2016

Hiding Traffic Patterns in VoIP Communication

Jialue Fang
Iowa State University

Hiding traffic patterns in VoIP communication

by

Jialue Fang

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering

Program of Study Committee:

Yong Guan, Major Professor

Neil Zhenqiang Gong

Jennifer Lee Newman

Iowa State University

Ames, Iowa

2016

DEDICATION

I would like to dedicate this thesis to my father Ligang and to my mother Suixin without whose support I would not have been able to complete this work. I would also like to thank my friends and family for their loving guidance and financial assistance during the writing of this work.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr.Guan Yong for his guidance, patience and support throughout this research and the writing of this thesis. His insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my committee members for their efforts and contributions to this work: Dr.Newman Jennifer Dr.Gong Zhenqiang. I would additionally like to thank Dr. Zhu Ye for his guidance throughout the whole stages of my research.

ABSTRACT

Voice over IP(VoIP) is widely used in today's communication, VoIP is a methodology that able to converts analog voice signals into digital data packets and support real-time, two-way transmission of conversations using Internet Protocol. Despite of the fact that VoIP technology have greatly developed since the earliest design, it still suffer from the common problem that affect Internet security: hacker. Currently Timing-based attack is the most famous attack method on VoIP. Timing-based traffic analysis attacks mainly based on packet inter-arrival time. Attackers are able to analyze the packet sending time intervals and export user's talking pattern. Finally, attacker can identify the user by comparing the exported talking pattern with the talking pattern in their databases. Therefore, to protect user's identity, we propose a new application to hide user's talking pattern.

In this thesis, we address issues related to traffic analysis attacks and the corresponding countermeasures in VoIP traffic. We focus on a particular class of traffic analysis attack, timing-based correlation attacks, by which an adversary attempt to analyze packet inter-arrival time of a user and correlate the output traffic with the traffic in their database. Correlation method that is used in this type of attack, namely Dynamic Time Warping(DTW) based Correlation. Based on our threat model and known strategies in existing VoIP communication, we develop methods that can effectively counter the timing-based correlation attacks. The empirical results shows the effectiveness of the proposed scheme in term of countering timing-based correlation attacks.

Our experimental result showed that our application is able to hide user's identity in VoIP communication, with a few modifications in the sending process.

CHAPTER 1. INTRODUCTION

1.1 Research Motivation

Voice over IP (VoIP) communications are continuing gaining popularity due to their cost savings and rich features. By using this type of technology, users are able to use the telepone calls over the Internet and do not need to pay for any extra cost except for the Internet access fees. Because of the popularity of VoIP, a increasing number of Internet hackers started to focus on attacking VoIP users. In the past, the most famous type of attack is based on packet size. In this type of attack, attackers are able to analyze packets information and grab the information they want. In order to prevent this type of attack, numerous efforts such as SRTP [2] and ZRTP used in Zfone [18] have been put into securing VoIP communications. However VoIP communications are still vulnerable to traffic analysis attacks based on VoIP traffic patterns. Through the traffic analysis attacks, attackers can identify speeches [14], identify languages used into the VoIP communications [15], and identify speakers [16]. Thus, VoIP traffic patterns based attacks are aim to identify user's identity: their language, their topic, etc. Currently, the most common way to hide user's identity in the Internet is using anonymous communication softwares, like Tor [11], however, these softwares still potentially suffer from VoIP traffic patterns based attacks since it is designed for hiding traffic information rather than VoIP traffic pattern.

This project studies user identification attacks and the corresponding countermeasures in VoIP traffic. With rapid growth of the Internet as a tool of communication and information sharing, VoIP technology has been widely applied in Internet communication application software, such as Skype, X-Lite, Google Hangouts, etc. However, these VoIP Applications can potentially be attacked by many methods. In the past, the most common way to attack VoIP

network traffic is based on packet size, which are now perfectly modified by constant bit rate codecs, which generates the same packet size. Thus, attackers needs to find another way to hack the network communication, so packet inter-arrival time becomes the new target for attackers.

VoIP traffic pattern based attacks differ from traffic information based attacks by their attack target. Traffic information attacks are focus on grab packets information send by their target user and traffic pattern based attacks are focus grab packets sending time send by their target user. Recent research has proved that packet information can be successfully hide by Constant Byte Rate which generate same packet size, thus, it is difficult for attacker to analyze packet by their size. However, there is no securing action to hide traffic pattern, which defined as a series of talk spurts[1] and silence gaps[2]. From Dr. Zhu Ye's Paper "On Privacy of Encrypted Speech Communications" [17], attackers are able to "detect speakers of encrypted speech communications with high accuracy based on traces of 15 minutes long on average." So it is significant for us to find a way to fix this defect before this technology can move forward.

In this thesis, we propose a pattern hiding approach to mitigate traffic analysis attacks on VoIP communications. The approach hides traffic patterns by adding dummy packets, dropping VoIP packets, and delaying VoIP packets. The approach optimizes pattern hiding in terms of dissimilarity from the original traffic pattern and the optimization is under constraints on dummy packet rate, VoIP packet drop rate, and VoIP packet delay.

We formally modeled the behavior of an adversary who launches traffic analysis attacks. In order to successfully identify the user who is sending packets through the VoIP Application, the correlation techniques must accurately measure the similarity of user's output traffic and adversary's sample traffic. Correlation method that is used in this type of attack, namely DTW based Correlation. DTW based Correlation is used to measure the similarity of two traffic with different length. Moreover, we developed a pattern hiding module and measure the effectiveness in countering traffic analysis attacks.

[1]Talk spurt is a continuous segment of speech between two silence intervals
[2]Silence gap is the time intervals between two talk spurts

1.2 Thesis Organization

The rest of this thesis organized as follow: Chapter 2 covers the literature survey on existing researches on VoIP, anonymous network and pattern hiding. Chapter 3 defines the formal problem statement. Chapter 4 introduces the design of our pattern hiding module and its detailed implementation. Chapter 5 set up a series experiment based on our pattern hiding module and analyzes the experiment result. Chapter 6 we makes conclusion based on experiment result and discuss the limitation. Chapter 7 summarizes the thesis and future works.

CHAPTER 2. REVIEW OF LITERATURE

Internet communication security become increasingly important with the popularity of VoIP software. A lot of effort had been put on this area: anonymous communication, voice traffic camouflage, etc. Our goal is to design a pattern hiding module that can help increase the security of VoIP communication. So this chapter, we review previous work that is related to VoIP and its security technologies.

Anonymous communication has been proved very useful for hiding user's identification from outside observer. The most famous anonymous application on web browser Tor [5] can provide the user relatively safe web browsing by distributing user's transactions over several places on the Internet. But we note that Tor does not directly provide anonymity service for a VoIP communication, thus, attacker still have a greater chance to identify users.

2.1 Potential Attacks on VoIP

Skype, as one of the most popular VoIP service provider is able to protect users' privacy by using some unique features, such as: strong encryption, proprietary protocols, unknown codecs, dynamic path selection, and the constant packet rate. However, a number of researchers have shown that there still possible for attackers to compromise users' privacy according to a new traffic analysis attacks which is based on application-level features extracted from VoIP call traces[16]. Some recent research shows that when the audio is encoded using variable bit rate codecs, the length of encrypted VoIP packets can be used to identify the phrase spoken within a call and the language of the conversation.[14] [15]

In Zhu Ye's paper: "Traffic Analysis Attacks on Skype VoIP Calls" [16], author proposed a class of traffic analysis attacks that can extract feature of VoIP call traces. In this type of attack,

adversary will first collect Victim Alice's VoIP call traces. Then the adversary can extracts application-level features of Alice's VoIP calls and trains a Hidden Markov Model(HMM) with these extracted features. Finally, the adversary is able to calculate likelihood of the call being made by Alice.

In Fabian Monrose's paper: "Spot Me if You Can: Uncovering Spoken Phrases in Encrypted VoIP Conversations"[14], author proposed a technique that use the lengths of encrypted VoIP packets to identify the phrases spoken with a call. In this technique, even if the audio is encoded using variable bit rate(VBR), the average identification accuracy can reach 50% and 90% for some phrases.

In Fabian Monrose's paper: "Language Identification of Encrypted VoIP Traffic: Alejandra Y Roberto or Alice and Bob?"[15], author proposed a techniques that used the lengths of encrypted VoIP packets to identify the conversation language in VoIP communication. The research experiment result with 2066 native speacker of 21 different languages shows that encrypted VoIP communication traffic can be identify with very high accuracy.

2.2 Pattern Hiding Techniques

Some of the countermeasure methods have been developed for hiding network traffic. For example, NetCamo [8] is able to camouflage network traffic.par In [5], Tor proved to be a useful for web browsing anonymous, but it is not able to effectively hide voice traffic. In paper [16][14][15], the length of encrypted VoIP packets are being used to identify users and languages. NetCamo [8] provide a useful way to camouflage the traffic to avoid these identifications. In our paper, we focus pattern traffic hiding in VoIP communications without compromising the real-time requirement.

In speech communications, an analog voice signal is first converted into a voice data stream by a chosen codec. Typically in this step, compression is used to reduce the data rate. The voice data stream is then packetized in small units of typically tens of milliseconds of voice, and encapsulated in a packet stream over the Internet.

Silence suppression, also called voice activity detection (VAD), is designed to further save bandwidth. The main idea of the silence suppression technique is to disable voice packet

transmissions when silence is detected. To prevent the receiving end of a speech communication from suspecting that the speech communication stops suddenly, comfort noise is generated at the receiving end. Silence suppression is a general feature supported in codecs, speech communication software, and protocols such as RTP.

A silence detector makes voice-activity decisions based on the voice frame energy, equivalent to average voice sample energy of a voice packet. If the frame energy is below a threshold, the voice detector declares silence.

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.

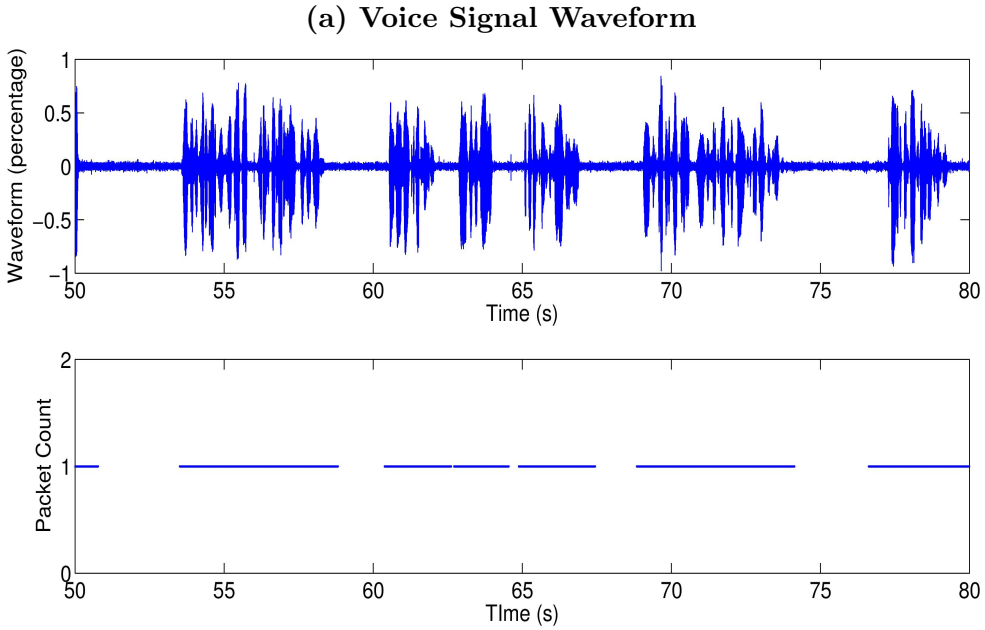


Figure 2.1: An Example of Silence Suppression

Figure 2.1 shows an example of the silence suppression. Figure 2.1.(a) shows the waveform of a sheriff's voice signal extracted from a video published at cnn.com [4]. Figure 2.1.(b)

shows the packet train generated by feeding the voice signal to X-Lite [1], a popular speech communication tool. From Figure 2.1, we can easily observe the correspondence between the silence periods in the voice signal and the gaps in the packet train. The length of a silence period is slightly different from the length of the corresponding gap in the packet train because of the hangover technique. The on-off pattern shown in Figure 2.1.(b) can leak sensitive information.

CHAPTER 3. PROBLEM DEFINITION

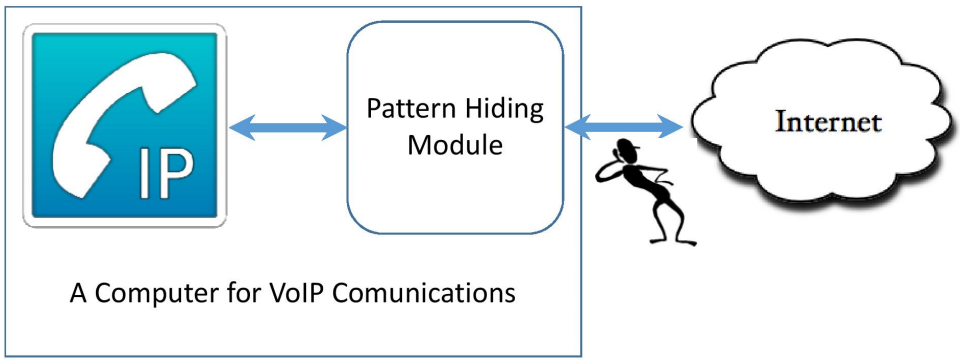


Figure 3.1: Pattern Hiding Module

The formal problem statement can be formulated as follows: design a module to hide the on-off traffic pattern shown in Figure 2.1. As shown in Figure 3.1, the pattern hiding module is installed on the same computer running VoIP software. The module intercepts VoIP packets generated by the VoIP software, add timing perturbation to hide traffic pattern, and then send perturbed traffic to the Internet.

From previous researches, a lot of effort had been put on securing speech communication, so we assume the VoIP traffic is encrypted with one of the secure versions of the RTP protocol such as SRTP [2] or ZRTP used in Zfone [18] to protect confidentiality of speech communications.

We also assume VoIP packets are of the same size because of the following reasons:

1. Most codecs used in current speech communications are CBR codecs[1].

2. During encryption, speech packets can be padded to a fixed length.

[1]Variable bit rate (VBR) codecs are primarily used for coding audio files instead of real-time speech communications [13, 3].

We assume attackers uses following speaker detection methods: detect speaker with a specific encrypted speech communication, such as the online course instructor, e-conference meeting speaker. In this project, we assume the interest speaker is Alice. Before apply speaker detection on Alice, attacker, we call it Eve, will first collect encrypted speech communications data send by Alice in advance so that Eve can compare the data he got with these encrypted speech communication data and see if they are match.

In order to define adversary's power, we also make following assumptions:

1. We assume an adversary is able to eavesdrop VoIP traffic to and from the computer running VoIP software.

2. Since VoIP packets are encrypted and of the same length, the adversary attempts to disclose sensitive information through timing of VoIP packets.

To sum up, In this project, we assume that the adversary uses a classical timing analysis attack, which summarized as follow:

1. The adversary observe user?s output network traffic, collects the inter-arrival times of the packet and generate user?s talk spurts and silence gap with optimal threshold.

2. To maximize adversary?s power, we assume that he can catch all the traffic from his observed user.

3. The optimization model?s techniques and strategies are known to the adversary. This is a typical assumption in the study of security systems. Above two assumptions create worst case scenario in terms of security analysis.

4. The adversary cannot correlate input talk spurts and silence gaps to output talk spurts and silence gaps. Content and packet size correlation is prevented by encryption and packet timing based correlation is prevented by batching.

5. Finally we assume that the specific objective of the adversary is to identify the user of output traffic.

CHAPTER 4. METHODS AND PROCEDURES

4.1 Overview

The pattern hiding module is designed to hide the on-off pattern in VoIP traffic. We quantify the hiding performance as the correlation between the on-off pattern in the original traffic and the on-off pattern in the perturbed traffic. We denote the length of the ith talk spurt and the ith silence gap in the original traffic as x_i^t and x_i^s respectively. Similarly the ith talk spurt and the ith silence gap in the perturbed traffic can be denoted as y_i^t and y_i^s respectively. So the on-off patterns in the original traffic and the perturbed traffic can be denoted as $X = [x_1^t, x_1^s, x_2^t, x_2^s, \cdots, x_i^t, x_i^s, \cdots, x_n^t, x_n^s]$ and $Y = [y_1^t, y_1^s, y_2^t, y_2^s, \cdots, y_i^t, y_i^s, \cdots, y_n^t, y_n^s]$ where n is the number of talk spurts and silence gaps. The correlation between the on-off patterns can be written as:

$$D(X,Y) = \frac{\sum_{i=1}^{n}(x_i^t - \bar{x})(y_i^t - \bar{y}) + \sum_{i=1}^{n}(x_i^s - \bar{x})(y_i^s - \bar{y})}{\sqrt{\sum_{i=1}^{n}[(x_i^t - \bar{x})^2 + (x_i^s - \bar{x})^2]\sum_{i=1}^{n}[(y_i^t - \bar{y})^2 + (y_i^s - \bar{y})^2]}} \tag{4.1}$$

where $\bar{x} = \frac{\sum_{i=1}^{n}(x_i^t + x_i^s)}{2n}$ and $\bar{y} = \frac{\sum_{i=1}^{n}(y_i^t + y_i^s)}{2n}$.

The goal of the module is to minimize the correlation defined in Equation 4.1. The time perturbation to the traffic can be adding dummy packets, dropping VoIP packets, and delaying VoIP packets. Any of the timing perturbation techniques incur costs:

1. Adding dummy packets can increase bandwidth usage.

2. Dropping VoIP packets can degrade QoS of VoIP communications. QoS of VoIP communications is acceptable if the packet drop rate is less than 5%.

3. Delaying VoIP packets can increase the overall delay of VoIP packets and cause QoS degradation of VoIP communications.

Add Dummy Packets: When we add a packet, we will insert a dummy packet between two VoIP packets, so that these packets can either generate two silence gaps instead of one. (Because insert a packet in talk spurt will not change the pattern, here we assume all the new packets are inserted during silence gap) or cover the silence gap. As Figure 4.1 shows, we have two packets (Original Packet 1 and Original Packet 2) to be send and there is a gap between these two packets. Now, we insets a new packet (Dummy Packet) between Original Packet 1 and Original Packet 2, thus, two new gap has been created or the original gap has been covered. Either way, user?s talk pattern has been changed.

Drop VoIP Packets:When we drop a VoIP packet, we drop a packet to create ether a new silence gap or a silence gap that is larger than the previous one. As Figure 4.2 shows, the original data contains 3 packets (Original Packet 1, Original Packet 2 and Original Packet 3) and 2 gaps between these 3 packets. New we dropped Packet 3, which also means we combine two gaps into ones. If Original Packet 1, Original Packet 2 and Original Packet 3 are send out during a talk spurt, this action will generate a new silence gap; if the original gaps already are silence gaps, this action will combine this two silence gaps into a large silence gap. Either way, this action will change the user?s talking pattern.

Delay VoIP Packets:When we delay a packet, we will hold the packet for certain period of time before send it out, so that we can create a new silence gap or enlarge the original silence gap. In Figure 4.3, it shows the original VoIP packet data: 2 packets and 1 gap. In the lower half of the figure, it shows that Packet 2 has been moved to a further location in the timeline, which enlarge the gap between Original Packet 1 and Original Packet 2. In this situation, it will either generate a new silence gap or enlarge the original silence gap. Either way, this action will change the user's talking pattern.

So the module can be essentially formulated as an optimization problem: The goal is to minimize the objective function defined in Equation 4.1. The constraints of the optimization problem are the limit on the adding rate of dummy traffic (denoted as lim_{add}), the limit on the dropping rate of VoIP packets (denoted as lim_{drop}), and the limit on the delay to VoIP packets (denoted as lim_{delay}).

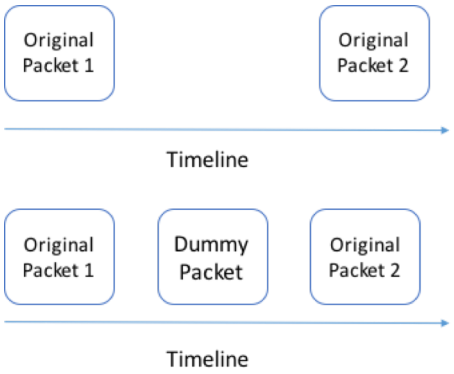


Figure 4.1: Model for Add Dummy Packets

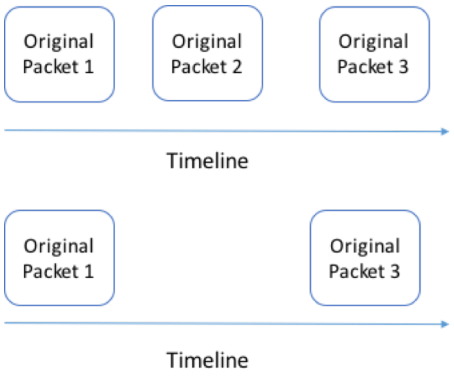


Figure 4.2: Model for Drop VoIP Packets

Recall that our applications objective is to minimize the correlation between input talk spurts/silence gaps and output talk spurts/silence gaps. This could be achieved by adding dummy packets into the normal traffic, dropping actual packets from the normal traffic and delay original packets in the normal traffic. Using these methods, we are able to generate a modified output inter-arrival time, which are different from input inter arrival time. To find the minimum correlation between input and output inter-arrival time, we have two options, first, we can use mathematic correlation formula to make decision on add,drop and delay packet, in our research, we used Pearson?s correlation coefficient formula. Another method is using dynamic time warping algorithm to find the optimal match between two given sequences with certain restriction.

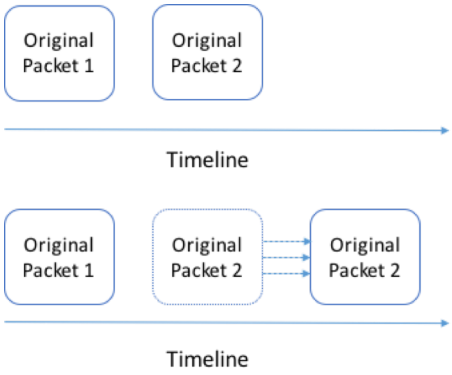


Figure 4.3: Model for Delay VoIP Packets

The optimization has to run as an online algorithm as the input to the optimization such as the on-off pattern in the original traffic is not known in advance. The online optimization starts with replicating the first $n-1$ talk spurts and silence gaps from the input of the module, i.e., the original traffic, to the output of the module, i.e., the perturbed traffic. Given the first $n-1$ talk spurts and silence gaps in both the input and the output of the module, the optimization algorithm computes the optimal length of the nth talk spurt in the output. From then on, the optimization computes the optimal length of the next talk spurt or silence gap in the output based on the previous $n-1$ talk spurts and silence gaps in the input and the output of the module.

Since the optimization has to run as an online algorithm, the packet delay caused by the optimization needs to be taken into account. For example, to compute the optimal length of the ith talk spurt in the output traffic, the optimization algorithm needs to know the length of the corresponding talk spurt in the input traffic. The optimization will not know the end of the talk spurt until one packetization delay after the arrival of the last packet of the talk spurt, which is approximately 20ms or 30ms for most codecs. Since the optimization also needs computation time, the last packet of the talk spurt needs to be delayed at least for one packetization delay and the computation delay of the optimization before a decision can be made for the packet. The excessive delay is not acceptable for VoIP communications.

To avoid the excessive delay, our optimization algorithm does not compute based on the actual length of the current talk spurt or silence gap. Instead, the algorithm computes based on the predicted length of the next talk spurt or silence gap.

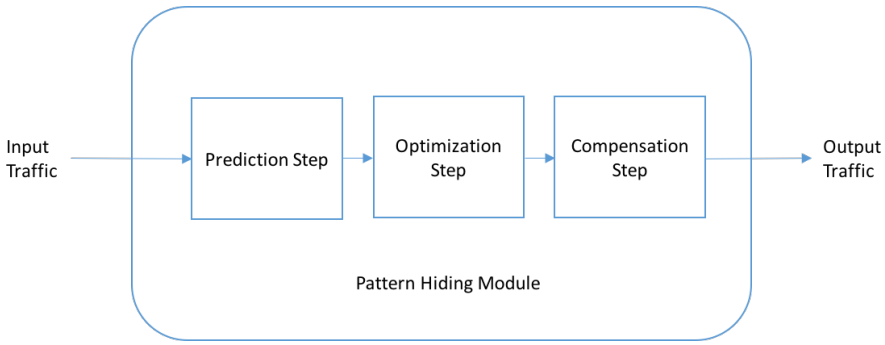


Figure 4.4: Pattern Hiding Module

As shown in 4.4, the pattern hiding module has following three steps:

- The prediction step predicts the length of the next talk spurt or silence gap based on the history of the on-off patterns.
- The optimization step calculates the optimal length of the next talk spurt or silence gap in the output traffic based on the predicted length of the next talk spurt or silence gap.
- The compensation step computes compensation needed to achieve the optimal pattern hiding because of prediction error. Randomization is also included in the compensation step to randomize output traffic and the randomization can make output traffic traces generated from the same input traffic different from each other.

We describe the details of each step in the rest of this section.

4.2 Prediction Step

In this paper, we use a neural network to predict the length of the next talk spurt or silence gap. Neural networks have been successfully applied to predict time series data such as stock index [12] and solar activity [6]. The neural network used in this paper is the nonlinear autoregressive network with exogenous inputs (NARX) model [9]. As shown in Figure 4.5, the

NARX model used in this paper is a two-layer feedforward network with one hidden layer and one output layer. In Figure 4.5, the prediction is on silence gaps and the past talk spurts are used as the external input. When predicting length of talk spurts, past silence gaps are used as the external input.

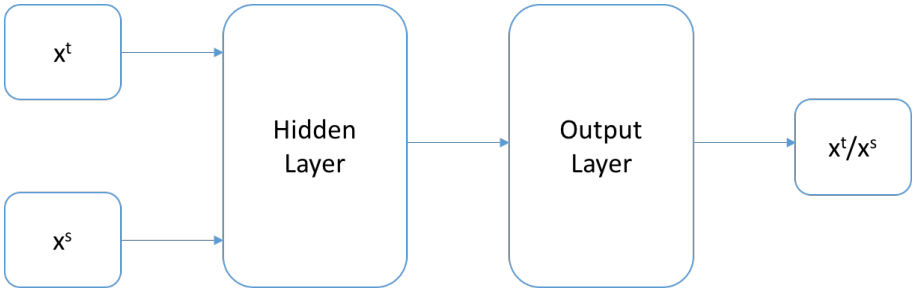


Figure 4.5: NARX Model Used to Predict Length of Silence Gaps (X^t: a vector of talk spurts, X^s : $a vector of silence gaps$.)

The input of this step is a neural network model trained with previous VoIP communication traces. If we assume the index of the next talk spurt or silence gap is j, this step outputs $x_j^{p,t}$, the predicted length of the next talk spurt, or $x_j^{p,s}$ the predicted length of the next silence gap.

4.3 Optimization Step

Given the predicted length of the next talk spurt or silence gap in the input traffic from the previous step, the optimization step outputs the optimal length of the next talk spurt or silence gap. Without loss of generality, we assume the input and output of this step are $x_j^{p,s}$, the predicted length of the next silence gap in the input traffic, and $y_j^{o,s}$, the optimal length of the output traffic respectively. The objective function is as shown in (4.2).

In (4.2), $\bar{x} = \frac{\sum_{i=j-n+1}^{j} x_i^t + \sum_{i=j-n+1}^{j-1} x_i^s + x_j^{p,s}}{2n}$ and $\bar{y} = \frac{\sum_{i=j-n+1}^{j} y_i^t + \sum_{i=j-n+1}^{j-1} y_i^s + y_j^{o,s}}{2n}$.

In the objective function (4.2), the only variable is $y_j^{o,s}$. Since the optimization is online, all the lengths of the previous talk spurts and silence gaps are known.

The single-variable optimization problem can be solved with the classical approach based on the derivative test. The solution of the optimization problem can be found in Appendix .

$$D(y_j^{o,s}) = \frac{\sum_{i=j-n+1}^{j}(x_i^t-\bar{x})(y_i^t-\bar{y}) + \sum_{i=j-n+1}^{j-1}(x_i^s-\bar{x})(y_i^s-\bar{y}) + (x_j^{p,s}-\bar{x})(y_j^{o,s}-\bar{y})}{\sqrt{[\sum_{i=j-n+1}^{j}(x_i^t-\bar{x})^2 + \sum_{i=j-n+1}^{j-1}(x_i^s-\bar{x})^2 + (x_j^{p,s}-\bar{x})^2][\sum_{i=j-n+1}^{j}(y_i^t-\bar{y})^2 + \sum_{i=j-n+1}^{j-1}(y_i^s-\bar{y})^2 + (y_j^{o,s}-\bar{y})^2]}} \quad (4.2)$$

To avoid repetition, we focus on the optimizing the length of the next silence gap only in this subsection. The length of the next talk spurt can be optimized in the same way.

4.4 Compensation Step

The compensation step is designed for two purposes:

1. The optimization step is based on the predicted length of the next talk spurt or silence gap and any prediction error can lead to performance degradation in pattern hiding. This step is designed to compensate the degradation in hiding performance due to the prediction error.

2. This step is also designed to add randomization in pattern hiding and the randomization makes two traces of perturbed traffic corresponding to the same original traffic different. The differences can mitigate replay attacks by replaying the original traffic.

There are four cases in the compensation steps. Without loss of generality, we assume the next talk spurt or silence gap is the jth talk spurt or silence gap.

Recall that input traffic are consists of talk spurts and silence gaps, thus, optimization can be classified into two classes: talk spurt optimization and silence gap optimization. We will see that different classes should have different optimization method. For both talk spurt optimization and silence gap optimization, output value come up with two parts: (1) optimal value calculated by Pearson's Correlation Coefficient Formula, called O and (2) makeup value,called M, is generated by prediction error and random coefficient, If we define predicted value which generated by Optimization Model is P, Since both optimal value of talk spurt and silence gap could be either grater than or equal to predicted value or less than predicted value, we can divide each class into two subclasses. Based on these classes, we will discuss each in detail:

Situation1: talk spurt optimization and optimal value O is grater than or equal to predicted value P. As Figure 4.6 shows, with O and P given, actual talk spurt could end in following three place: $a1$: less than both O and P; $a2$: grater than P and less than O; $a3$: grater than both O and P.

$a1$: if actual talk spurt end at $a1$, Optimization Model keeps padding to $f1 = O - M$. M is generated by prediction error and random coefficient, in this case, $M = \theta(p - a1)$, where θ is random coefficient.

$a2$: if actual talk spurt end at $a2$, Optimization Model keeps padding to $f2 = O + M$. M is generated by prediction error and random coefficient, in this case, $M = \theta(a2 - p)$, where θ is random coefficient.

$a3$: if actual talk spurt end at $a3$, Optimization Model keeps padding to $f3 = O + M$. M is generated by prediction error and random coefficient, in this case, $M = \theta(a3 - p)$, where θ is random coefficient.

Situation2: talk spurt optimization and optimal value O is less than predicted value P. As Figure 4.7 shows, with O and P given, actual talk spurt could end in following three place: $a1$: less than both O and P; $a2$: grater than O and less than P; $a3$: grater than both O and P.

$a1$: if actual talk spurt end at $a1$, the optimization model will hold packets in buffer as long as possible and start to drop packets at $a1$. To minimize effect of prediction error and add randomness to the optimal value, optimization model will generate makeup value $M = \theta(p - a1)$, where θ is random coefficient. So final optimal value for this situation is $f1 = O - \theta(p - a1)$.

$a2$: if actual talk spurt end at $a2$, the optimization model will hold packets in buffer as long as possible and start to drop packets at $a2$. To minimize effect of prediction error and add randomness to the optimal value, optimization model will generate makeup value $M = \theta(p - a2)$, where θ is random coefficient. So final optimal value for this situation is $f2 = O - \theta(p - a2)$.

$a3$: if actual talk spurt end at $a1$, the optimization model will hold packets in buffer as long as possible and start to drop packets at $a3$. To minimize effect of prediction error and add randomness to the optimal value, optimization model will generate makeup value $M = \theta(a3 - p)$, where θ is random coefficient. So final optimal value for this situation is $f3 = O - \theta(a3 - p)$.

Situation3: silence gap optimization and optimal value O is grater than or equal to predicted value P. As Figure 4.8 shows, with O and P given, actual silence gap could end in following three place: $a1$: less than both O and P; $a2$: grater than P and less than O; $a3$: grater than both O and P.

$a1$:If actual silence gap end at a1, the optimization model will hold packets as long as possible and start to drop packets at $a1$. To minimize effect of prediction error and add randomness to the optimal value, optimization model will generate makeup value $M = \theta(p - a1)$, where θ is random coefficient. So final optimal value for this situation is $f1 = O - \theta(p - a1)$.

$a2$:If actual silence gap end at a2, the optimization model will hold packets as long as possible and start to drop packets at $a2$. To minimize effect of prediction error and add randomness to the optimal value, optimization model will generate makeup value $M = \theta(p - a2)$, where θ is random coefficient. So final optimal value for this situation is $f2 = O + \theta(p - a2)$.

$a3$:If actual silence gap end at a3, the optimization model will hold packets as long as possible and start to drop packets at $a3$. To minimize effect of prediction error and add randomness to the optimal value, optimization model will generate makeup value $M = \theta(a3 - p)$, where θ is random coefficient. So final optimal value for this situation is $f3 = O + \theta(a3 - p)$.

Situation4: silence gap optimization and optimal value O is less than predicted value P. As Figure 4.9 shows, with O and P given, actual silence gap could end in following three place: $a1$: less than both O and P; $a2$: grater than O and less than P; $a3$: grater than both O and P.

$a1$:If actual silence gap end at a1, the optimization model adds dummy packets to $f1 = O - M$. M is generated by prediction error and random coefficient, in this case, $M = \theta(p - a1)$, where θ is random coefficient.

$a2$:f actual silence gap end at a2, the optimization model adds dummy packets to $f2 = O - M$. M is generated by prediction error and random coefficient, in this case, $M = \theta(p - a2)$, where θ is random coefficient.

$a3$:If actual silence gap end at a3, the optimization model adds dummy packets to $f3 = O + M$. M is generated by prediction error and random coefficient, in this case, $M = \theta(a3 - p)$, where θ is random coefficient.

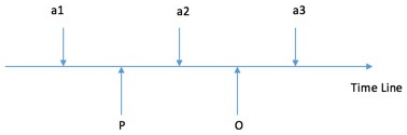


Figure 4.6: Situation 1

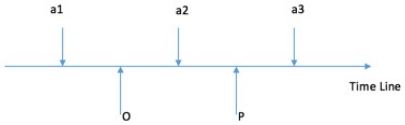


Figure 4.7: Situation 2

Summarize above situations, we can formally define these 4 algorithms:

$y_j^{o,t} \geq x_j^{p,t}$**:** In this case, $y_j^{o,t}$, the optimal length of the talk spurt from the previous step, is greater than or equal to $x_j^{p,t}$, the predicted length of the talk spurt from the prediction step. The compensation will be determined as follows: The prediction error is calculated as the difference between $x_j^{p,t}$, the predicted length of the jth talk spurt, and x_j^t, the actual length of the jth talk spurt. The compensation M is θ times the prediction error and θ is a random number. The random number θ is added to mitigate replay attacks and a different random number will be generated for each talk spurt or silence gap. The length of the talk spurt in the output traffic is determined based on the optimal length of the talk spurt and the compensation. The pseudo-code of the compensation in this case is shown in Algorithm 1.

$y_j^{o,t} < x_j^{p,t}$**:** In this case, $y_j^{o,t}$, the optimal length of the talk spurt from the previous step, is less than $x_j^{p,t}$, the predicted length of the talk spurt from the prediction step. The compensation will be determined as follows: The prediction error is calculated as the difference between $x_j^{p,t}$, the predicted length of the jth talk spurt, and x_j^t, the actual length of the jth talk spurt. The compensation M is θ times the prediction error and θ is a random number. The random number θ is added to mitigate replay attacks and a different random number will be generated for each talk spurt or silence gap. The length of the talk spurt in the output traffic is determined

Algorithm 1: Compensation in Case $y_j^{o,t} \geq x_j^{p,t}$

Data: x_j^t: the actual length of the jth talk spurt
$y_j^{o,t}$: the optimal length of the jth talk spurt
$x_j^{p,t}$: the predicted length of the jth talk spurt
τ: packetization delay
t_j: the end of the jth talk spurt
$t \longleftarrow t_j$;
generate a random number θ between 0 and θ_{max};
if $x_j^t \leq x_j^{p,t}$ **then**
    $M \longleftarrow \theta(x_j^{p,t} - x_j^t)$;
    while *the add rate of dummy packets is less than* lim_{add} **do**
        $t \longleftarrow t + \tau$;
        if $t < y_j^{o,t} - M$ **then**
            add a dummy packet at t;
        else
            break;
        end
    end
else
    $M \longleftarrow \theta(x_j^t - x_j^{p,t})$;
    while *the add rate of dummy packets is less than* lim_{add} **do**
        $t \longleftarrow t + \tau$;
        if $t < y_j^{o,t} + M$ **then**
            add a dummy packet at t;
        else
            break;
        end
    end
end

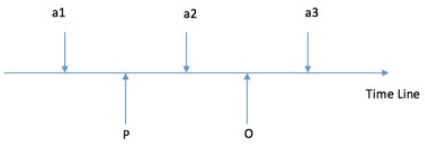


Figure 4.8: Situation 3

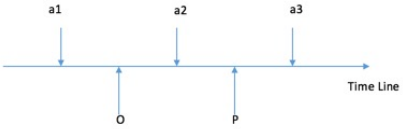


Figure 4.9: Situation 4

based on the optimal length of the talk spurt and the compensation. The pseudo-code of the compensation in this case is shown in Algorithm 2.

$y_j^{o,s} \geq x_j^{p,s}$**:** In this case, $y_j^{o,s}$, the optimal length of the silence gap from the previous step, is greater than or equal to $x_j^{p,s}$, the predicted length of the silence gap from the prediction step. The compensation will be determined as follows: The prediction error is calculated as the difference between $x_j^{p,s}$, the predicted length of the jth silence gap, and x_j^s, the actual length of the jth silence gap. The compensation M is θ times the prediction error and θ is a random number. The random number θ is added to mitigate replay attacks and a different random number will be generated for each talk spurt or silence gap. The length of the talk spurt in the output traffic is determined based on the optimal length of the talk spurt and the compensation. The pseudo-code of the compensation in this case is shown in Algorithm 3.

$y_j^{o,s} < x_j^{p,s}$**:** In this case, $y_j^{o,s}$, the optimal length of the silence gap from the previous step, is less than $x_j^{p,s}$, the predicted length of the silence gap from the prediction step. The compensation will be determined as follows: The prediction error is calculated as the difference between $x_j^{p,s}$, the predicted length of the jth silence gap, and x_j^s, the actual length of the jth silence gap. The compensation M is θ times the prediction error and θ is a random number. The random number θ is added to mitigate replay attacks and a different random number will be generated

Algorithm 2: Compensation in Case $y_j^{o,t} < x_j^{p,t}$

Data: x_j^t: the actual length of the jth talk spurt
$y_j^{o,t}$: the optimal length of the jth talk spurt
$x_j^{p,t}$: the predicted length of the jth talk spurt
τ: packetization delay
t_j: the end of the jth talk spurt
$t \longleftarrow t_j$;
generate a random number θ between 0 and θ_{max};
if $x_j^t \leq y_j^{o,t}$ **then**
    $M \longleftarrow \theta(x_j^{p,t} - x_j^t)$;
    while *the drop rate of actual packets is less than* lim_{drop} **do**
        $t \longleftarrow t + \tau$;
        if $t < y_j^{o,t} - M$ **then**
            drop actual packets at t;
        else
            break;
        end
    end
else
    $M \longleftarrow \theta(x_j^t - x_j^{p,t})$;
    while *the drop rate of actual packets is less than* lim_{drop} **do**
        $t \longleftarrow t + \tau$;
        if $t < y_j^{o,t} - M$ **then**
            drop actual packets at t;
        else
            break;
        end
    end
end

Algorithm 3: Compensation in Case $y_j^{o,s} \geq x_j^{p,s}$

Data: x_j^s: the actual length of the jth silence gap
$y_j^{o,s}$: the optimal length of the jth silence gap
$x_j^{p,s}$: the predicted length of the jth silence gap
τ: packetization delay
s_j: the end of the jth silence gap
$t \longleftarrow t_j$;
generate a random number θ between 0 and θ_{max};
switch x_j^s **do**
  case $x_j^s \leq x_j^{p,s}$
    $M \longleftarrow \theta(x_j^{p,s} - x_j^s)$;
    while *the drop rate of actual packets is less than* lim_{drop} **do**
      $t \longleftarrow t + \tau$;
      if $t < y_j^{o,s} - M$ **then**
        drop actual packet at t;
      else
        break;
      end
    end
  end
  case $x_j^s > x_j^{p,s}$ *and* $x_j^s \leq y_j^{o,s}$
    $M \longleftarrow \theta(x_j^{p,s} - x_j^s)$;
    while *the drop rate of actual packets is less than* lim_{drop} **do**
      $t \longleftarrow t + \tau$;
      if $t < y_j^{o,s} + M$ **then**
        drop actual packet at t;
      else
        break;
      end
    end
  end
  case $x_j^s > y_j^{o,s}$
    $M \longleftarrow \theta(x_j^s - x_j^{p,s})$;
    while *the drop rate of actual packets is less than* lim_{drop} **do**
      $t \longleftarrow t + \tau$;
      if $t < y_j^{o,t} + M$ **then**
        drop actual packet at t;
      else
        break;
      end
    end
  end
endsw

for each talk spurt or silence gap. The length of the talk spurt in the output traffic is determined based on the optimal length of the talk spurt and the compensation. The pseudo-code of the compensation in this case is shown in Algorithm 4.

Algorithm 4: Compensation in Case $y_j^{o,s} < x_j^{p,s}$

```
Data: x_j^s: the actual length of the jth silence gap
y_j^{o,s}: the optimal length of the jth silence gap
x_j^{p,s}: the predicted length of the jth silence gap
τ: packetization delay
t_j: the end of the jth silence gap
t ⟵ t_j;
generate a random number θ between 0 and θ_max;
if x_j^s ≤ y_j^{o,s} then
    M ⟵ θ(x_j^{p,s} − x_j^s);
    while the add rate of dummy packets is less than lim_add do
        t ⟵ t + τ;
        if t < y_j^{o,s} − M then
            add dummy packets at t;
        else
            break;
        end
    end
else
    M ⟵ θ(x_j^s − x_j^{p,s});
    while the add rate of dummy packets is less than lim_add do
        t ⟵ t + τ;
        if t < y_j^{o,s} + M then
            add dummy packets at t;
        else
            break;
        end
    end
end
```

CHAPTER 5. EXPERIMENT AND RESULT

In this chapter, we evaluate the performance of the pattern hiding module. The evaluation is on the effectiveness of pattern hiding and resistance to replay attacks.

5.1 Experiment Setup

In order to get natural audio traces for our experiment, we set up the experiment as Figure 5.1. Basically, we collect 40 speeches from YouTube.com for the experiment. The length of the speeches is between 10 and 15 minutes. We feed the speeches to the X-Lite 3.0 VoIP client software. Detail shows follow:

1. Software

   In our experiment, we use two machines (a data collection machine(Computer1) and a support machine(Computer2)) which both installed X-Lite 3.0 for the network communication. We also installed Wireshark 1.12.2 on the data collection machine that use for catch the packets.

   X-Lite 3.0: is used as VoIP software that send audio packet from a computer to another computer. For the codec part, we choose the μlaw codec in X-Lite to covert the speeches into VoIP packets due to the popularity of the μlaw codec.

   Wiresharks 1.12.2: is used for collect packet between above two computers.

2. SIP Account

   In this experiment, we used same network communication tool in two different machines with two SIP Account: jlfang@sip2sip.info and jlfang@iptel.org.

3. Data Collection

   After we setup the two X-Lite on both data collection and support machines, we made a call from the data collection machine to the support machine and at the same time, we run the data collection software: Wireshark on the data collection machine. Then we play speeches h that we collected from Youtube.com. Picture **??** is the picture for part of data we collected. From this picture, we can easily identify the packet number, packet send time, source, destination, protocol and other specific data information.

4. Data Analysis After we collected data from Wireshark 1.12.2, we import these data into Microsoft Excel for further analysis. In this analysis, we mostly focused on the time interval between two packets in different situation: talk and silence, which related to our research information: talk spurt and silence gap.

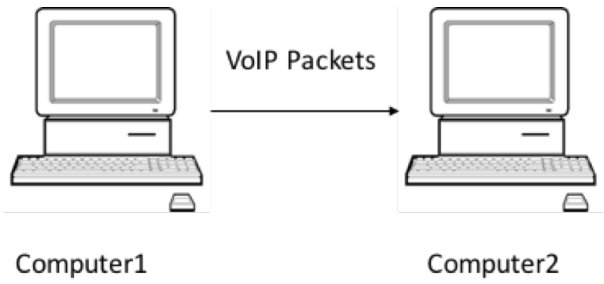


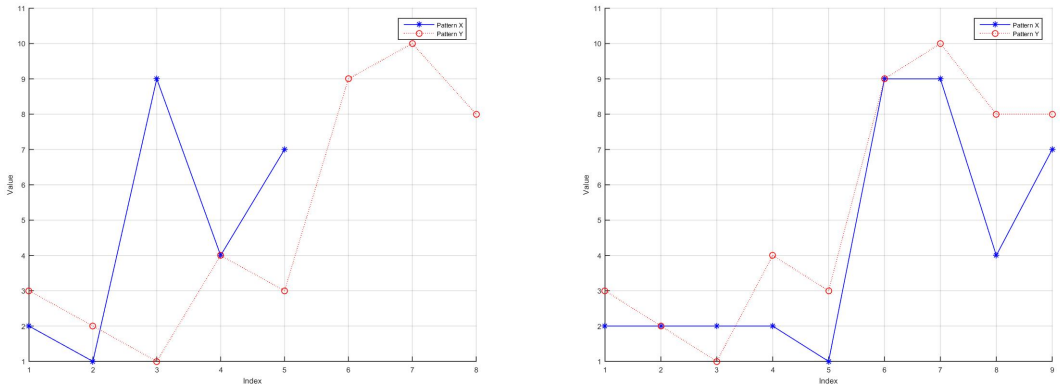
Figure 5.1: Data Collection Mode

5.2 Performance Metrics

We use DTW correlation, a correlation metric based on the Dynamic Time Warping (DTW) algorithm to evaluate the hiding performance. We do not use Pearson's correlation defined in (4.1) because silence gaps may be covered by dummy packets and talk spurts may be removed through packet drops. The "missing" data can significantly reduce Pearson's correlation and an adversary has no idea on the location of the "missing" talk spurts and silence gaps because the adversary has no access to content of encrypted VoIP packets.

A classical approach to measure similarity between two time series of different length is the DTW algorithm, which has been used in various traffic analysis research topics such as website

fingerprinting [7] and denial of service (DoS) attack detection [10]. In this research project, we use the DTW algorithm to find the best alignment of the on-off pattern in the input traffic and the on-off pattern in the output traffic. The DTW correlation is calculated as Pearson's correlation of the aligned on-off patterns in the input traffic and in the output traffic. As shown in Figure 5.2.(a), the two on-off patterns, represented by $X = [x_1, x_2, \cdots, x_i, \cdots, x_m]$ and $Y = [y_1, y_2, \cdots, y_j, \cdots, y_n]$ respectively, are of different length. The DTW algorithm find the best alignment function $f(i) = j$ where i and j are the indexes of the X and Y vectors. The best alignment minimizes the distance between the two vectors defined as $Dist = \sum_{i=1}^{m} |x_i - y_{f(i)}|$. Usually the dynamic programming is used to minimize the distance. Figure 5.2.(b) shows the aligned vectors.



(a) Original Patterns (b) Warped Patterns

Figure 5.2: Pattern Alignment with DTW

5.3 Pattern Hiding Performance

Figure 5.3 shows the hiding performance with various rate limits on dummy packets (lim_{add}). We have the following observation from these experiments:

1. When lim_{add}, the rate limit on adding dummy packets, increases, the DTW correlation decreases. The trend is expected as more dummy packets can fill more silence gaps and in turn hide traffic patterns more effectively.

2. The two curves in Figure 5.3 are close to each other. It means:

   (a) For the same rate limit on dummy packets (lim_{add}), the 5% increase in the limit of drop rate (lim_{drop}) and 100ms increase in the delay limit (lim_{delay}) can only slightly improve the hiding performance.

   (b) The hiding performance changes significantly with the rate limit on dummy packets (lim_{add}). From our experiment data, we also observe that the actual dummy packet rate is much lower than the limit lim_{add}. For example, a typical actual dummy packet rate is 42%.46 when lim_{add} is 100%. The limit lim_{add} is not fully utilized as the optimization solutions may not lie at the constraint boundaries.

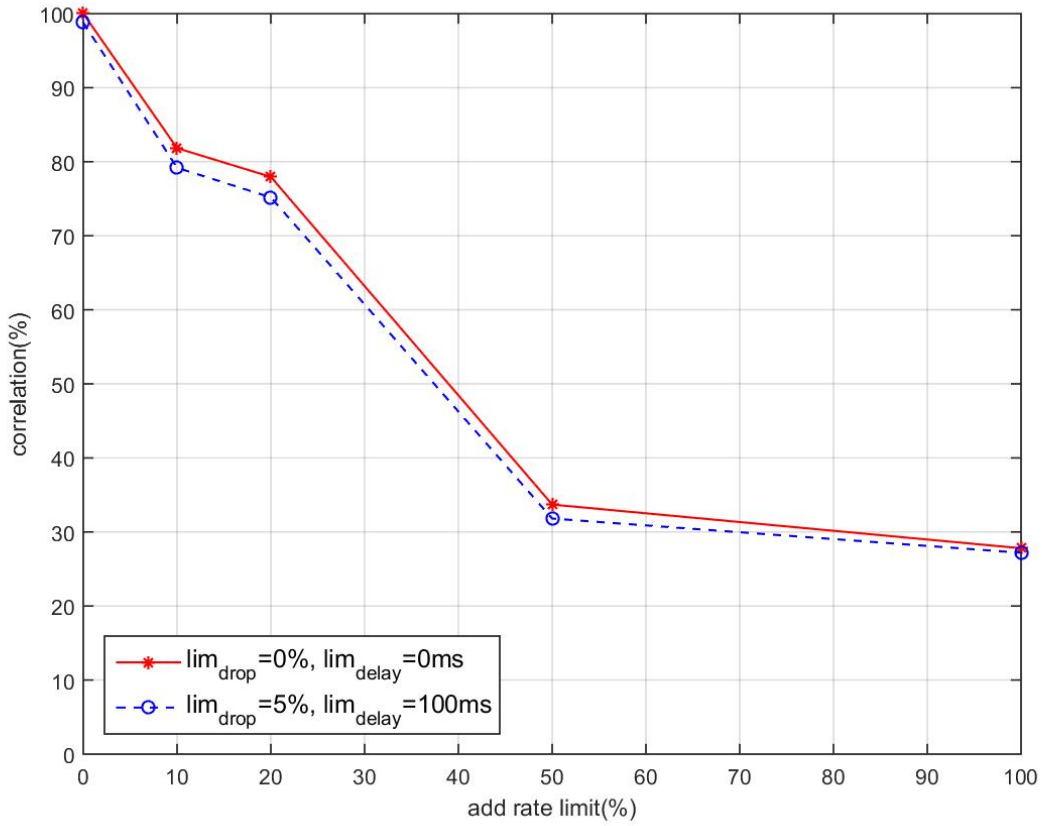


Figure 5.3: Limit on Adding Dummy Packets (lim_{add})

Figure 5.4 shows hiding performance under various limits on packet drop rate (lim_{drop}). We have the following observation from these experiments:

1. The DTW correlation decreases when the limit on the drop rate increases. It is because more packet drops can also lead to better pattern hiding.

2. When the limit lim_{drop} approaches 100%, the DTW correlation is still close to 0.7. We checked the experiment data and found that the typical drop rate was 43.52%, still far far from 100% when the limit lim_{drop} was 100%. It is because the optimization solutions may not occur at the constraint boundaries. For VoIP communications, a large drop rate causes significant QoS degradation and conversations may not be able to continue. So in the following experiments, we limit the drop rate within 5%.

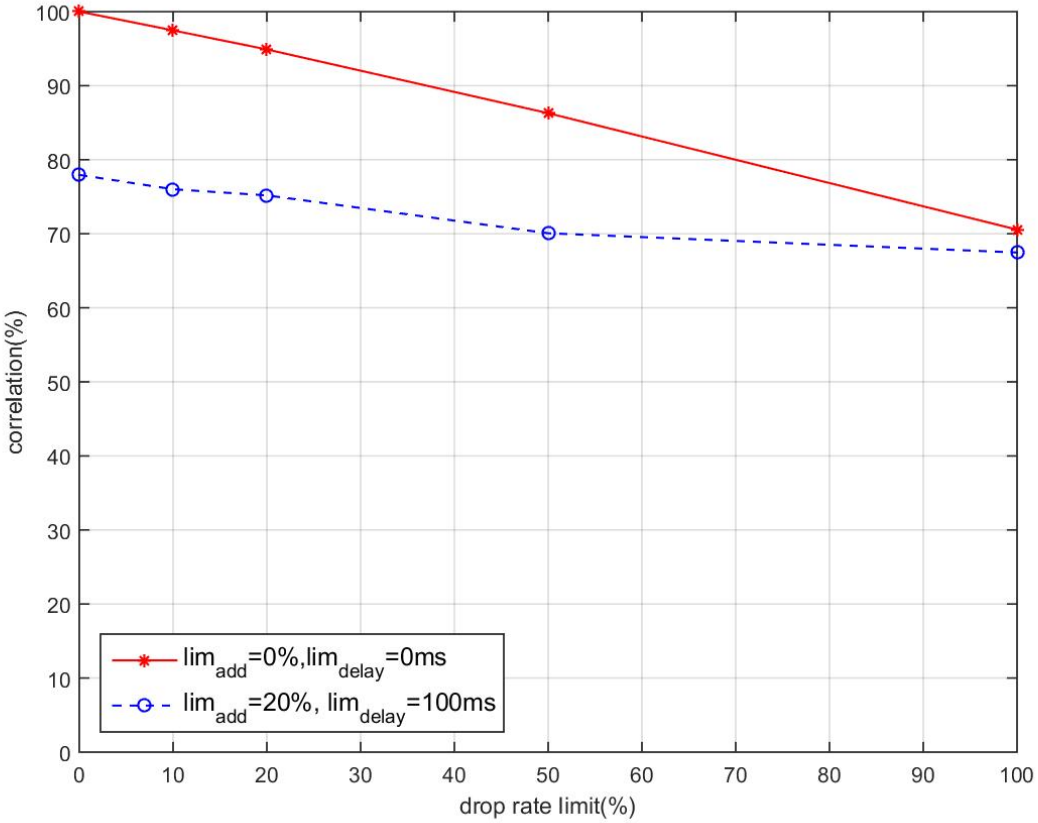


Figure 5.4: Limit on Packet Drop Rate (lim_{drop})

Figure 5.5 shows hiding performance with various delay limits on VoIP packets. We have the following observation from these experiments:

1. The hiding performance improves when the delay limit increases. It is consistent with our intuition as a larger delay limit gives the optimization module more flexibility in scheduling VoIP packets to optimize the pattern hiding.

2. We can also observe that when the rate limit on dummy packets is 20% and the limit on the drop rate are 5%, the pattern hiding performance does not improve significantly when the delay limit increases.

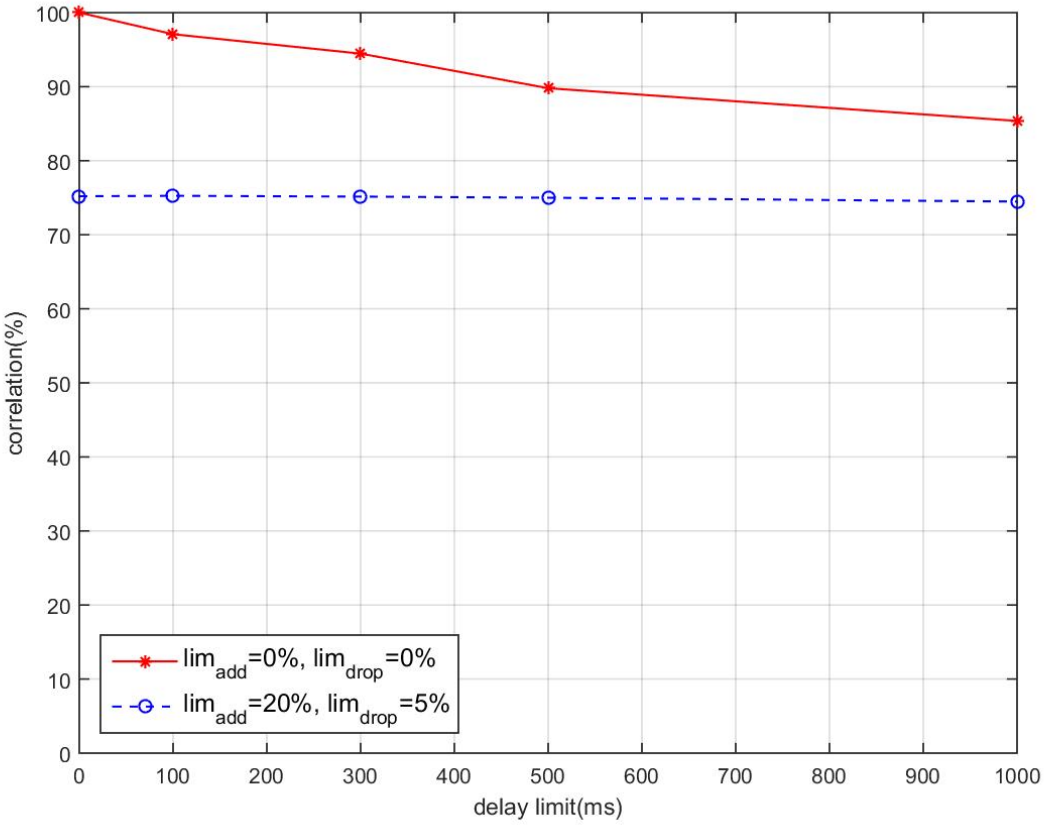


Figure 5.5: Limit on Packet Delay lim_{delay}

5.4 Resistance to Replay Attacks

In this set of experiments, we replay the same speech to the pattern hiding module. The goal of the replay attacks is to find out the output traffic traces that are generated from the same speech. The resistance to the replay attacks is evaluated with the detection rate, defined as the

ratio of the correct detections to the number of attempts. In each attempt, the candidate pool has one trace generated from the same speech as the trace of interest and 19 traces generated from other speeches. So a random guess results in a detection rate of $\frac{1}{19}$.

Figure 5.6 shows the detection rate with various limits on the dummy packets, packet drop rate, and packet delay. We make the following observations from the experiment results:

1. In both curves, the detection rate decreases when the limit on the dummy packet rate (lim_{add}) increases. When $lim_{add} = 100\%, lim_{drop} = 5\%, and lim_{delay} = 100ms$, the detection rate reaches 24%, close to the detection rate of a random guess.

2. A increase of lim_{drop} from 0 to 5% and a increase of lim_{delay} from 0ms to 100ms can bring down the detection rates by around 5% when $lim_{add} \geq 20\%$.

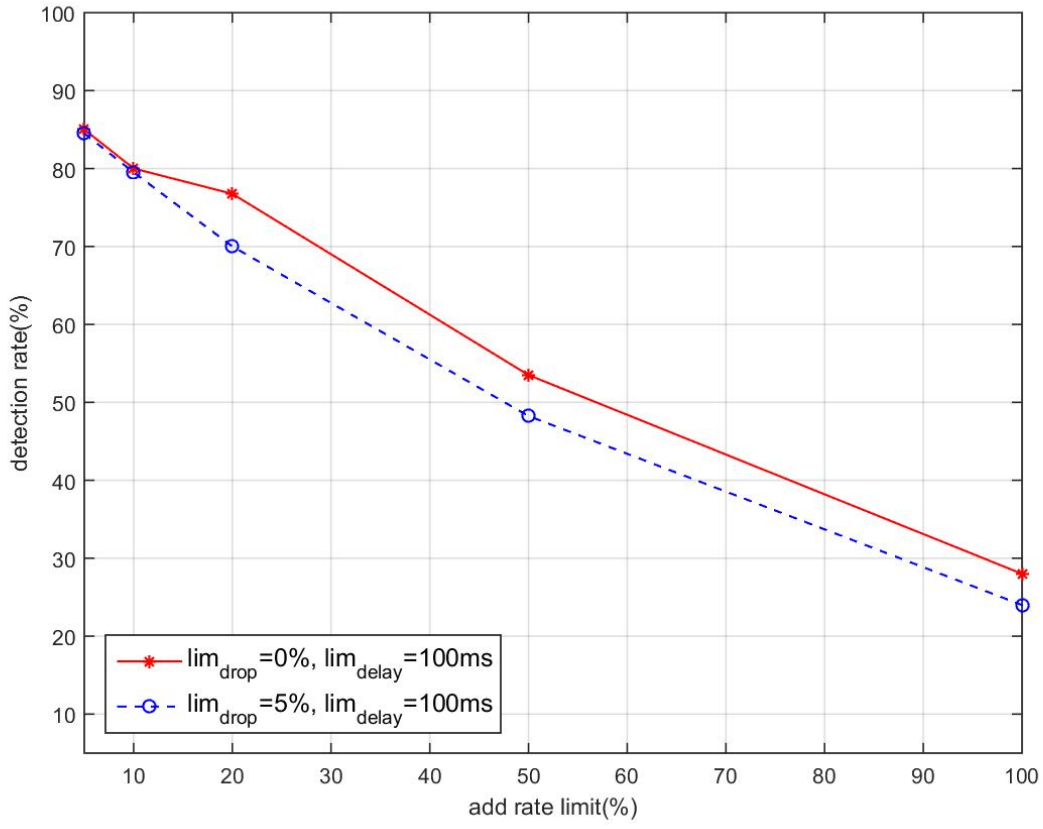


Figure 5.6: Different Application Performance

CHAPTER 6. CONCLUSION AND DISCUSSION

In this section, we discuss the optimization step, the experiments, and extension of the hiding approach.

We use Pearson's correlation in the optimization step and use the DTW correlation in evaluating the hiding approach. We choose Pearson's correlation instead of the DTW correlation in the optimization step because of the following two reasons:

1. The DTW correlation also contains an optimization process that finds the best alignment of the input traffic pattern and the output traffic pattern. Usually dynamic programming is used for the optimization. So the optimization process is time-consuming and it is not suitable for the online optimization required by the hiding approach.

2. The optimization based on Pearson's correlation has closed-form solutions. So the optimization can be finished in 5ms, which is much shorter than even the packetization delay of VoIP packets. As we explain in the previous section, Pearson's correlation can not be used for evaluation as talk spurts and silence gaps can be removed by the hiding approach.

We evaluate the effectiveness of the hiding approach on its resistance to replay attacks. Essentially the replay attack is equivalent to the speech identification, which aims to identify traffic traces generated from the same speech. In our future work, we plan to evaluate the effectiveness of the pattern hiding approach with other identification tasks such as speaker identification and language identification. We choose speech identification in this paper because the speech identification can achieve much higher identification rates than other identification tasks when no pattern hiding approach is in use.

CHAPTER 7. SUMMARY AND FUTURE WORK

7.1 Summary

Considering the threat of VoIP attackers, and inspired by the work of researchers on VoIP traffic pattern recognition, we propose a pattern hiding approach to mitigate traffic analysis attacks on VoIP communications. The approach hides traffic patterns by adding dummy packets, dropping VoIP packets, and delaying VoIP packets. The approach optimizes pattern hiding in terms of dissimilarity from the original traffic pattern and the optimization is under constraints on dummy packet rate, VoIP packet drop rate, and VoIP packet delay. Our contributions to the field of voice pattern hiding for VoIP communications are as follows:

- A adversary behavior model which lunches traffic analysis attacks.
- New pattern hiding module that are able to hide voice traffic pattern in Internet.
- New measuring method that used to measure the effectiveness in countering traffic analysis attacks.

In Chapter 2, we included a literature survey of existing researches on Internet and VoIP communication security to show their advantages and defect. Consequently, we summarized their common defect as follows: unable to prevent timing-based traffic analysis attack.

In Chapter 3, we defined the problem statement and our assumption. So we normalized the behavior of adversary and our pattern hiding module.

In Chapter 4, we formally proposed our pattern hiding module. Our pattern hiding module has 3 steps: prediction step, optimization step and compensation step. Prediction step provide forecast value of talk spurts and silence gap which will be used in later optimization step. Optimization step provide optimal value which minimized the correlation between original

time series and new time series and then pass it to compensation step. Compensation step modify the optimal value based on constrain and give the final value and position of packets that will be output to the Internet.

In Chapter 5, we set up a series of experiments and analyzed the experiment result. First, we collect 40 audio traces from Youtube.com then we set up four experiment based on these traces, which used to test different module function. For the first three experiments, we test the following methods: add dummy packets, drop original packets and delay original packets. Experiment result shows that all these methods are able to decrease the correlation between new times series and original time series just like our hypothesis. The fourth experiment, we designed to test our pattern hiding module's resistance to relay attacks. The experiment result shows that, with appropriate constraint, our module has very high resistance to these attackers who even knows our module design theory and structure.

In Chapter 6, we discussed our optimization step, experiments and extension of the hiding approach. First, we explained why we used two different correlation techniques in the module. Then we evaluated the experiment result and effectiveness on relay attacks: our experiments show the hiding approach can effectively hide traffic patterns and resist replay attacks to identify the same speech.

7.2 Future Work

In this thesis, we focus on hiding the on-off pattern in VoIP communications. We believe the approach can also be extended to hide traffic patterns in other communications with various QoS requirements. The approach can also be more effective for delay-tolerant communications such as email and ftp because of the removal of the delay constraints. We plan to work on the extension in our future work.

APPENDIX . SOLUTION OF THE OPTIMIZATION PROBLEM

The objective function (4.2) can be simplified as follows:

$$D(y_j^{o,s}) = \frac{ay_j^{o,s} + b}{\sqrt{cy_j^{o,s2} + dy_j^{o,s} + e}} \tag{3}$$

where $a = \sum_{i=j-n+1}^{j} \frac{\bar{x}-x_i^t}{2n} + \sum_{i=j-n+1}^{j-1} \frac{\bar{x}-x_i^s}{2n} + [(\frac{2n+1}{2n})x_j^{p,s} - (\frac{2n-1}{2n})\bar{x}]$,

$b = \sum_{i=j-n+1}^{j-1}(\bar{x}-x_i^t)\frac{\sum_{i=j-n+1}^{j} y_i^t + \sum_{i=j-n+1}^{j-1} y_i^s}{2n} + \sum_{i=j-n+1}^{j-1}(\bar{x}-x_i^s)\frac{\sum_{i=j-n+1}^{j} y_i^t + \sum_{i=j-n+1}^{j-1} y_i^s}{2n} + ((\bar{x}-x_j^{p,s})\frac{\sum_{i=j-n+1}^{j} y_i^t + \sum_{i=j-n+1}^{j-1} y_i^s}{2n})$,

$c = (\delta+\epsilon+\zeta)(\sum_{i=j-n+1}^{j} 1 + \sum_{i=j-n+1}^{j-1} 1 + 1)$,

$d = (\delta+\epsilon+\zeta)(\sum_{i=j-n+1}^{j} \frac{\alpha}{n} + \sum_{i=j-n+1}^{j-1} \frac{\beta}{n} + \frac{2n-1}{n})$,

$e = (\delta+\epsilon+\zeta)(\sum_{i=j-n+1}^{j} \alpha^2 + \sum_{i=j-n+1}^{j-1} \beta^2 + \gamma^2)$,

$\alpha = \frac{2ny_i^t - \sum_{i=j-n+1}^{j} y_i^t + \sum_{i=j-n+1}^{j-1} y_i^s}{2n}$,

$\beta = \frac{2ny_i^s - \sum_{i=j-n+1}^{j} y_i^t + \sum_{i=j-n+1}^{j-1} y_i^s}{2n}$,

$\gamma = -\frac{\sum_{i=j-n+1}^{j} y_i^t + \sum_{i=j-n+1}^{j-1} y_i^s}{2n}$,

$\delta = \sum_{i=j-n+1}^{j}(x_i^t - \bar{x})^2$, $\epsilon = \sum_{i=j-n+1}^{j}(x_i^s - \bar{x})^2$, and $\zeta = (x_j^{p,s} - \bar{x})^2$.

The derivative of $D(y_j^{o,s})$ is

$$D(y_j^{o,s})' = \frac{(ad-bc)y_j^{o,s} + 2ea - bd}{2(cy_j^{o,s2} + dy_j^{o,s} + e)^{\frac{3}{2}}} \tag{4}$$

To find the critical point, we solve the equation $D'(y_j^{o,s}) = 0$. So the critical point is $y_j^{o,s} = \frac{2ea-bd}{2bc-ad}$.

To find out whether the minimum occurs at the critical point, we calculate the second derivative of $D(y_j^{o,s})$ as follows:

$$D''(y_j^{o,s}) = \frac{-4acdy_j^{o,s2} - 12acex - ad^2y_j^{o,s} - 4ade}{4(cy_j^{o,s2} + dy_j^{o,s} + e)^{\frac{5}{2}}} + \frac{8bc^2y_j^{o,s2} + 8bcdy_j^{o,s} - 4bce + 3bd^2}{4(cy_j^{o,s2} + dy_j^{o,s} + e)^{\frac{5}{2}}} \tag{5}$$

So if $D''(y_j^{o,s}) > 0$ when $y_j^{o,s} = \frac{2ea-bd}{2bc-ad}$, the minimum occurs at the critical point. Otherwise the minimum occurs at the end points defined by the constraints.

REFERENCES

[1] X-Lite 3.0 free softphone. Available: http://www.xten.com/index.php?menu=Products&smenu=xlite.

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

[3] P. T. Brady. A technique for investigating on-off patterns of speech. *The Bell System Technical Journal*, 44.

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

[5] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proc. of the 13th USENIX Security Symposium*, pages 303–320, San Diego, CA, August 2004.

[6] F. Fessant, S. Bengio, and D. Collobert. On the prediction of solar activity using different neural network models. *Annales Geophysicae*, 14(1):20–26.

[7] Xun Gong, Nikita Borisov, Negar Kiyavash, and Nabil Schear. *Privacy Enhancing Technologies: 12th International Symposium, PETS 2012, Vigo, Spain, July 11-13, 2012. Proceedings*, chapter Website Detection Using Remote Traffic Analysis, pages 58–78. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[8] Yong Guan, Xinwen Fu, Dong Xuan, P. U. Shenoy, R. Bettati, and Wei Zhao. Netcamo: camouflaging network traffic for qos-guaranteed mission critical applications. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 31(4):253–265, Jul 2001.

[9] S.S. Haykin. *Neural Networks: A Comprehensive Foundation*. International edition. Prentice Hall, 1999.

[10] Xiapu Luo, Edmond W. W. Chan, and Rocky K. C. Chang. Detecting pulsing denial-of-service attacks with nondeterministic attack intervals. *EURASIP J. Adv. Signal Process*, 2009:8:1–8:13, January 2009.

[11] A. Panchenko, F. Lanze, and T. Engel. Improving performance and anonymity in the tor network. In *2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC)*, pages 1–10, Dec 2012.

[12] Dan W. Patterson. *Artificial Neural Networks: Theory and Applications*. Prentice-Hall Series in Advanced Communications. Prentice Hall, 1996.

[13] J. M. Valin. Speex: A free codec for free speech. In *Australian National Linux Conference*, 2006.

[14] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot me if you can: Uncovering spoken phrases in encrypted voip conversations. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 35–49, May 2008.

[15] Charles V. Wright, Lucas Ballard, Fabian Monrose, and Gerald M. Masson. Language identification of encrypted voip traffic: Alejandra y roberto or alice and bob? In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, SS'07, pages 4:1–4:12, Berkeley, CA, USA, 2007. USENIX Association.

[16] Ye Zhu and Huirong Fu. Traffic analysis attacks on skype voip calls. *Comput. Commun.*, 34(10):1202–1212, July 2011.

[17] Ye Zhu, Yuanchao Lu, and Anil Vikram. On privacy of encrypted speech communications. *IEEE Trans. Dependable Secur. Comput.*, 9(4):470–481, July 2012.

[18] P. Zimmermann, A. Johnston, and J. Callas. Zrtp: Media path key agreement for secure rtp draft-zimmermann-avt-zrtp-11. RFC,United States, 2008.