Voice Pattern Hiding for VoIP Communications

Jialue Fang
*Department of Electrical
and Computer Engineering
Iowa State University
Ames, Iowa 50011
jlfang@iastate.edu*

Ye Zhu
*Electrical Engineering and
Computer Science Department
Cleveland State University
Cleveland, Ohio 44115
y.zhu61@csuohio.edu*

Yong Guan
*Department of Electrical
and Computer Engineering
Iowa State University
Ames, Iowa 50011
guan@iastate.edu*

Abstract—**In this paper, we address issues related to traffic analysis attacks and the corresponding countermeasures in Voice over IP(VoIP) traffic. VoIP is widely used in today's communication, it converts analog voice signals into digital data packets and supports real-time, two-way transmission of conversations using Internet Protocol. In this paper, 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.**

1. Introduction

Voice over IP (VoIP) communications are continuing gaining popularity due to their cost savings and rich features. Numerous efforts such as SRTP [1] and ZRTP used in Zfone [2] 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 [3], identify languages used into the VoIP communications [4], and identify speakers [5].

In this paper, 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 major contributions are summarized as follows:

- We formally model 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.
- We develop a pattern hiding module and measure the effectiveness in countering traffic analysis attacks.

The remainder of this paper is organized as follows: Section 2 reviews the related work. In Section 3, we outline the background of the problem and definition of some attack methods and technologies. Section 4 introduces threat model. Section 5 presents the pattern hiding module. In Section 6, we evaluate the performance of our method. Section 7 discuss the optimization step, the experiments and extension of the hiding approach. Consequently, in Section 8 we conclude this paper and discuss the future work.

2. Related Work

In this section, we review previous work that is related to our study. Anonymous communication has been proved very useful for hiding user's identify from outside observer. The most famous anonymous application on web browser Tor [6] 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.

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, 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 [5]. Some recent research shows that when the audio is encoded using variable bit rate codecs, the length of encrypted VoIP packets can be used

978-1-5090-2279-3/16/$31.00 ©2016 IEEE

to identify the phrase spoken within a call and the language of the conversation. [3] [4]

Some of the countermeasure methods have been developed for hiding network traffic. For example, NetCamo [7] is able to camouflage network traffic.par In [6], Tor proved to be a useful for web browsing anonymous, but it is not able to effectively hide voice traffic. In paper [5] [3] [4], the length of encrypted VoIP packets are being used to identify users and languages. NetCamo [7] 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.

3. Background

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 1 shows an example of the silence suppression. Figure 1.(a) shows the waveform of a sheriff's voice signal extracted from a video published at cnn.com [8]. Figure 1.(b) shows the packet train generated by feeding the voice signal to X-Lite [9], a popular speech communication tool. From Figure 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 1.(b) can leak sensitive information.

4. Threat Model

Our goal is to design a module to hide the on-off traffic pattern shown in Figure 1. As shown in Figure 2, 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.

We assume the VoIP traffic is encrypted with one of the secure versions of the RTP protocol such as SRTP [1] or ZRTP used in Zfone [2] 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.

We assume an adversary is able to eavesdrop VoIP traffic to and from the computer running VoIP software. Since VoIP packets are encrypted and of the same length, the adversary attempts to disclose sensitive information through timing of VoIP packets.

5. Pattern Hiding Module

5.1. 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:

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

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 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. So the module can be essentially formulated

[1] Variable bit rate (VBR) codecs are primarily used for coding audio files instead of real-time speech communications [10], [11].

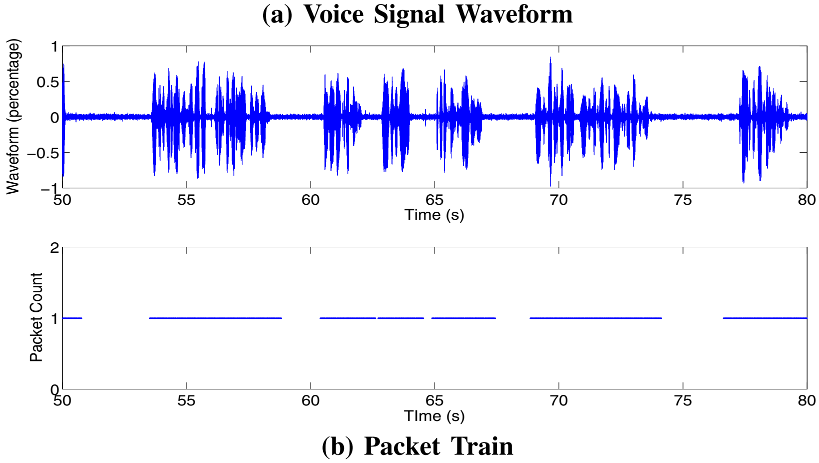


Figure 1: An Example of Silence Suppression

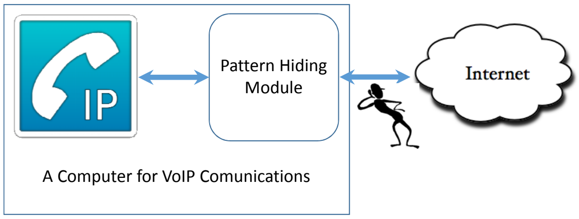


Figure 2: Pattern Hiding Module

as an optimization problem: The goal is to minimize the objective function defined in Equation 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}).

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.

The pattern hiding module has 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.

5.2. 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 [13]. The neural network used in this paper is the nonlinear autoregressive network with exogenous inputs (NARX) model [14]. As shown in Figure 3, the NARX model used in this paper is a two-layer feedforward network with one hidden layer and one output layer. In Figure 3, 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.

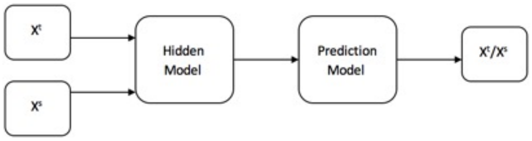


Figure 3: NARX Model Used to Predict Length of Silence Gaps (X^t: a vector of talk spurts, $X^s : avectorofsilencegaps.$)

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.

5.3. 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 (2). In (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 (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 A.

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.

5.4. 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.

$y_j^{o,t} \geq x_j^{p,t}$**:** 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.

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

```
Data: $x_j^t$: the actual length of the $j$th talk spurt
$y_j^{o,t}$: the optimal length of the $j$th talk spurt
$x_j^{p,t}$: the predicted length of the $j$th talk spurt
$\tau$: packetization delay
$t_j$: the end of the $j$th talk spurt
$t \longleftarrow t_j$;
generate a random number $\theta$ between 0 and $\theta_{max}$;
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
```

$y_j^{o,t} < x_j^{p,t}$**:** 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

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

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 2.

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

```
Data: $x_j^t$: the actual length of the $j$th talk spurt
$y_j^{o,t}$: the optimal length of the $j$th talk spurt
$x_j^{p,t}$: the predicted length of the $j$th talk spurt
$\tau$: packetization delay
$t_j$: the end of the $j$th talk spurt
$t \longleftarrow t_j$;
generate a random number $\theta$ between 0 and $\theta_{max}$;
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
```

$y_j^{o,s} \geq x_j^{p,s}$: 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 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.

6. Performance Evaluation

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

6.1. Experiment Setup

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. We choose the μlaw codec in X-Lite to covert the speeches into VoIP packets due to the popularity of the μlaw codec.

6.2. 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 (1) because silence gaps may be covered by

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

```
Data: $x_j^s$: the actual length of the $j$th silence gap
$y_j^{o,s}$: the optimal length of the $j$th silence gap
$x_j^{p,s}$: the predicted length of the $j$th silence gap
$\tau$: packetization delay
$s_j$: the end of the $j$th silence gap
$t \longleftarrow t_j$;
generate a random number $\theta$ between 0 and $\theta_{max}$;
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
```

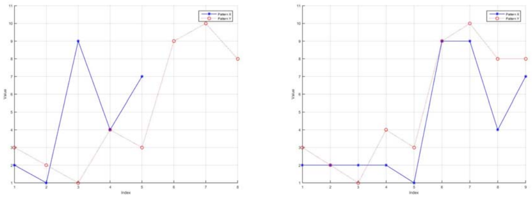
Algorithm 4: Compensation in Case $y_j^{o,s} < x_j^{p,s}$

```
Data: $x_j^s$: the actual length of the $j$th silence gap
$y_j^{o,s}$: the optimal length of the $j$th silence gap
$x_j^{p,s}$: the predicted length of the $j$th silence gap
$\tau$: packetization delay
$t_j$: the end of the $j$th silence gap
$t \longleftarrow t_j$;
generate a random number $\theta$ between 0 and $\theta_{max}$;
if $x_j^s \leq y_j^{o,s}$ then
    $M \longleftarrow \theta(x_j^{p,s} - x_j^s)$;
    while the add rate of dummy packets is less than $lim_{add}$ do
        $t \longleftarrow t + \tau$;
        if $t < y_j^{o,s} - M$ then
            add dummy packets at $t$;
        else
            break;
        end
    end
else
    $M \longleftarrow \theta(x_j^s - x_j^{p,s})$;
    while the add rate of dummy packets is less than $lim_{add}$ do
        $t \longleftarrow t + \tau$;
        if $t < y_j^{o,s} + M$ then
            add dummy packets at $t$;
        else
            break;
        end
    end
end
```

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 [15] and denial of service (DoS) attack detection [16]. 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 4.(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 4.(b) shows the aligned vectors.



(a) Original Patterns (b) Warped Patterns

Figure 4: Pattern Alignment with DTW

6.3. Pattern Hiding Performance

Figure 5 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 are close to each other. It means: (1) 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. (2) 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.

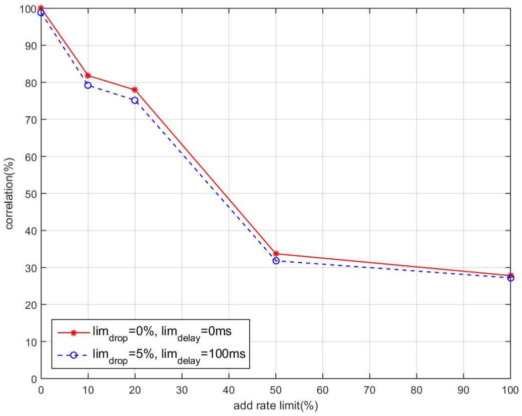


Figure 5: Limit on Adding Dummy Packets (lim_{add})

Figure 6 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%.

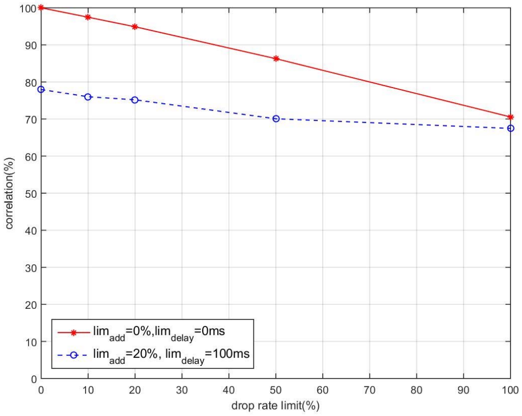


Figure 6: Limit on Packet Drop Rate (lim_{drop})

Figure 7 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.

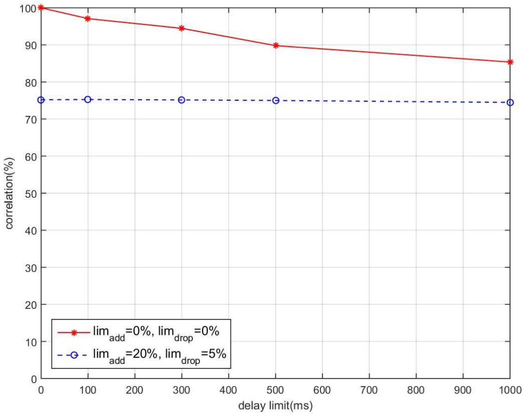


Figure 7: Limit on Packet Delay lim_{delay}

6.4. 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 8 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 $lim_{add} \geq 20\%$.

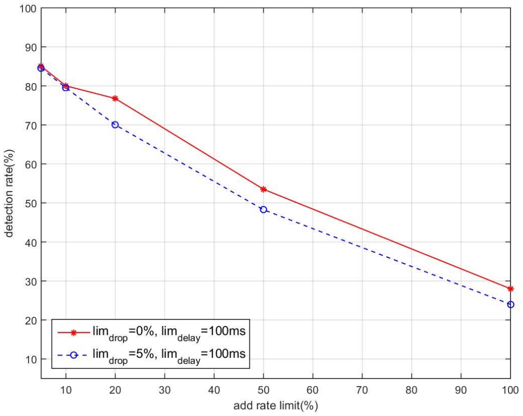


Figure 8: Different Application Performance

7. Discussion

In this section 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.

In this paper, 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.

8. Conclusion and Future Work

In this paper, 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 experiments show the hiding approach can effectively hide traffic patterns and resist replay attacks to identify the same speech.

9. Acknowledgement

This work was supported in part by the National Science Foundation under Grants CNS-1338105, CNS-1343141, and CNS-1527579.

Appendix
Solution of the Optimization Problem

The objective function (2) can be simplified as follows:

$$D(y_j^{o,s}) = \frac{a y_j^{o,s} + b}{\sqrt{c y_j^{o,s2} + d y_j^{o,s} + e}} \tag{3}$$

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

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

[3] 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)*, May 2008, pp. 35–49.

[4] C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson, "Language identification of encrypted voip traffic: Alejandra y roberto or alice and bob?" in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, ser. SS'07. Berkeley, CA, USA: USENIX Association, 2007, pp. 4:1–4:12. [Online]. Available: http://dl.acm.org/citation.cfm?id=1362903.1362907

[5] Y. Zhu and H. Fu, "Traffic analysis attacks on skype voip calls," *Comput. Commun.*, vol. 34, no. 10, pp. 1202–1212, Jul. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2010.12.007

[6] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proc. of the 13th USENIX Security Symposium*, San Diego, CA, August 2004, pp. 303–320.

[7] Y. Guan, X. Fu, D. Xuan, P. U. Shenoy, R. Bettati, and W. Zhao, "Netcamo: camouflaging network traffic for qos-guaranteed mission critical applications," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 31, no. 4, pp. 253–265, Jul 2001.

[8] cnn.com, "Police reveal the identity of shooting suspect," available: http://www.cnn.com/2006/US/09/29/school.shooting/index.html.

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

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

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

[12] D. W. Patterson, *Artificial Neural Networks: Theory and Applications*, ser. Prentice-Hall Series in Advanced Communications. Prentice Hall, 1996. [Online]. Available: https://books.google.com/books?id=tJokAQAAIAAJ

[13] F. Fessant, S. Bengio, and D. Collobert, "On the prediction of solar activity using different neural network models," *Annales Geophysicae*, vol. 14, no. 1, pp. 20–26. [Online]. Available: http://dx.doi.org/10.1007/s00585-996-0020-z

[14] S. Haykin, *Neural Networks: A Comprehensive Foundation*, ser. International edition. Prentice Hall, 1999. [Online]. Available: https://books.google.com/books?id=bX4pAQAAMAAJ

[15] X. Gong, N. Borisov, N. Kiyavash, and N. Schear, *Privacy Enhancing Technologies: 12th International Symposium, PETS 2012, Vigo, Spain, July 11-13, 2012. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ch. Website Detection Using Remote Traffic Analysis, pp. 58–78. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-31680-7_4

[16] X. Luo, E. W. W. Chan, and R. K. C. Chang, "Detecting pulsing denial-of-service attacks with nondeterministic attack intervals," *EURASIP J. Adv. Signal Process*, vol. 2009, pp. 8:1–8:13, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1155/2009/256821