheating with game bots in on line gaming since game bots and human players play games in different ways so that their gaming patterns at the application layer are different. One of our future tasks is to explore the potential of the framework.

9. Conclusion

In this paper, we propose a class of passive traffic analysis attacks to compromise privacy of Skype 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 greatly compromise the privacy of Skype calls with only a small number of training traces. We propose a countermeasure to mitigate the proposed traffic analysis attacks by camouflaging. The proposed countermeasure has negligible effect on quality of Skype calls.

References

[1] S.A. Baset, H.G. Schulzrinne, An analysis of the skype peer-to-peer internet telephony protocol, in: INFOCOM 2006. 25th IEEE International Conference on Computer Communications, Proceedings, 2006, pp. 1–11. Available from: <http://dx.doi.org/10.1109/INFOCOM.2006.312>.

[2] P2p telephony explained – for geeks only. Available from: <http://www.skype.com/help/guides/p2pexplained/>.

[3] T. Berson, Skype security evaluation, Tech. Rep. ALR-2005-031, Anagram Laboratories, 2005.
[4] K.-T. Chen, C.-Y. Huang, P. Huang, C.-L. Lei, Quantifying Skype user satisfaction, (2006) 399–410.
[5] M. Perényi, S. Molnár, Enhanced Skype traffic identification, in: ValueTools '07: Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2007, pp. 1–9.
[6] R. Dingledine, N. Mathewson, P. Syverson, Tor: the second-generation onion router, in: Proceedings of the 13th USENIX Security Symposium, San Diego, CA, 2004, pp. 303–320.
[7] O. Berthold, H. Federrath, S. Köpsell, Web MIXes: a system for anonymous and unobservable Internet access, in: H. Federrath (Ed.), Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, Springer-Verlag, LNCS 2009, 2000, pp. 115–129.
[8] D.L. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, Communications of the ACM 24 (2) (1981) 84–90. Available from: <http://doi.acm.org/10.1145/358549.358563>.
[9] A. Pfitzmann, B. Pfitzmann, M. Waidner, ISDN-mixes: untraceable communication with very small bandwidth overhead, in: Proceedings of the GI/ITG Conference on Communication in Distributed Systems, 1991, pp. 451–463.
[10] M. Rennhard, B. Plattner, Introducing morphmix: peer-to-peer based anonymous internet usage with collusion detection, in: WPES '02: Proceedings of the 2002 ACM Workshop on Privacy in the Electronic Society, ACM Press, New York, NY, USA, 2002, pp. 91–102. Available from: <http://doi.acm.org/10.1145/644527.644537>.
[11] K. Bennett, C. Grothoff, GAP – practical anonymous networking, in: R. Dingledine (Ed.), Proceedings of Privacy Enhancing Technologies Workshop (PET 2003), Springer-Verlag, LNCS 2760, 2003.
[12] M.J. Freedman, R. Morris, Tarzan: A peer-to-peer anonymizing network layer, in: Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, 2002, pp. 193–206. Available from: <http://doi.acm.org/10.1145/586110.586137>.
[13] D.M. Goldschlag, M.G. Reed, P.F. Syverson, Hiding routing information, Information Hiding (1996) 137–150.
[14] B.N. Levine, M.K. Reiter, C. Wang, M.K. Wright, Timing attacks in low-latency mix-based systems, in: Proceedings of Financial Cryptography (FC '04), Key West, FL, 2004, pp. 251–265.
[15] S.J. Murdoch, G. Danezis, Low-cost traffic analysis of Tor, in: Proceedings of the 2005 IEEE Symposium on Security and Privacy, IEEE CS, 2005.
[16] Y. Zhu, X. Fu, B. Graham, R. Bettati, W. Zhao, Correlation-based traffic analysis attacks on anonymity networks, IEEE Transactions on Parallel and Distributed Systems 99 (PrePrints). Available from: <http://doi.ieeecomputersociety.org/10.1109/TPDS.2009.146>.
[17] Y. Zhu, R. Bettati, Compromising anonymous communication systems using blind source separation, ACM Transactions on Information and System Security 13 (1) (2009) 1–31. Available from: <http://doi.acm.org/10.1145/1609956.1609964>.
[18] D.X. Song, D. Wagner, X. Tian, Timing analysis of keystrokes and timing attacks on ssh, in: SSYM'01: Proceedings of the 10th Conference on USENIX Security Symposium, USENIX Association, Berkeley, CA, USA, 2001, p. 25.
[19] Q. Sun, D.R. Simon, Y.-M. Wang, W. Russell, V.N. Padmanabhan, L. Qiu, Statistical identification of encrypted web browsing traffic, in: IEEE Symposium on Security and Privacy, Society Press, 2002.
[20] T.S. Saponas, J. Lester, C. Hartung, S. Agarwal, T. Kohno, 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, Berkeley, CA, USA, 2007, pp. 1–16.
[21] C.V. Wright, L. Ballard, S.E. Coull, F. Monrose, G.M. Masson, Spot me if you can: uncovering spoken phrases in encrypted voip conversations, in: SP '08: Proceedings of the 2008 IEEE Symposium on Security and Privacy, IEEE Computer Society, Washington, DC, USA, 2008, pp. 35–49. Available from: <http://dx.doi.org/10.1109/SP.2008.21>.
[22] C.V. Wright, L. Ballard, F. Monrose, G.M. Masson, Language identification of encrypted voip traffic: Alejandray roberto or alice and bob? in: SS'07: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, USENIX Association, Berkeley, CA, USA, 2007, pp. 1–12.
[23] Speex:a free codec for free speech. Available from: <http://www.speex.org/>, <http://www.speex.org/>.
[24] X. Wang, S. Chen, S. Jajodia, 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, Washington, DC. Available from: <http://dx.doi.org/10.1109/SP.2007.30>.
[25] W. Jiang, H. Schulzrinne, Analysis of on-off patterns in voip and their effect on voice traffic aggregation, in: Computer Communications and Networks, 2000, Proceedings. Ninth International Conference on, 2000, pp. 82–87. Available from: <http://dx.doi.org/10.1109/ICCCN.2000.885474>.
[26] C. Rathinavelu, L. Deng, 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, IEEE Computer Society, Washington, DC, USA, 1996, pp. 9–12. Available from: <http://dx.doi.org/10.1109/ICASSP.1996.540277>.
[27] M.-P. Schambach, 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, Washington, DC, USA, 2003, p. 119.
[28] An HMM-based approach for gesture segmentation and recognition, in: ICPR '00: Proceedings of the International Conference on Pattern Recognition, IEEE Computer Society, Washington, DC, USA, 2000, p. 3683.
[29] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, in: A. Waibel, K.-F. Lee (Eds.), Readings in Speech Recognition, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990, pp. 267–296.
[30] V. Pavlovic, J.M. Rehg, A. Garg, T.S. Huang, Multimodal speaker detection using error feedback dynamic bayesian networks, Computer Vision and Pattern Recognition, IEEE Computer Society Conference, vol. 2, 2000, p. 2034. Available from: <http://doi.ieeecomputersociety.org/10.1109/CVPR.2000.854730>.
[31] F. Jelinek, Continuous speech recognition by statistical methods, Proceedings of the IEEE 64 (4) (1976) 532–556.
[32] R. Bakis, Continuous speech recognition via centisecond acoustic states, The Journal of the Acoustical Society of America 59 (S1) (1976) S97. Available from: <http://dx.doi.org/10.1121/1.2003011> <http://link.aip.org/link/?JAS/59/S97/2>.
[33] O. Berthold, A. Pfitzmann, R. Standtke, The disadvantages of free MIX routes and how to overcome them, in: Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, 2000, pp. 30–45.
[34] G. Danezis, A. Serjantov, Statistical disclosure or intersection attacks on anonymity systems, in: Proceedings of 6th Information Hiding Workshop (IH 2004), Toronto, Canada, 2004, pp. 293–308.
[35] O. Berthold, H. Langos, Dummy traffic against long term intersection attacks, in: Proceedings of Privacy Enhancing Technologies Workshop (PET 2002), San Francisco, CA, 2002, pp. 110–128.
[36] FindnotProxyList. Available from: <http://www.findnot.com>.
[37] ResearchChannels. Available from: <www.researchchannel.org>.
[38] Zfone project home page. Available from: <http://zfoneproject.com/>.
[39] T. Szigeti, C. Hattingh, End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs (Networking Technology), Cisco Press, 2004.
[40] J.P. Campbell, Speaker recognition: a tutorial, Proceedings of the IEEE 85 (9) (1997) 1437–1462. Available from: http://dx.doi.org/10.1109/5.628714 http://dx.doi.org/10.1109/5.628714.