Int. J. Inf. Secur. (2010) 9:313–325
DOI 10.1007/s10207-010-0111-4

REGULAR CONTRIBUTION

Analysis of information leakage from encrypted Skype conversations

Benoît Dupasquier · Stefan Burschka · Kieran McLaughlin · Sakir Sezer

Published online: 24 July 2010

Abstract Voice over IP (VoIP) has experienced a tremendous growth over the last few years and is now widely used among the population and for business purposes. The security of such VoIP systems is often assumed, creating a false sense of privacy. This paper investigates in detail the leakage of information from Skype, a widely used and protected VoIP application. Experiments have shown that isolated phonemes can be classified and given sentences identified. By using the dynamic time warping (DTW) algorithm, frequently used in speech processing, an accuracy of 60% can be reached. The results can be further improved by choosing specific training data and reach an accuracy of 83% under specific conditions. The initial results being speaker dependent, an approach involving the Kalman filter is proposed to extract the kernel of all training signals.

Keywords Information security · Privacy · Skype · Voice over IP (VoIP)

1 Introduction

In the last few years, the evolution of voice over IP (VoIP) has been tremendous. The fact that it is free of charge, easy to use and apparently secure has promoted this technology among the population. Quickly, however, groups, with illegal or criminal intent, realized the potential of using VoIP as a covert channel. This has led several federal agencies, such as the National Security Agency (NSA) and the German Federal Police, and recently the European Union, to demand the ability to eavesdrop on VoIP conversations. However, several service providers reject these demands without a legal warrant, to preserve privacy rights. Therefore, alternative methods, such as using domain specific knowledge to determine whether a given person uses sentences like *"I put the bomb in the train"*, have to be found to enable law enforcement to acquire an indication that might help to obtain a warrant for further investigations.

As VoIP becomes increasingly used for professional purposes, the need for encryption of VoIP conversations seems obvious. However, contrary to what most people think, an encrypted VoIP conversation does not completely prevent an attacker from eavesdropping. Using only side channel knowledge such as packet sizes or inter-arrival times, this paper demonstrates the false sense of privacy provided by Skype. To ensure the validity of the proposed results, more than 6 GB of Skype traces have been collected as training and test material. Starting from the classification of phonemes, this paper investigates how an eavesdropper can determine the content of an encrypted VoIP conversation. This goal is achieved by identifying particular sentences in the flow of IP packets using algorithms from speech processing, such as dynamic time warping (DTW), and from radar tracking analysis, such as the Kalman filter. DTW is an algorithm used to compare two series. By stretching the series, i.e., mapping several elements of one series to a single element of the other series, the algorithm is able to deal with time or speed variations. The Kalman filter is used to reduce the noise present in a set of measurements and to predict the real value that would be measured in an ideal, noise-free environment. Finally, DTW and the Kalman filter algorithms, along with their assumptions, are presented in depth in Sects. 2.3.1 and 2.3.2, respectively.

This paper is organized as follows: some related work concerning VoIP security and encryption is briefly presented

B. Dupasquier (✉) · S. Burschka · K. McLaughlin · S. Sezer
Centre for Secure Information Technologies (CSIT),
Queen's University of Belfast, Belfast, Northern Ireland, UK
e-mail: bdupasquier01@qub.ac.uk

in Sect. 1.1. The motivation is explained in Sect. 1.2. The methodology is presented in Sect. 2, including the experimental setup and the requisite background for understanding the algorithms proposed. The results are discussed in Sect. 3. Finally, the conclusion is presented in Sect. 4.

1.1 Related work

Literature shows that privacy is not yet fully guaranteed with VoIP, as illustrated by number of published papers, ranging from inferring speech activity to identifying sentences or the language of the conversation.

A traditional speaker identification system usually has to decompress the compressed stream before being able to analyze it. Aggarwal et al. [2] show that decompression can be avoided. They propose to lower the CPU and memory requirements while still having a high accuracy of speaker identification and also an accuracy comparable to the one obtained when decompressing the stream. This challenging goal is achieved by using a so-called *micro-clustering algorithm*, detailed in [1]. They reach an accuracy rate of 80% with a three times higher speed than the Gaussian mixture model (GMM), used in the traditional approach. Moreover, they are able to recognize the speaker in real time after five seconds of seeing the first packet in a stream.

In order to maintain user datagram protocol (UDP) bindings at the network address translation (NAT) and to obtain better voice quality, Skype does not suspend sending packets during silence. This means that silence is encoded, encrypted and sent over the network, therefore complicating the task of inferring speech activity, a process commonly called voice activity detection (VAD). Nevertheless, Chang et al. [4] propose a method they named network-level VAD to infer speech activity from encrypted voice traffic. Their work is based on the fact that speech activity is highly correlated with the size of encrypted packets, i.e., more information is encoded in a voice packet while the user is speaking than while the user is silent. In their paper, they devise an adaptive thresholding algorithm allowing silence to be differentiated from voice activity with an accuracy of 85%.

In their paper *Privacy of Encrypted Voice over IP* that focuses on Google Talk, Lella and Bettati [10] assume that the packet sizes do not carry any information but that all the information is contained in the inter-arrival times. They use the silence phases, which are encoded in smaller packets and with bigger inter-arrival times, to isolate words. Then, they compare the length of the words, namely the timestamp of the last packet belonging to the word minus the timestamp of the first one, against their models. Their first approach, based on a simple Bayesian classifier, is context unaware. Subsequently, they propose a second approach based on hidden Markov models (HMMs) to produce context awareness. A drawback to their approach is that a congested network or a slow or lossy link will render their attack ineffective. Moreover, it relies on human analysis to determine the plausibility of the proposed sentence.

Wright et al. [17] propose to use the size of transmitted packets to determine whether a given phrase is spoken. By reducing speech to phonemes, the most basic speech unit, they are able, using HMMs, to reconstruct sentences using the sequence of packet lengths. By reducing the problem of encrypted VoIP speech recognition to a substring matching problem on the packet sizes, the authors are able to achieve an average accuracy of 50% in telling whether a sentence is contained in a stream of packets. By choosing some phonetically rich sentences, such as *"Young children should avoid exposure to contagious diseases"*, they achieve an accuracy above 90%. By using a similar approach and a variant of the χ^2 test, Wright et al. [18] have also demonstrated that determining the language of an encrypted conversation is possible. Their algorithm achieves an accuracy of 66% in identifying the language of a conversation among a set of 21 models. Furthermore, 14 languages can be identified with an accuracy greater than 90%. Finally, their algorithm achieves an overall binary classification rate of 86.6%. However, a significant limitation to their approach is that it needs the binary of the audio codec used and requires the VoIP application to encode voice using a variable bit rate (VBR) codec and a length preserving encryption mechanism.

To conclude, many experiments have been conducted on this topic that do not take account of the time dependency of the packet signals. The work presented in this paper will now show that it is a factor with information gain even without the availability of the codec.

1.2 Motivation

This paper addresses the challenge of extracting information from encrypted VoIP traffic. Although several studies have already tackled this issue, none of them have tried to attack Skype or other closed-source codecs. This represents a real challenge as the measurement and process noise have to be taken into account. This requires the investigation into preprocessing methods able to remove, attenuate or take the effect of noise into account.

The efficiency of the approach presented in [10] is strongly dependent on the quality of the network. Furthermore, sentences of similar lengths cannot be reliably identified. This paper overcomes this restriction by focusing on packet sizes rather than inter-arrival times. In addition, it is proposed that by using the DTW algorithm presented in Sect. 2.3, variability in time or speed can be efficiently dealt with. Furthermore, unlike HMMs, the DTW algorithm does not require an extensive amount of training data to work properly. This means that a single occurrence of a sentence spoken by a

known speaker might theoretically be sufficient to generate a model for this particular speaker.

Experiments presented in Sect. 3.1 will show that the approach proposed in [17] does not work with Skype traffic as the encoding of phoneme depends strongly on the surrounding sounds. Furthermore, the approaches proposed in [17] and [18] require the possession of a binary of the audio codec to create a match between the VBR output and the encrypted packet length. However, several audio codecs are closed source and do not share their binary. Without a binary, such a mapping could not be created. The non-availability of the codec renders the attack more difficult, and this clearly creates an issue that cannot be solved by the approach proposed in [17] or [18]. Indeed, to investigate unknown or closed-source codecs, methods that eliminate the influence of the encryption algorithm have to be developed. The strength of the methodology presented in this paper compared to the referenced techniques is that it can be applied to any VoIP application, regardless of the availability of the codec's binary.

2 Methodology

All experiments are conducted on Skype version 4.0.0.224 working in direct UDP mode. The laboratory setup is composed of two Microsoft Windows XP workstations (Fig. 1). Some prerecorded speech is played on one side, while all the traffic between the two stations is captured using *WinDump* [16].

To facilitate the analysis process, Skype is operated in direct UDP mode and in a noise-free environment. Furthermore, synthetic speech, being close to the nature of Skype's vocoder, is being used. Finally, several recordings of the model are created. By observing Internet Protocol (IP) traces of Skype conversations, it has been determined that information is carried in the time-dependent flow of packet lengths. The number of packets and the amount of bytes transferred have been identified as features relating to a given sentence. Perceiving the sequence of packets as a signal inspired the application of methods from speech processing or radar signal analysis.

An extensive analysis of the packet sizes used by encrypted Skype traffic revealed that a given sentence is always encoded

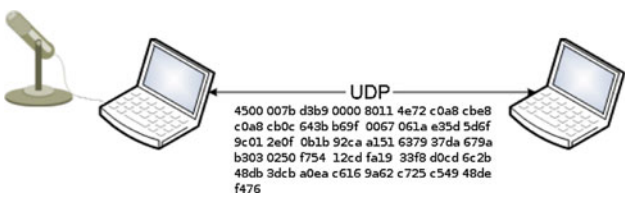

Fig. 1 Laboratory setup

in a similar way, and conversely, different sentences produce different outputs. Figure 2 shows two instances of the same sentence overlaid. The outputs produced are almost identical. Figure 3 shows the output produced by two very different sentences, which clearly produce two very different outputs. This is an important observation as it implies that, in order to determine the most probable model, an algorithm able to compare sequences of packets is sufficient. The DTW algorithm has been chosen for this purpose. The results obtained are presented in the following section.

In order to avoid the generation of several speaker models per sentence, the Kalman filter is used toward the generation of speaker-independent models. To begin with, the first observation is chosen as the initial estimator of the state of the system, and then, the Kalman filter is recursively applied to the training data. Finally, the DTW algorithm is used to compare the test data against the Kalman's models.

2.1 Hypotheses

The first hypothesis is the existence of distinct transfer functions between the audio input and the observed IP packet lengths. Stated otherwise, it has been presumed that given the input, the output can be predicted, and conversely, given the output, the input can be estimated with a reasonable probability. The next hypothesis is the relation between the application characteristics and the time sequence of the IP packets produced, namely frequency, amplitude, rhythm, noise, silence or phonemes. Those parameters have been chosen among others because speech is dynamic and it is possible to generate most of the sounds produced by the human voice by mixing those parameters.

2.2 Limitations and assumptions

All the experiments have been conducted using the English language, since it is the most common one on the Internet. Nevertheless, this work can be adapted to analyze streams of any other language.

Although it is focused on Skype, this work can be adapted to any other existing VoIP software. Skype works independently of the network topology and security measures. In order to focus on the vital part, the content appraisal from IP packets, this work concentrates first on the ideal case without a firewall or NAT. This also prevents too many unknown variables from influencing the experiments. To simplify the experiments and in order to understand the fundamental principles behind Skype traffic, this work focuses on an idealistic case. First, it is assumed that the direct UDP mode is used. This setup has been chosen because it is the one that induces the least interferences, e.g., delays and losses. However, it is not the most common in the Internet.

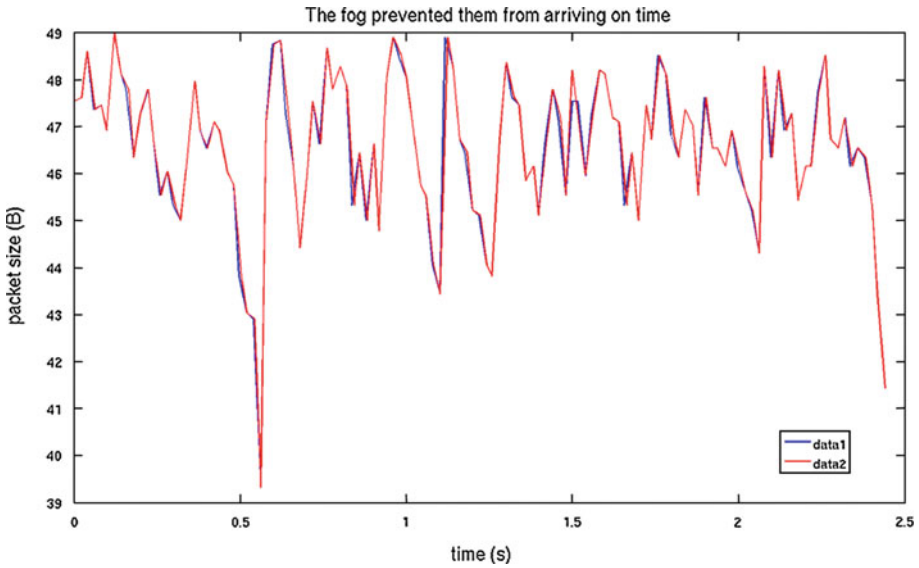

Fig. 2 Same sentences, similar outputs

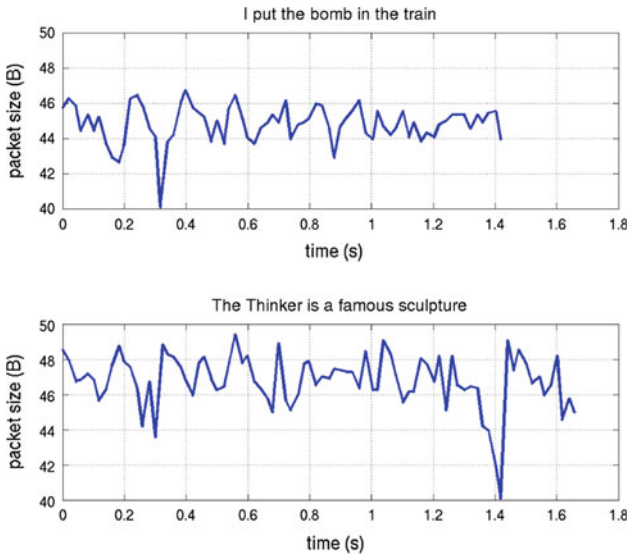

Fig. 3 Different sentences, different outputs

Second, samples of prerecorded synthetic speech are used, which do not vary between the different experiments. In practice, speech is dynamic, and a sentence uttered by the same speaker will not always sound exactly the same way. Third, a conversation is only defined by the audio traces of Skype, i.e., no text messages, file transfer or video embedded.

Since Microsoft Windows XP is one of the most common operating systems (OS), two Skype clients working on Microsoft Windows XP are used for the experiments. This choice is lead by the desire to work on the latest technologies offered by Skype, mainly its new audio codec SILK and to provide results valid for as many users as possible.

2.3 Background

In this section, the algorithms relevant to the present attack method are briefly presented. The two key algorithms used are dynamic time warping (DTW) and the Kalman filter. Several references are provided for anyone requiring a deeper understanding of those algorithms.

2.3.1 Dynamic time warping (DTW)

DTW is a dynamic programming algorithm allowing sequences, possibly of different length, to be compared. Therefore, DTW is an algorithm suitable for comparing two sequences that may vary in time or speed, as illustrated by Fig. 4, giving an insight into how time series of different lengths are handled. The abscissa represents the time t, and the ordinate indicates the value observed at time t. Albeit both time series have similar y-values, they have been separated to highlight the warping operations of the algorithm. A typical warping example can be observed in the beginning of the first time series, where a single point is matched against three points in the second time series. Similar behavior can be observed in the middle and at the end of the series. DTW has been extensively used, especially in speech processing before being replaced by HMMs.

DTW assumes that the time interval between all occurrences is constant. Although this is not the case for Skype, it has been shown that there is a correlation between the inter-arrival times and the packet sizes, namely, the smaller the inter-arrival time, the smaller the packet size and the bigger the number of packets. Moreover, the evolution of the inter-arrival times is constant and repeated over time and,

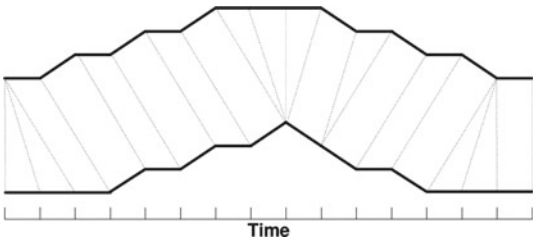

Fig. 4 DTW dealing with time variations (Source: [11])

as only voice packets are analyzed, no bias that could alter the behavior of DTW is induced.

Algorithm

DTW algorithm's first step is to compute the so-called local cost matrix. It requires a distance or cost function. In this paper, the squared Euclidean distance function has been chosen.

The **squared Euclidean** distance between two points $P = (x_i)$ and $Q = (y_j)$ is computed as follows:

$$dist_{\text{eucl}}(P, Q) = \left(\sqrt{(x_i - y_j)^2}\right)^2 = (x_i - y_j)^2$$

The local cost matrix C is a matrix of size $m \times n$, where the element $C(i, j)$ is defined as

$$C(i, j) = dist(x_i, y_j)$$

where $dist$ is the distance or cost function.

After having computed the local cost matrix, the accumulated cost matrix DW, denoting the minimal distance between the two sequences, must be computed. This matrix is computed as follows:

1. Initialize the starting point (initial condition):

$$DW(1, 1) = 0$$

2. Compute the first column:

$$DW(i, 1) = DW(i - 1, 1) + C(i, 1)$$

3. Compute the first line:

$$DW(1, j) = DW(1, j - 1) + C(1, j)$$

4. Compute the rest of the matrix:

$$DW(i, j) = C(i, j) + min\{DW(i - 1, j), DW(i - 1, j - 1), DW(i, j - 1)\},$$

The three last steps depend on the conditions imposed on the DTW path through the matrix. In this example, only transitions to the point (i, j) from the points $(i - 1, j)$, $(i - 1, j - 1)$ or $(i, j - 1)$ are allowed.

Given the matrix DW, the minimal distance DTW is simply given by the element $DW(m, n)$, yielding the following equation:

$$DTW = DW(m, n)$$

Finally, the optimal path through the matrix needs to be backtracked. Intuitively, the algorithm to perform this task is very simple. Starting from the end, $DW(m, n)$, the allowed cell with the least cost is chosen each time, until $DW(1, 1)$ is finally reached. More formally, the algorithm is presented in Algorithm 1.

Algorithm 1 Optimal warping path

```
1: i ← m
2: j ← n
3: while i > 1 and j > 1 do
4:   if i = 1 then
5:     j ← j − 1
6:   else if j = 1 then
7:     i ← i − 1
8:   else
9:     minVal ← min{DW(i − 1, j),
                    DW(i − 1, j − 1),
                    DW(i, j − 1)}
10:    if minVal = DW(i − 1, j) then
11:      i ← i − 1
12:    else if minVal = DW(i, j − 1) then
13:      j ← j − 1
14:    else
15:      i ← i − 1
16:      j ← j − 1
17:    end if
18:    add (i, j) to the optimal path
19:  end if
20: end while
```

In order to compare the calculated minimal distance with those obtained from other experiments, it has to be normalized in order to make the comparison meaningful. For this purpose, the distance has to be divided by a normalization factor N, which is the length of the optimal warping path.

Having the normalization factor, computing the normalized distance $NDTW$ is trivial:

$$NDTW = \frac{DTW}{N}$$

Obviously, the smaller the distance, the more similar the input sequences are. Note that computing the DTW between X and X reveals a distance of 0.

Interpretation

The result of the DTW algorithm comes in many forms. The first one is a value representing the distance, i.e., the correlation factor, between both sequences. The closer to zero this distance is, the more similar the input sequences are. It also supplies a graphical representation, displaying a matrix of size $m \times n$, where the abscissa represents one sequence and the ordinate axis the other one. Each cell (i, j) of the matrix is colored according to the distance $dist(x_i, y_j)$ that will be introduced later, along with the DTW algorithm. The "colder" the color, the smaller the distance.

Figure 5 presents a matching and a non-matching DTW path. The optimal DTW path is marked by a white line. A perfect matching results in a straight line from the upper left of the matrix to the bottom right, namely the point (m, n).

The two sequences used for this example are $a = (1, 2, 3, 4, 5, 6)$ and $b = (9, 2, 13, 4, 2, 1, 6)$. The matching path is the path between a and a, with a normalized distance, i.e., the distance divided by the length of the path, of 0, while the non-matching path is the path between a and b, with a normalized distance of 17.66.

Restrictions

First, the alignment sequence, that is the path through the grid, has to be monotonic in the time index, which means that it is not possible to go back in time. This restriction guarantees that a given feature is not repeated. Second, the alignment path has to be continuous, which means there cannot be holes in the time process. The last condition specifies where the process starts, namely at the origin $(0, 0)$, and stops, namely at the point (m, n).

Complexity

The main drawback of DTW is its complexity, namely $\mathcal{O}(mn)$, where m and n are the length of the input sequences.

More details on the DTW algorithm can be found in [9,11] and [5].

2.3.2 Kalman filter

The Kalman filter is a recursive linear filter that estimates the state of a linear, discrete-time dynamical system from a series of noisy measurements. It is mainly used for radar or missile tracking problems but is also widely used in computer vision, economics and navigation (aerospace, land and marine).

The Kalman filter combines all available measurement data, plus prior knowledge about a system and measuring devices and can be used to filter, predict or smooth a discrete-time linear process. Moreover, it can be shown that the Kalman filter is the optimal linear filter, namely the one that most minimizes the variance of the estimation error.

Basically, the goal of the algorithm is, given the current state of the system, to predict the next state and its uncertainty. Then, the prediction is corrected with the new measured values. Figure 6 proposes an application of the Kalman filter to estimate the position of a plane. At time t_1, Bob announces that the plane has been seen at position x_1. Later on, Alice reports that the plane has just been seen at position x_2. Finally, based on the reports from Alice and Bob, the Kalman filter will predict the position of the plane x_3 at time t_3.

Assumptions

First, it must be possible to describe the process that is measured in terms of a linear system. Recall that a linear system

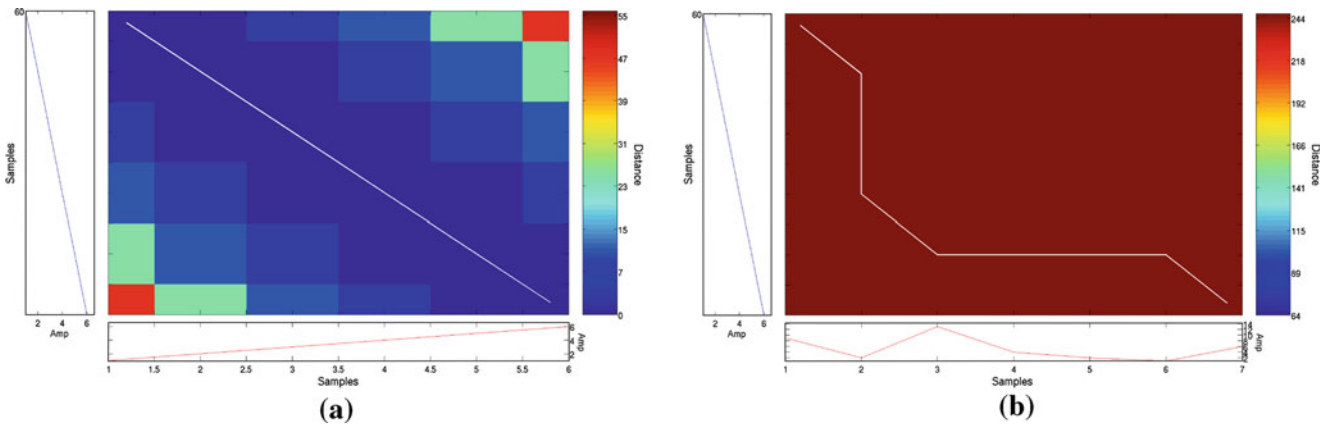

Fig. 5 Matching and non-matching DTW path. **a** Matching DTW path. **b** Non-matching DTW path

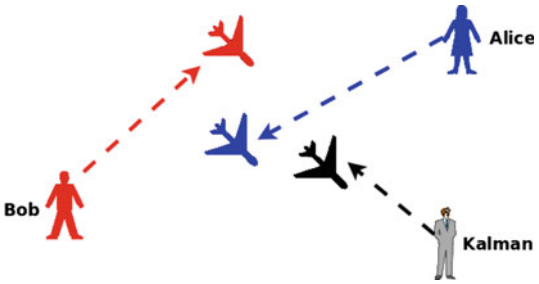

Fig. 6 Kalman estimating the position of a plane

is a process than can be described by the following two equations:

(State equation)

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1}$$

(Output equation)

$$y_k = H_k x_k + v_k$$

where

F = State transition matrix
G = Input matrix
H = Observation matrix
k = Time index
x = State of the system
u = Known input to the system
y = Measured output
w = Process noise
v = Measurement noise

Second, the Kalman filter assumes that both the process and the measurement noise are additive, white and Gaussian, with zero mean, and with covariance S_w and S_v, respectively. Moreover, the process noise w is uncorrelated with the measurement noise v, i.e., at any time k, w_k and v_k are independent random variables. These assumptions are summarized in the following equations:

$$w_k \sim N(0, Q_k)\ v_k \sim N(0, R_k)$$

$$S_w = E\left[w_k w_j^T\right] = \begin{cases} Q_k & \text{if } k = j \\ 0 & \text{if } k \neq j \end{cases}$$

$$S_v = E\left[v_k v_j^T\right] = \begin{cases} R_k & \text{if } k = j \\ 0 & \text{if } k \neq j \end{cases}$$

Algorithm

Given a dynamic system governed by the following equations:

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1}$$
$$y_k = H_k x_k + v_k$$
$$E\left[w_k w_j^T\right] = Q_k \delta_{k-j}$$
$$E\left[v_k v_j^T\right] = R_k \delta_{k-j}$$
$$E\left[w_k v_j^T\right] = 0$$

where

$\hat{x}_k^-$ = Estimate of x_k before the measurement at time k has been processed
$\hat{x}_k^+$ = Estimate of x_k after the measurement at time k has been processed
P_k^- = Covariance of the estimation error of $\hat{x}_k^-$
P_k^+ = Covariance of the estimation error of $\hat{x}_k^+$
K_k = Kalman gain factor
I = Identity matrix
M^T = Transpose of the matrix M
M^{-1} = Inverse of the matrix M

and δ_{k-j} is the Kronecker's delta function, namely,

$$\delta_{k-j} = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k \neq j \end{cases}$$

then, the discrete-time Kalman filter is initialized as follows:

$$\hat{x}_0^+ = E\left[x_0\right]$$
$$P_0^+ = E\left[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T\right]$$

If the initial state is perfectly known, then $P_0^+ = 0$. Otherwise, if the value of x_0 is completely unknown, then $\hat{x}_0^+ = 0$ and $P_0^+ = I$.

Finally, the Kalman filter is given by the following equations, which are computed for each time step $k = 1, 2, \ldots, n$ and can be divided in two phases, namely a prediction step and a correction step:

– Time update equations (prediction step)

$$\hat{x}_k^- = F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1}$$
$$P_k^- = F_{k-1}P_{k-1}^+ F_{k-1}^T + Q_{k-1}$$

– Measurement update equations (correction step)

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$
$$= P_k^+ H_k^T R_k^{-1}$$
$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-)$$
$$P_k^+ = (I - K_k H_k)P_k^-(I - K_k H_k)^T + K_k R_k K_k^T$$
$$= \left[(P_k^-)^{-1} + H_k^T R_k^{-1} H_k\right]^{-1}$$
$$= (I - K_k H_k)P_k^-$$

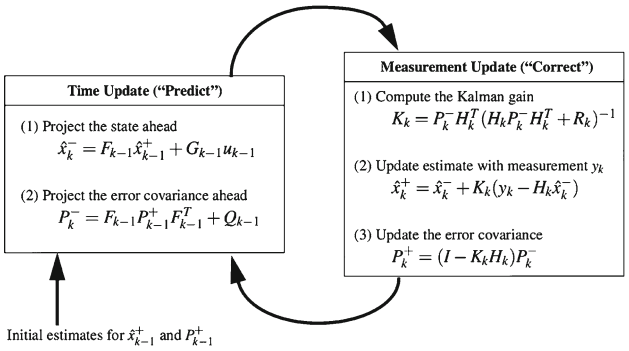

Fig. 7 The Kalman filter algorithm (Source: [15])

The error between the true state and the estimated state is denoted as

$$\tilde{x}_k = x_k - \hat{x}_k$$

To conclude, a graphical representation summarizing the operations of the Kalman filter is proposed in Fig. 7.

Simplifications

Most of the time, it can be assumed that F_k, H_k, Q_k and R_k are time independent. The subscript k for these matrices can thus be dropped. Moreover, in the case presented in this paper, no control on the input is assumed, and therefore, $u_k = 0 \, \forall k$. As the measurement is comprised of the state value and some noise, it can be assumed that $H = 1$. In practice, it is uncommon to encounter real-life cases where H is different from 1. It has been assumed that the initial values are not known, and therefore, $x_0 = 0$ and $P_0 = I$ have been chosen. Note that P_0 must be different from 0, otherwise, this would mean that there is no noise in the environment and would lead all consequent values of $\hat{x}_k$ to be zero.

Finally, the following equations complete the description of the linear system:

$$\begin{aligned} x_k &= Fx_{k-1} + w_{k-1} \\ y_k &= x_k + v_k \\ E\left[w_k w_j^T\right] &= Q\delta_{k-j} \\ E\left[v_k v_j^T\right] &= R\delta_{k-j} \\ E\left[w_k v_j^T\right] &= 0 \end{aligned}$$

and the Kalman filter is initialized as follows:

$$\begin{aligned} \hat{x}_0^+ &= 0 \\ P_0^+ &= I \end{aligned}$$

The time and measurement update equations become:

- Time update equations (prediction step)

$$\begin{aligned} \hat{x}_k^- &= F\hat{x}_{k-1}^+ \\ P_k^- &= FP_{k-1}^+F^T + Q \end{aligned}$$

- Measurement update equations (correction step)

$$\begin{aligned} K_k &= P_k^-(P_k^- + R_k)^{-1} \\ &= P_k^+R^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - \hat{x}_k^-) \\ P_k^+ &= (I - K_k)P_k^- \end{aligned}$$

Details on the original proposals for using Kalman filters can be found in [7] and [8], while more recent work and development can be found in [3,12–15] and [6].

3 Results

As mentioned in the previous section, a first encouraging observation is the coherence and reproducibility of the experiments. By analyzing the sequence of IP packets generated for an experiment run several times under the same conditions, only minor changes can be observed. This proves that Skype does not generate random packet sizes as the output appears to be deterministic. This observation, however, does not give any indication about eventual padding or the way Skype generates packets.

As mentioned in Sect. 2.1, all the sounds the human voice can produce can be generated by combining amplitude, frequency, silence and noise. Since noise is omnipresent in VoIP, its effect on Skype is examined first. Figure 8 shows the output generated by Skype when excited with various types of noise, namely brown, pink and white. Recall that Brown noise (also referred to as Brownian, red or random walk noise) has a spectral density proportional to $\frac{1}{f^2}$, pink or $\frac{1}{f}$ noise has a spectral density proportional to $\frac{1}{f}$ and white noise is evenly distributed at all frequencies. As illustrated, Skype's audio codec "learns" that this is noise, implying that SILK uses an adaptive codebook. Another expected behavior is that brown noise is encoded in smaller packets than pink and white noise. Nonetheless, the observed difference is not very obvious, and no further interesting information can be extracted from this set of experiments. Due to its highly random nature, white noise exhibits higher fluctuations between the different experiments than brown and pink noise.

Additional experiments further showed that neither the signal amplitude nor its frequency has a correlated influence on the IP signal when basic signals are present. Indeed, as the results presented so far have shown, Skype does not

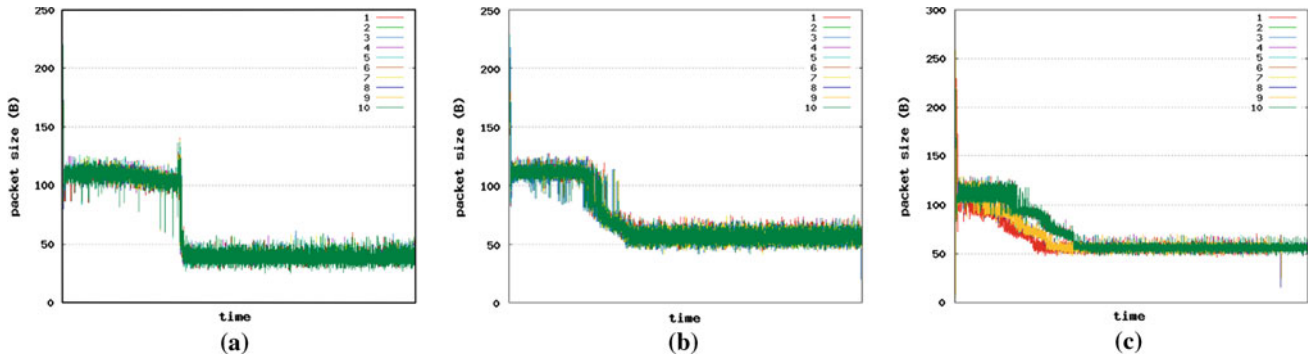

Fig. 8 Output generated by Skype when excited with various types of noise. **a** Brown noise. **b** Pink noise. **c** White noise

have a simple output response to "basic" input signals. It can thus be deduced that Skype does not encode the frequency nor the amplitude directly. The fast Fourier transform (FFT) approach has also been explored. However, transforming the time series into the frequency spectrum reveals no usable information. Therefore, this suggests there is no significant correlation between encrypted sentences and amplitude of the packet size frequencies in the spectral view. This result might be due to the presence of noise or because of the special assumptions that FFT makes, namely the continuity of the transformed signal. Mirroring the signal, to avoid this assumption, does not change the results. These results can be explained by the characteristics of the vocoding process in Skype which also prevented conventional frequency analysis. Therefore, an alternative approach has been explored, where Skype's response when excited with phonemes has been analyzed.

3.1 Phoneme classification

Recall that a phoneme is the smallest linguistically distinctive unit of sound. It is also important to note that it does not carry semantic content in itself, e.g., a word is a concatenation of phonemes.

As phonemes have an expected duration greater than 60 ms, the maximal inter-arrival time value observed, it can be expected that a phoneme will be encoded using several packets, as confirmed by Fig. 9. Therefore, a phoneme cannot be matched to a particular size but can be matched against a sequence of packet sizes or a sequence of "features" that will be discussed shortly.

By carefully examining the generated graphs of Figs. 10, 11, 12 and 13, some invariants in the coding of phonemes can be observed, namely a "start" phase, a "comb" phase and a "decay" phase. The comb phase produces characteristic fluctuations in the packet length. All the phases appeared to be different for the phonemes tested in the experiments. Therefore, these phases seem to be features that are characteristic of a given phoneme. Furthermore, the length of the signal seems to be preserved; therefore, a long vowel can easily be distinguished from a short one (Figs. 10 and 11).

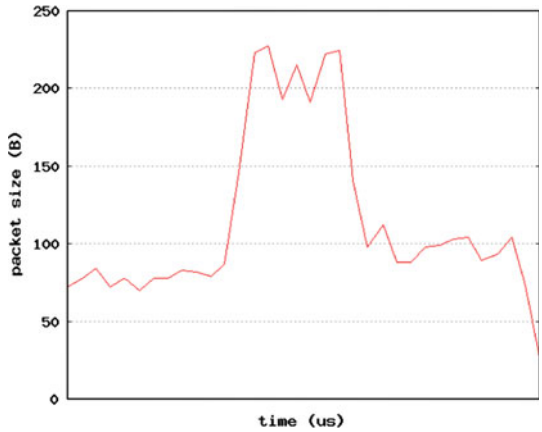

Fig. 9 Encoding of the phoneme /ʒ/, e.g., found in the word pleasure

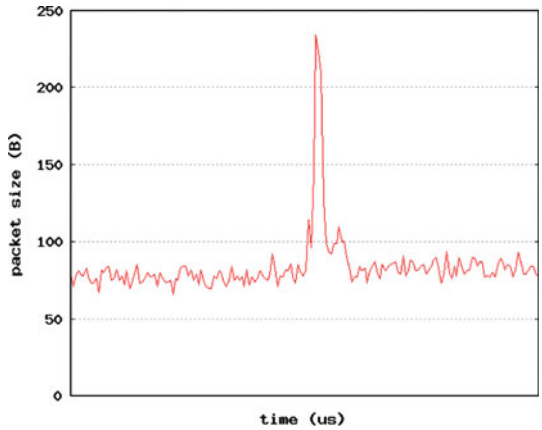

Fig. 10 Encoding of the short vowel /ə/

It can also be noted that voiced consonants can be derived from voiceless consonants by applying a low-pass filter (Figs. 12 and 13). Recall that voiceless consonants, e.g., /p/, /t/, /k/, /f/ and /s/, do not require the vocal cords to vibrate, while voiced consonants, e.g., /b/, /d/, /g/, /v/ and /z/, do.

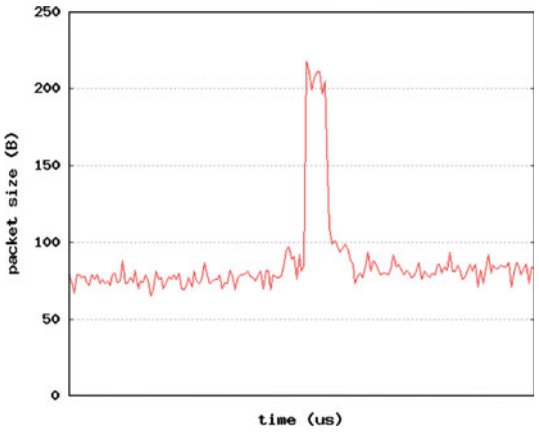

Fig. 11 Encoding of the long vowel */i:/*

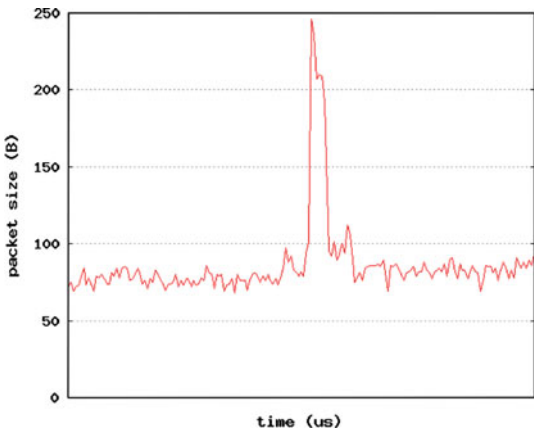

Fig. 12 Encoding of the voiceless consonant */p/*

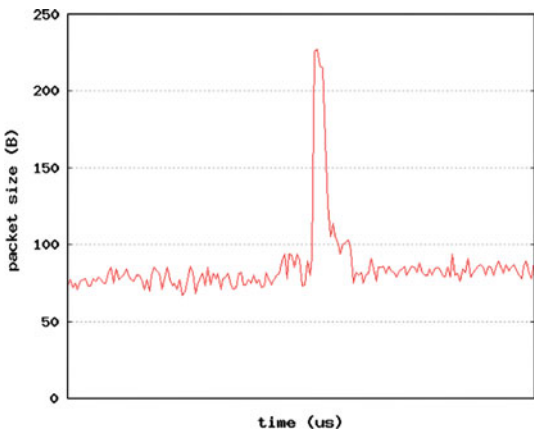

Fig. 13 Encoding of the voiced consonant */b/*

It is rare to find a single isolated phoneme in normal speech activity. Indeed, most of the time, diphones, triphones or even longer series of phonemes are encountered, e.g., words are composed of many phonemes. Therefore, it must be investigated whether invariants can still be detected when coding words, i.e., series of phonemes, or sentences. For ease of detection, a word would ideally be encoded the same way as its phoneme sequence, e.g., */kæt/*, for the word *cat*. Yet, it is unlikely to be the case as the encoding of a phoneme depends on its surrounding sounds. Logically, it might be expected to find the beginning of the first phoneme, an overlapping of the first and second phoneme and finally, the end of the third phoneme.

The experimental results show that this is not the case. As expected, phonemes do overlap. However, contrary to what had been suggested, the start phase does not correspond to the start phase of the first phoneme. The same holds for the slow-down phase, which do not match the end phase of the last phoneme. Therefore, it seems quite difficult to map words by simply concatenating their series of phonemes as done by [17].

As the approach of trying to encode pairs or triplets of phonemes (or, semantically spoken, diphones or triphones) does not yield conclusive results, the problem of mapping words or sentences to sequences of packet sizes or other features will be approached.

3.2 Sentence recognition using DTW

In order to evaluate sentences, reference models are required for comparison with the recordings under examination. Experiments have shown that the inter-arrival time evolves from 60 to 40 ms to finally reach 20 ms. This behavior, illustrated in Fig. 14, appears to be due to the adaptive nature of the codec learning to differentiate silence and noise from speech. Therefore, the models are generated by a WAVE file consisting of three occurrences of a sentence separated by 30 seconds of silence, so sentences can easily be extracted from the training data.

Then, labeled data is generated, namely several speech samples are recorded in which the given sentence appears in different positions. By manually extracting the given

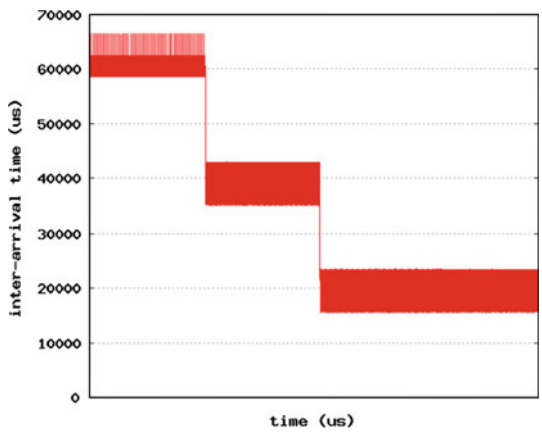

Fig. 14 Evolution of the inter-arrival time

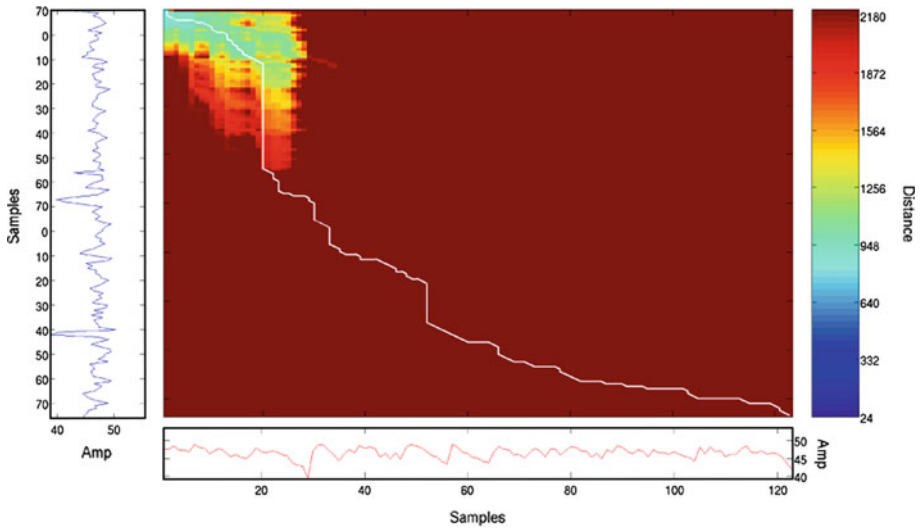

Fig. 15 A non-matching DTW path

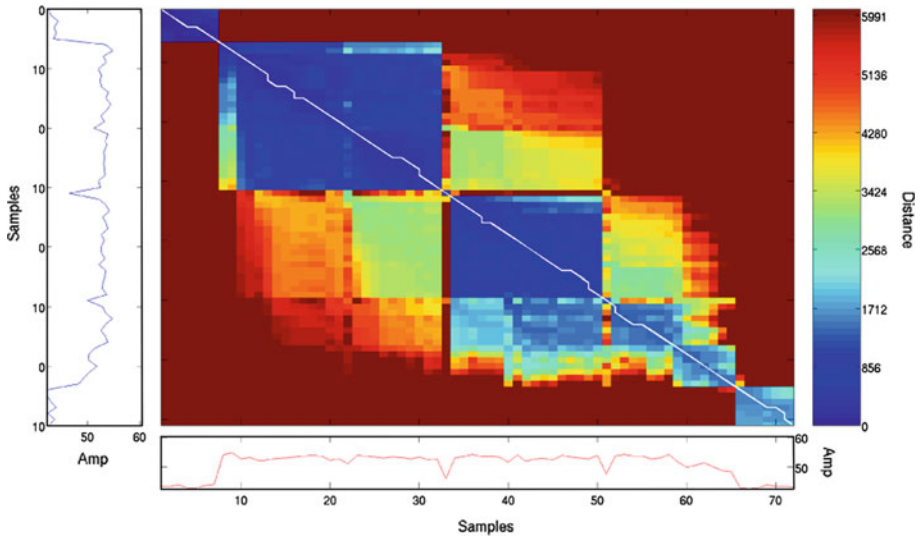

Fig. 16 A matching DTW path

sentence from the traces, the DTW algorithm, presented in Sect. 2.3.1, can successfully be used to detect the sentence. Figures 15 and 16 show the graphic results of the DTW experiment for a non-matching sentence and a matching sentence, respectively. Recall that, as explained in Sect. 2.3, a perfect match would result in a straight line through the diagonal. The smaller graphs on the left and the bottom of each DTW graph represent the input sequences, respectively, the model and the test data.

In order to evaluate the efficiency of the content appraisal attack proposed, several models are generated for sentences chosen to illustrate the ability to detect and distinguish even similar sentences from encrypted Skype traces. Variations of the sentences *"I put the bomb in the plane"* were used, as they present an obvious interest to law enforcement. The other sentences have been chosen for their length or the similarity of the content to this initial sentence. Finally, the following test set is used as an input to the sentence detector:

1. Although always alone we survive,
2. Biblical scholars argue history,
3. I put the bomb in the bus,
4. I put the bomb in the plane,
5. I put the bomb in the tower,
6. I put the bomb in the train,
7. The bomb is in the train,
8. The Thinker is a famous sculpture,
9. There is a bomb in the train.

The test set is composed of 6 experiments, namely of all the possible permutations of sentences 1, 2 and 6. This allows

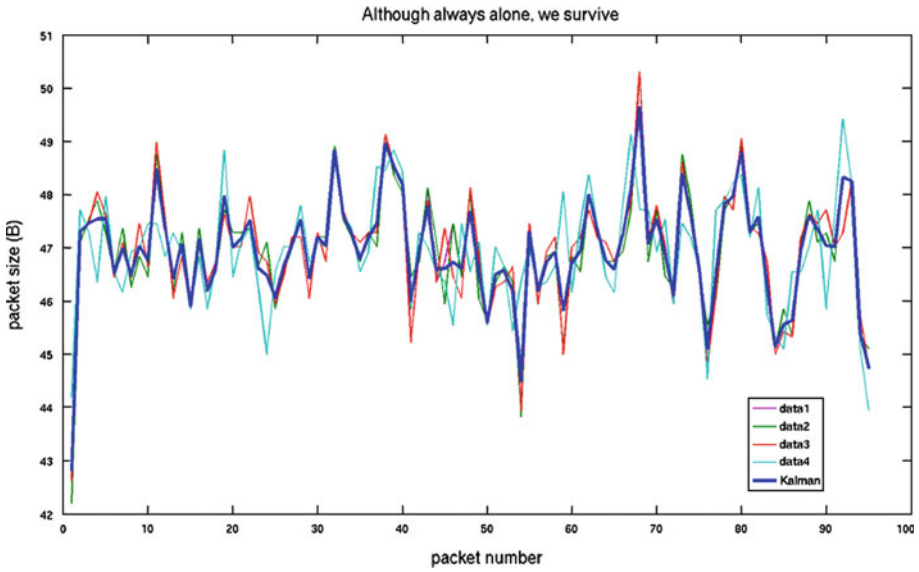

Fig. 17 Model generation using the Kalman filter

the testing of the models against 18 samples (namely 3 sentences * 3! permutations). By creating models and extracting sentences manually, two sentences out of three, i.e., 66%, have been correctly identified. If model 3 is removed, then an accuracy above 70% can be reached. It is also interesting to note that the results could have been improved by using different models and test data. For example, in the case (1, 2, 6), 6 was identified as 3. If model 3 is removed, the sentence is correctly identified. This can be explained by the similarity of the sentences 3 and 6, differing only in the last word. Furthermore, the DTW algorithm has also been tested against sentences longer than the model, and as expected, each time the resulting distance was too big to be accepted as a match. This means that the recall and precision of the method can easily be improved by generating models for sentences that differ in duration.

A given sentence is not always encoded exactly the same way; therefore, by generating several models per sentence, an 83% correct identification rate has been achieved. However, the need to generate several models per sentence is not practical as the number of different models is potentially unbounded due to the variability of speech and noise. Therefore, in the next section, the use of the Kalman filter is proposed in order to extract the kernel of all signals.

3.3 Speaker-independent models using the Kalman filter

The effect of the Kalman filter, introduced in Sect. 2.3.2, on four occurrences of the sentence *"Although always alone, we survive"* is illustrated in Fig. 17. Although not obvious in the figure, a problem faced with this approach is that the training and test data do not always have the same length. This is particularly true between different speakers. Indeed, the model will be as long as the longest sample in the training set. Depending on the difference in size between the sample and the model to be compared against, there is a chance of misclassifying the sentence. Therefore, a slightly different approach is taken. The idea is not to compare the model against the sample, but rather to measure the distortion of the model when updated with the sample.

Let m be the model generated by the Kalman filter and s the test data to compare against the model. Then, the Kalman model is updated with the test data: $k' = Kalman(m, s)$. This means that m is the model and k' is the model corrected with the test data. Now, the distance between m and k' is computed with DTW, as it was done before without Kalman.

Matching sentences should not distort the model too much, while non-matching sentences should induce more variations. However, the problem observed with this approach is that the Kalman filter does not weight the sample enough. Therefore, no matter how different the test data and the model are, the distortion was minimal and the DTW algorithm inefficient. To approach this problem, the input parameters, such as the covariance of the estimation error, were adjusted. The values were selected in the interval of (0, 1], but none of them produced satisfying results. This can be explained by the assumptions inherent to the Kalman filter, which assumes that both the process and the measurement noise are additive, white and Gaussian with zero mean. In the case of Skype, the measurement and process noise are unknown. Therefore, the assumptions made by the Kalman filter might lack the precision necessary to describe the actual noise faced by Skype.

The primary objective of the presented research was the demonstration (proof of concept) that significant information is leaked from Skype conversations. Therefore, further optimization of the above approach is considered as future

research on speaker-independent speech recognition. Albeit there is a limited amount of testing and a small set of targeted sentences, this research work highlights the fact that extraction of information from encrypted VoIP conversations can be achieved without relying on the availability of public information on the codec, encryption, etc. to attempt a successful side channel attack.

4 Conclusions

This paper has demonstrated the false sense of privacy provided by Skype, a widely used VoIP application, known for its strong security policy. Through a comprehensive analysis of encrypted Skype traffic, it has been shown that isolated phonemes can be classified and given sentences identified with an accuracy greater than 60%. An accuracy of 83% can be reached under specific conditions. This challenging goal is achieved with the help of the DTW algorithm, commonly used in the speech processing community. The algorithm is successfully applied on vectors of packet sizes, allowing the probability of a recording being a particular sentence to be determined. Furthermore, the Kalman filter, traditionally used for radar tracking problems, has been proposed to extract the kernel of the training data in order to generate speaker independent models.

To the knowledge of the authors, this is the first published research that demonstrates that Skype encryption is not entirely secure and that information is leaked allowing content to be inferred. In addition, dealing with one of the most challenging VoIP software constructs currently available and proposing a methodology easily adaptable to any other VoIP software means the proposed approach is very versatile. Furthermore, by using the DTW algorithm, the drawbacks of an approach involving HMMs, e.g., an extensive amount of training data, can be avoided. Also, the capacity of DTW to deal with time or speed variability along with the use of packet sizes as discriminant features renders the proposed approach stronger than the one described in [10]. Finally, because Skype is closed source and its audio codec is not widely published, the approach proposed in [17] and [18] could not have been applied. Therefore, it has been necessary to study alternate methods able to attenuate or take the effect of noise into account.

Another interesting study to conduct would be to design a speaker identification or verification system. Speaker identification addresses the problem of determining who is talking from a set of known speakers, while speaker verification is concerned with the problem of verifying that the speaker is really the person they purport to be. Any other profiling information might also be of interest, especially for law enforcement agencies. So far, only the language identification issue has been addressed [18], although not for Skype. However, based on the results of this paper, age, gender, language or emotion detection also seems feasible for Skype.

References

1. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: VLDB 2003: Proceedings of the 29th International Conference on Very Large Data Bases, pp. 81–92. Morgan Kaufmann (2003)
2. Aggarwal, C.C., Olshefski, D., Saha, D., Shae, Z.Y., Yu, P.: CSR: speaker recognition from compressed VoIP packet stream. In: ICME 2005: Proceedings of the IEEE International Conference on Multimedia & Expo, pp. 970–973. IEEE Computer Society, Los Alamitos, CA, USA (2005)
3. Brookner, E.: Tracking and Kalman Filtering Made Easy. Wiley, New York (1998)
4. Chang, Y.C., Chen, K.T., Wu, C.C., Lei, C.-L.: Inferring speech activity from encrypted Skype traffic. In: Proceedings of IEEE Globecom 2008. IEEE Computer Society, Los Alamitos, CA, USA (2008)
5. Fu, A.W.C., Keogh, E., Lau, L.Y.H., Ratanamahatana, C.A., Wong, R.C.W.: Scaling and time warping in time series querying. VLDB J. **17**(4), 899–921 (2008)
6. Grewal, M.S., Andrews, A.P.: Kalman Filtering: Theory and Practice Using MATLAB. 3rd edn. Wiley, New York (2008)
7. Kalman, R.E.: A new approach to linear filtering and prediction problems. Transaction of the ASME. J. Basic Eng. **82**(Series D), 35–45 (1960)
8. Kalman, R.E., Bucy, R.S.: New results in linear filtering and prediction theory. Transactions of the ASME. J. Basic Eng. **83**(Series D), 95–107 (1961)
9. Keogh, E.J., Pazzani, M.J.: Derivative dynamic time warping. In: 1st SIAM International Conference on Data Mining. SIAM (2001)
10. Lella, T., Bettati, R.: Privacy of encrypted voice-over-IP. In: SMC'07: Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics, pp. 3063–3068. IEEE Computer Society (2007)
11. Salvador, S., Chan, P.: FastDTW: toward accurate dynamic time warping in linear time and space. Intell. Data Anal. **11**(5), 561–580 (2007)
12. Simon, D.: Kalman filtering. Embed. Syst. Program. **14**(6), 72–79 (2001)
13. Simon, D.: Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches. Wiley, New York (2006)
14. Simon, D.: Using nonlinear Kalman filtering to estimate signals. Embed. Syst. Design. **19**(7), 38–53 (2006)
15. Welch, G., Bishop, G.: An introduction to the Kalman filter. Tech. Rep. TR 95-041, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA (2006)
16. WinDump: Official website. http://www.winpcap.org/windump (2010)
17. Wright, C.V., Ballard, L., Coull, S.E., Monrose, F., Masson, G.M.: Spot me if you can: uncovering spoken phrases in encrypted VoIP conversations. In: SP'08: Proceedings of the 29th IEEE Symposium on Security and Privacy, pp. 35–49. IEEE Computer Society, Washington, DC, USA (2008)
18. Wright, C.V., Ballard, L., Monrose, F., Masson, G.M.: Language identification of encrypted VoIP traffic: Alejandra y Roberto or Alice and Bob? In: SS'07: Proceedings of the 16th USENIX Security Symposium, pp. 1–12. USENIX Association, Berkeley, CA, USA (2007)