Classifying Service Flows in the Encrypted Skype Traffic

Maciej Korczyński and Andrzej Duda
Grenoble Institute of Technology, CNRS Grenoble Informatics Laboratory UMR 5217
Grenoble, France. Email: [maciej.korczynski, andrzej.duda]@imag.fr

***Abstract*—In this paper, we consider the problem of detecting Skype traffic and classifying Skype service flows such as voice calls, skypeOut, video conferencing, chat, file upload and download. We propose a classification method for Skype encrypted traffic based on the Statistical Protocol IDentification (SPID) that analyzes statistical values of some traffic attributes. We have evaluated our method on a representative dataset to show excellent performance in terms of Precision and Recall.**

I. Introduction

Accurate traffic identification and classification are essential for proper network configuration and security monitoring. Application-layer encryption can however bypass restrictions set by network configuration and security checks. In this paper, we focus on Skype as an interesting example of encrypted traffic and provide a method for identifying different Skype flows inside encrypted TCP traffic—we want to discriminate between voice calls, video conferencing, skypeOut calls, chat, and file sharing. Previous papers on Skype concentrated on its architecture and the authentication phase [1], [2], [3], on the mechanisms for firewall and NAT traversal [4] as well as on characterizing traffic streams generated by VoIP calls and Skype signaling [5], [6]. Bonfiglio et al. proposed identification methods for encrypted UDP Skype traffic [7], but no work has considered encrypted TCP Skype flows.

Skype exemplifies the problem of identifying encrypted flows, because it multiplexes several services using the same ports: VoIP calls, video conferencing, instant messaging, or file transfer. A network administrator may assign a higher priority to VoIP calls, but other flows may also benefit in an illegitimate way from a higher priority if we cannot distinguish them from VoIP calls.

We propose a classification method for Skype encrypted traffic based on the Statistical Protocol IDentification (SPID) [8] that analyzes statistical values of flow and application layer data. We consider a very special case of Skype traffic that is, in addition to proprietary encryption, tunneled over Transport Layer Security (TLS) protocol version 1.0. We propose an appropriate set of attribute meters to detect encrypted Skype TCP traffic and identify Skype service flows. Our method involves three phases with progressive identification. To select the right attribute meters for each phase, we applied a method called *forward selection* [9] that evaluates how a given attribute meter improves classification performance and promotes it to the traffic model if its influence is significant. *Forward selection* uses the Analysis of Variance (ANOVA) [10]. We have evaluated our classification method on a representative dataset to show excellent performance in terms of Precision and Recall.

To the best of our knowledge, this is the first work that proposes an accurate method for classifying encrypted Skype service TCP flows tunneled over the TLS protocol.

II. Issues in the Analysis of Skype Traffic

Skype traffic presents a major challenge for detection and classification, because of proprietary software, several internal obfuscation mechanisms, and a complex connection protocol designed for bypassing firewalls and establishing communication regardless of network policies.

Skype differs from other VoIP applications, because it relies on a Peer-to-Peer (P2P) infrastructure while other applications use the traditional client-server model. Skype nodes include clients (ordinary nodes), supernodes, and servers for updates and authentication. An ordinary node with a public IP address, sufficient computing resources and network bandwidth may become a supernode. Supernodes maintain an overlay network, while ordinary nodes establish connections with a small number of supernodes. Authentication servers store the user account information. A Skype client communicates with other nodes directly or in an indirect way via other peers that relay packets. Skype can multiplex different service flows on an established connection: voice calls to another Skype node, skypeOut calls to phones, video conferencing, chat, file upload and download. Our goal is to detect and classify the service flows in Skype traffic. We cannot use traditional port-based flow identification methods, because Skype randomly selects ports and switches to port 80 (HTTP) or 443 (TLS 1.0) if it fails to establish a connection on chosen ports.

Another feature of the Skype design is the possibility of using both TCP and UDP as a transport protocol. Skype uses TCP to establish an initial connection and then it can interchangeably use TCP or UDP depending on network restrictions.

Skype encrypts its traffic with the strong 256-bit Advanced Encryption Standard (AES) algorithm to protect from potential eavesdropping. However, some information in the UDP payload is not encrypted so that a part of the Skype messages encapsulated in UDP can be obtained and used for identification [7]. We propose an accurate method for classification of service flows inside encrypted TCP Skype traffic tunneled

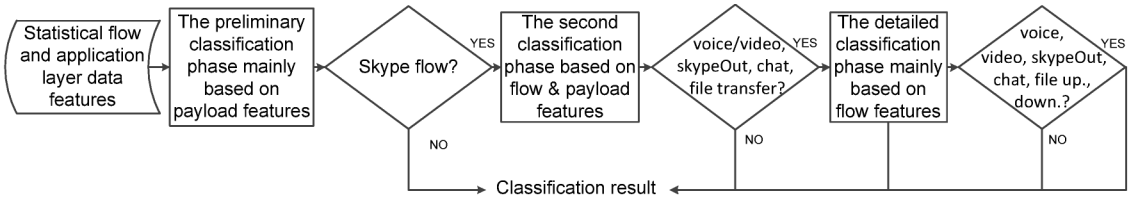

Figure 1. Simplified classification process.

over TLS (cf. Figure 1). It is a hybrid method combining traffic flow metering with Deep Packet Inspection (DPI) elements.

III. Classification Method

We need to apply a new methodology to classify encrypted TCP Skype flows. We propose to consider various statistical flow and application layer data features.

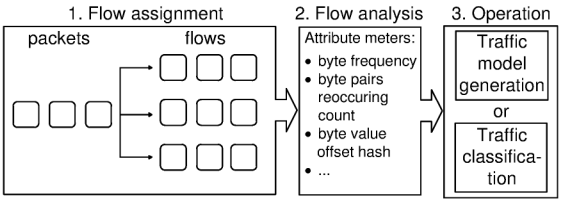

Figure 2. Three steps of SPID.

A. Classification Based on SPID

We build our method upon SPID (Statistical Protocol IDentification) [8] (cf. Figure 2). It is based on *traffic models* that contain a set of *attribute fingerprints* represented as probability distributions. They are created through frequency analysis of traffic properties called *attribute meters* of application layer data or flow features. An example of such an attribute meter is *byte frequency* that measures the frequency at which all of the possible 256 values occur in a packet. Other attribute meters defined later in Table I include for instance byte offset, byte re-occurring, direction change, and packet size.

As illustrated in Figure 2, SPID operates in three steps. First, packets are classified into bi-directional flows. All connections are represented as 5-tuples according to the source IP address, source port, destination IP address, destination port, and transport layer protocol. However, only packets carrying data are significant, because the analysis is based on both the application layer data and flow features. Then, each flow is analyzed in terms of attribute meters to obtain a collection of attribute fingerprints. Finally, the obtained attribute fingerprints are used either in traffic model generation or in traffic classification.

To illustrate the process of fingerprint creation, consider an example of the *byte frequency* attribute meter computed on the first 5 bytes of the TLS `Server Hello` packet, a part of the TLS handshake protocol. The first 3 bytes refer to the message type (0x16) and the TLS version (0x03 01), while the last two bytes correspond to the size of the remaining part of the TLS record (0x00 4a). Each time we observe a particular value, its counter is incremented. In the example, all five counters referring to the five values will be incremented. Then, SPID maintains a probability vector—the normalized counter vector with all elements summing up to one.

At the initial training phase, the method creates *traffic models*—attribute fingerprints representative for the traffic we want to detect. During the classification phase, the method computes attribute fingerprints on the flows to classify and compares them with traffic models by means of the Kullback-Leibler (K-L) divergence [11]:

$$D(P||Q) = K\text{-}L(P,Q) = \sum_{x \in X} P(x) log_2 \frac{P(x)}{Q(x)}. \quad (1)$$

The K-L divergence is a measure of the difference between two probability distributions $P(x)$ and $Q(x)$. $P(x)$ represents the distribution of a particular attribute of an observed flow and $Q(x)$ is the distribution corresponding to a known traffic model. Classification consists in comparing $P(x)$ with all known traffic models and selecting the protocol with the smallest average divergence $D(P||Q)$ and lower than a given threshold. We need to correctly set the divergence threshold to decrease the false positive rate for known traffic models—we only take into consideration the K-L divergence average values below the threshold.

Figure 1 presents a simplified process of the proposed classification method. In the first phase, it detects Skype traffic after a TCP three-way handshake based on the first five packets of the connection by considering attribute meters, the majority of which reflects application level data. Then, it changes the set of attribute meters to both packet independent and application level data features to detect service flows in the Skype traffic: voice/video, skypeOut, chat, and file transfer. This phase requires a larger number of packets to analyze to be effective: our calibration sets this value to 450 packets. Finally, the method considers more packets (the threshold is set to 760) to further distinguish between voice and video flows, and between file upload and download.

Table I present the set of attribute meters defined for classifying Skype traffic with notation presented in Table II.

B. Methodology for Attribute Meter Selection

Our classification process is based on three phases and each of them requires a proper set of attribute meters. We applied a method called *forward selection* for choosing attribute meters. It consists of starting with an initial attribute in the model, trying attributes out one by one, and adopting them, if they improve the classification performance. The selection terminates when adding an attribute does not improve the performance.

Table I
DEFINITION OF ATTRIBUTE METERS USED IN CLASSIFICATION

Attribute meter	Definition
byte-frequency	$\mathcal{M}_1 : \{(k, p_k)\}$, $k = 0, 1, ..., 255; p_k = \frac{m_k}{\sum m_k}$, $m_k = \sum_{i=1}^{8} \sum_{j=1}^{100} \delta_{x_j^i}$
action-reaction of first 3 bytes	$\mathcal{M}_2 : \{(h^i, p_{h^i}), \forall_{i \in (1,3)}\}$, $h : (y_{3\Delta}^i, z_{3\Delta}^i) \rightarrow h(y_{3\Delta}^i, z_{3\Delta}^i)$, $p_{h^i} = \frac{m_{h^i}}{\sum m_{h^i}}$, $m_{h^i} = \delta_{h(y_{3\Delta}^i, z_{3\Delta}^i)}$
byte value offset hash	$\mathcal{M}_3 : \{(h, p_h)\}$, $h : (j, x_j^i) \rightarrow h(j, x_j^i)$, $p_h = \frac{m_h}{\sum m_h}$, $m_h = \sum_{i=1}^{4} \sum_{j=1}^{32} \delta_{h(j, x_j^i)}$
first 4 packets byte reoccurring distance with byte	$\mathcal{M}_4 : \{(h, p_h)\}$, $\forall_{d<=16} : h : (x_j^i, d) \rightarrow h(x_j^i, d)$, $p_h = \frac{m_h}{\sum m_h}$, $m_h = \sum_{i=1}^{4} \sum_{j=1}^{32} \delta_{h(x_j^i, d)})$
first 4 packets first 16 byte pairs	$\mathcal{M}_5 : \{(h, p_h)\}$, $h : (x_j^i, x_{j+1}^i) \rightarrow h(x_j^i, x_{j+1}^i)$, $p_h = \frac{m_h}{\sum m_h}$, $m_h = \sum_{i=1}^{4} \sum_{j=1}^{16} \delta_{h(x_j^i, x_{j+1}^i)}$
first 4 ordered direction packet size	$\mathcal{M}_6 : \{(f, p_f)\}$, $f : (i, s(x^i), dir(x^i)) \rightarrow f(i, s(x^i), dir(x^i))$, $p_f = \frac{m_f}{\sum m_f}$, $m_f = \sum_{i=1}^{4} \delta_{f(i, s(x^i), dir(x^i))}$
first packet per direction first N byte nibbles	$\mathcal{M}_7 : \{(f, p_f)\}$, $\forall_{x^1 \in \{z^1, y^1\}} : f : (nib(x_j^1), j, dir(x^1)) \rightarrow f(nib(x_j^1), j, dir(x^1))$, $p_f = \frac{m_f}{\sum m_f}$, $m_f = \sum_{j=1}^{8} \delta_{f(nib(x_j^1), j, dir(x^1))}$
direction packet size distribution	$\mathcal{M}_8 : \{(f, p_f)\}$, $f : (s(x^i), dir(x^i)) \rightarrow f(s(x^i), dir(x^i))$, $p_f = \frac{m_f}{\sum m_f}$, $m_f = \sum_{i=1}^{s(x)} \delta_{f(s(x^i), dir(x^i))}$
byte pairs reoccurring count	$\mathcal{M}_9 : \{(f, p_f)\}$, $\forall_{x_j^i = x_j^{i+1}} : f : (x_j^i, dir(x_j^i), dir(x_j^{i+1})) \rightarrow f(x_j^i, dir(x_j^i), dir(x_j^{i+1}))$, $p_f = \frac{m_f}{\sum m_f}$, $m_f = \sum_{i=1}^{s(x)} \sum_{j=1}^{32} \delta_{f(x_j^i, dir(x_j^i), dir(x_j^{i+1}))}$

We consider a set of n attribute meters $x_1, ..., x_n \in X$ and a set of m Skype services. We begin with a model that includes the most significant attribute in the initial analysis. More precisely, we compute $F\text{-}Measure$ defined as:

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN},$$

$$F\text{-}Measure = \frac{2 * Precision * Recall}{Precision + Recall}, \quad (2)$$

for a particular Skype service and for each individual attribute meter. The True Positive (TP) term refers to all Skype flows that are correctly identified, False Positives (FPs) refer to all flows that were incorrectly identified as Skype traffic. Finally, False Negatives (FNs) represent all flows of Skype traffic that were incorrectly identified as other traffic.

We select attribute $x_i \in X$ with the largest average $F\text{-}Measure$ defined as $\max_{x \in X} \frac{1}{m} \sum_{a \in (1,m)} FM_a^x$, where FM_a^x denotes a^{th} observation of $F\text{-}Measure$ value corresponding to x^{th} attribute meter.

In the next step, each of the remaining attributes $x_1, ...x_{i-1}, x_{i+1}, ...x_n \in X$ is tested for inclusion in the model. We run several $F\text{-}tests$ (explained below) that compare the variance of $F\text{-}Measure$ values obtained in the preliminary selection, i.e. $FM_a^{x_i}$, where $a \in (1, m)$, with the corresponding values obtained after including each attribute meter separately.

Table II
NOTATION

$\mathcal{M} : \{(k, p_k)\}$ – attribute meter
m_k – attribute meter counter
$p_k, k = 0, 1, 2, ...$ – probability distribution of an attribute meter (corresponds to $Q(x)$ in traffic model generation and $P(x)$ in traffic classification)
δ – indicator function; $\delta : X \rightarrow \{0, 1\}, \delta_{x_j^i} = \begin{cases} 1 & \text{if } X = x_j^i \\ 0 & \text{if } X \neq x_j^i \end{cases}$
h – hash function, $h = 0, 1, 2, ...$
f – compressing function, $f = 0, 1, 2, ...$
x_j^i – byte j in packet i
$x_{j(m)}^i$ – bit m in byte j in packet i
$\sum_i x^i \leftrightarrow x$ – all packets in a TCP session
y^i – packet i, z^i – packet sent in a different direction than y^i
$x_{\Delta j}^i$ – first j bytes in packet i
d – distance between two identical bytes; if $x_j^i = x_{j-d}^i \Rightarrow d, 0 < d < j$
$s(x)$ – size of x; amount of packets in a TCP session
$s(x^i)$ – size of packet x^i in bytes
dir – packet direction
nib: $x_j^i \leftrightarrow x_{j(m \in (1...8))}^i$; $x_{j(m \in (1...4))}^i \; XOR \; x_{j(m \in (5...8)}^i \Rightarrow nib(x_j^i)$

Let us focus on a particular $F\text{-}test$ [10] that compares the influence of attribute meter $x_j \in x_1, ...x_{i-1}, x_{i+1}, ...x_n \in X$ with the first model based on $x_i \in X$. We examine two groups of $F\text{-}Measure$ values $FM_a^{x_i}$ and $FM_a^{x_{ij}}$ that respectively correspond to attribute x_i and to the set of two attribute meters, i.e. x_i and x_j. We test the null hypothesis that two means of the discussed population are equal. If we fail to reject it, the additional attribute meter does not improve the

classification performance and we need to exclude it from further consideration. To examine these two groups, we use the one-way Analysis of Variance (ANOVA) $F\text{-}test$ [10] that compares the variance between the groups to the variance within the groups. The *between–groups* variance is given by:

$$S_{bet} = m * \sum_{x} \frac{(\overline{FM}^{x} - \overline{FM})^2}{(k-1)}, \quad (3)$$

where $\overline{FM}^{x}$ denotes the mean of FM_a^x values, $\overline{FM}$ denotes the overall mean of $F\text{-}Measure$ observations, i.e. $FM_a^{x_i}$ and $FM_a^{x_{ij}}$, m is the number of $F\text{-}Measure$ values for Skype services and k is the number of groups (in the discussed case equal to 2). The *within–group* variance is given by:

$$S_{wit} = \sum_{x,a} \frac{(FM_a^x - \overline{FM}^{x})^2}{k * (m-1)}, \quad (4)$$

where FM_a^x denotes a^{th} observation corresponding to each x^{th} classification (in the discussed case to the classification based on x_i and the classification based on the set of two attributes x_i and x_j).

The $F\text{-}statistics$ is computed as $F = S_{bet}/S_{wit}$ and it follows the $F\text{-}distribution$ with $k-1$, $k*(m-1)$ degrees of freedom under the null hypothesis. If the null hypothesis is rejected and the average $F\text{-}Measure$ value corresponding to x_i is lower than $F\text{-}Measure$ related to the set of two attribute meters, i.e. x_i and x_j, then attribute x_j is considered as a candidate to be included in the model.

For each of the attribute meters, the method computes $F\text{-}statistics$ that reflects the contribution of attributes to the model. The most significant attribute is added to the model, if $F\text{-}statistics$ is above a predefined level set to 0.1. Moreover, if $F\text{-}statistics$ is above 1, it is included in the model and considered as a significant attribute meter. The *forward selection* method then computes $F\text{-}statistics$ again for the attribute meters still remaining outside the model and the evaluation process repeats. Therefore, attributes are added one by one to the model until no remaining attribute results in significant $F\text{-}statistics$.

IV. Evaluation Results

A. Dataset Description

The appropriate selection of packet traces containing ground truth information is one of the key aspects in the training and evaluation process. It should be as extensive as possible and should cover various environments. We have generated TCP Skype traffic in the following conditions: i) various operating systems: Linux, MacOS, Windows, ii) wireless and wired networks, iii) connections within one LAN as well as WAN connections between LANs located in France and Poland, and iv) different versions of Skype (2, 3, and 5). Overall, we gathered 479 Skype flow traces taking more than 770 MB.

For the traffic model generation purpose we have selected a group of traces generated by MacOS over a WAN connection between wireless LANs located in France and Poland. Our fingerprint database with 6 Skype service flow models has the size of 1.78MB in the XML format. We have used the remaining datasets to evaluate the classification mechanism.

Furthermore, we have gathered a separate set of traces without Skype traffic to test the discrimination of our method. It contains various types of traffic: SSL/TLS, SSH, HTTP, SCP, SFTP, VoIP, BitTorrent, and standard services like streaming, video conferencing, chat service, mail, file sharing. The traces contain 18945 flows of around 3GB and were gathered between December 2010 and March 2011.

B. Criteria of Classification Performance

We use three metrics to quantify the performance of classification: Precision, Recall, and $F\text{-}Measure$ (cf. Eq. 2). $F\text{-}Measure$ is an evenly weighted combination between Precision and Recall, which means that if the system can for instance identify skypeOut traffic with Precision 100% (no False Positives) and Recall is 96.6% then the $F\text{-}Measure$ is 98.2%.

C. Performance of Classification

To evaluate the proposed method, we have extended the version 0.4.6 of SPID [12].

Our method depends on three parameters: the amount of packets required for reliable traffic and flow identification during each of the three steps, the K-L divergence threshold, and the number of flows used in the training process. Due to the space limitation in this paper, we only present the final classification results after the calibration process: the number of packets in each phase is set to 5, 450, and 760 packets, respectively, the K-L divergence threshold of 1.9 and 15 training flows.

After each classification step, the classifier decides if there are any instances of Skype flow for further analysis. If the identification result is positive, then it continues with more detailed classification of TLS Skype flows with a different set of attribute meters. Otherwise, it finishes as no TLS Skype flows were recognized.

The objective of the first classification phase is to early detect encrypted Skype traffic tunneled in TLS connections. The most significant attribute meter chosen in the selection process is $\mathcal{M}_5$ (cf. Table I). Two other important attributes are $\mathcal{M}_7$ and $\mathcal{M}_6$ while $\mathcal{M}_3$, $\mathcal{M}_4$, and $\mathcal{M}_1$ are less meaningful. In addition to payload inspection attributes ($\mathcal{M}_5$, $\mathcal{M}_7$, $\mathcal{M}_3$, $\mathcal{M}_4$, and $\mathcal{M}_1$), we have chosen one typical flow based attribute that combines features like size, direction, and packet order number ($\mathcal{M}_6$). Such selection indicates that the first TLS packets contain some characteristic values that differ from the headers of other services that use TLS.

Table III
Performance of Phase 1, Early Recognition of Skype Traffic

Traffic	Precision %	Recall %	F-M. %
Skype	100	100	100
No Skype	100	100	100

Our experiments show that inspecting only the first five packets containing the payload is sufficient to reveal Skype traffic with F-Measure equal to 100% (cf. Table III).

Once the method detects Skype traffic, it classifies the underlying type of service, i.e. voice/video communication, skypeOut calls, chat, file sharing. In the second phase, the method uses another set of attribute meters ($\mathcal{M}_8$ as the most important, $\mathcal{M}_7$ as a significant one, and $\mathcal{M}_9$, $\mathcal{M}_2$, and $\mathcal{M}_5$ as additional ones). The selected set of attributes is composed of payload independent *direction packet size distribution* attribute meter ($\mathcal{M}_8$) with DPI attributes ($\mathcal{M}_7$, $\mathcal{M}_9$, $\mathcal{M}_2$, and $\mathcal{M}_5$).

Table IV
PERFORMANCE OF PHASE 2, CLASSIFICATION OF SKYPE FLOWS

Skype Service	Precision %	Recall %	F-M. %
voice/video	99.1	95.7	97.4
skypeOut	100	96.6	98.2
chat	86.4	100	92.7
file sharing	100	98.6	99.3

Table IV shows very good results of classification after inspecting 450 packets. However, this phase cannot distinguish between voice communications and voice/video calls, nor between file upload and download (denoted in Table IV as file sharing) due to similar traffic characteristics. Nevertheless, from the Quality of Service (QoS) perspective, network administrators may already give priority to Skype voice/video traffic and limit Skype file sharing flows regardless of the traffic direction.

Table V
PERFORMANCE OF PHASE 3, DETAILED CLASSIFICATION OF SKYPE FLOWS

Skype Service	Precision %	Recall %	F–M. %
voice	72.9	57.4	64.2
video	60.3	73.2	66.1
skypeOut	100	96.6	98.2
chat	90.2	97.4	93.7
file upload	100	96.9	98.4
file download	100	97.5	98.7

The objective of Phase 3 is to further refine the classification of voice and video flows as well as file sharing. We have applied $\mathcal{M}_8$ as the most important flow based attribute meter and DPI based $\mathcal{M}_7$ as an additional one. Table V presents the final results obtained after analyzing 760 packets. We can observe that the results are very good for most of Skype flows. We can now easily distinguish between file upload and download based on the flow attribute combining the direction with the packet size distribution (cf. attribute $\mathcal{M}_8$ in Table I). The classification is based on the fact that the sizes of packets sent from the client significantly differs from the sizes of packets sent in the opposite direction.

Classification of voice and video flows performs slightly worse, because our method does not capture some characteristics of the Skype behavior (it is meant to be applied to other classification problems as well). We have observed that in the case of Skype calls (both voice and video), the Skype client sends traffic simultaneously through several nodes depending on network conditions. In other words, the Skype voice or video traffic may spread on several TCP connections, which we cannot capture, because our method considers each TCP flow separately.

In contrast to voice/video communication and file sharing, we have noticed that chat messages and skypeOut calls seem to be sent through a single node. Considering chat messages, we have observed that when an intermediary node goes down, communication switches to another one without any interference for the users. This is not surprising if we take into account a small amount of data to send. For skypeOut calls, however, we have observed that the whole communication goes through a single intermediary node and the range of relay addresses is limited. This may come from higher requirements for bandwidth and computing resources to support high quality of calls. To sum up, in this classification step it was easier to identify these two type of services, because the whole traffic was sent over single flows.

V. CONCLUSIONS

In this paper, we have considered the problem of detecting encrypted Skype traffic tunneled over TLS and classifying Skype service flows. Our three-phase hybrid classification method is based on SPID and combines traditional statistical flow features with DPI elements. In each phase, we select a subset of relevant attribute meters through *forward selection* based on ANOVA. The performance of the method on a representative dataset is very promising—it achieves high Precision and Recall for most Skype service flows, whereas distinguishing between voice and video flows in the final classification phase is more challenging due to spreading traffic on several TCP connections.

ACKNOWLEDGMENTS

This work was partially supported by the EC FP7 project INDECT under contract 218086.

REFERENCES

[1] R. Alshammari and A. N. Zincir-Heywood, "Unveiling Skype Encrypted Tunnels Using GP," *IEEE CEC*, pp. 1–8, July 2010.
[2] D. Zhang, C. Zheng, H. Zhang, and H. Yu, "Identification and Analysis of Skype Peer-to-Peer Traffic," *5th International Conference on Internet and Web Applications and Services*, pp. 200–206, 2010.
[3] P. A. Branch, A. Heyde, and G. J. Armitage, "Rapid Identification of Skype Traffic Flows," *Proc. of the 18th Int. Work. on Net. and Operating Systems Support for Digital Audio and Video*, pp. 91–96, 2009.
[4] S. A. Baset and H. G. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol," *Proc. of INFOCOMM*, pp. 1–11, April 2006.
[5] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi, "Detailed Analysis of Skype Traffic," *IEEE Transactions on Multimedia*, vol. 11, no. 1, pp. 117–127, January 2009.
[6] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei, "Quantifying Skype User Satisfaction," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, October 2006.
[7] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing Skype Traffic: When Randomness Plays with You," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 37–48, 2007.
[8] E. Hjelmvik and W. John, "Statistical Protocol Identification with SPID: Preliminary Results," *6th Swedish National Computer Networking Workshop*, May 2009.
[9] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, 2003.
[10] G. E. P. Box, "Non-Normality and Tests on Variances," *Biometrika*, vol. 40, pp. 318–335, 1953.
[11] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *Annals of Mathematical Statistics*, vol. 22, pp. 49–86, 1951.
[12] "SPID Web Site," http://sourceforge.net/projects/spid.