

Optimum packet length masking

Alfonso Iacovazzi, Andrea Baiocchi

INFOCOM Dept. - University of Roma "Sapienza"

Via Eudossiana 18 - 00184 Roma, Italy

Phone: +39-0644585654, Fax: +39-064744481

Email: andrea.baiocchi@uniroma1.it, iacovazzi@infocom.uniroma1.it

Abstract—Application level traffic classification has been addressed in demonstrated recently based on statistical features of packet flows. Among the most significant characteristics is packet length. Even ciphered flows leak information about their content through the sequence of packet length values. There are obvious ways to destroy such side information, e.g. by setting all packet at maximum allowed length. This approach could entail an extremely large overhead, which makes it impractical. There is room to investigate the optimal trade-off between overhead/complexity of packet length masking and suppression of information leakage about flow content through packet length values. In this work we characterize the *optimum* first order statistical padding technique which guarantees indistinguishability of different application flows. We also discuss how to account for subsequent packet length correlation. Numerical results are shown with reference to real network traffic traces, specifically flows of HTTP, POP3, SSH, and FTP (control session) traffic.

Index Terms—Anonymization; privacy; traffic padding; traffic classification; packet length statistics

I. INTRODUCTION

A major issue of current and future Internet is traffic monitoring. A specific task of monitoring is identification of packet content at application layer for classification purposes [1]. Application level traffic classification can be useful for enforcement of security policies and traffic filtering, or it can support quality of service mechanisms. Several works and experimentations show that statistical traffic classification is possible based on a few features of the IP flows, namely packet inter-arrival times, direction (from client to server or vice-versa) and packet length, this last feature being a key one to classify application level content of a packet flow. Section II reports on some recent contributions in this direction. All of these works point out at the potential information leakage about user activity yielded by the ordered sequence of packet lengths of a traffic flow.

This paper addresses the problem of optimal masking of packet length information. To this end, two basic tools can be used: packet padding and packet fragmentation. Both means are effective if packets are enciphered, so as to forbid an attacker to learn the real packet length by stripping off padding or reconstructing fragmented packets, just as the receiver would do. Optimality refers to minimizing the average amount of overhead introduced to mask the real packet length.

We target a packet length concealing algorithm that can be deployed either at the data source or at an intermediate gateway, i.e. at an IPSec tunnel end point, so that actions that can be taken on packets are limited to those allowed by current IP technology (padding; packet fragmentation). Moreover, we

do not consider packet fragmentation, that requires significant processing of the traffic anonymization device and therefore brings higher costs as the channel speed grows. This work aims at exploring the trade-off and limitations of stochastic packet padding as to traffic anonymization against classifiers, added overhead and complexity in terms of knowledge required to set up the padding algorithm.

The rest of the paper is organized as follows. Related works are briefly reviewed in Section II. In Section III we outline the attack model and state the role of the traffic anonymization device. In Section IV we define the optimum padding problem and give an algorithm to compute an optimal random packet padding. In Section V we show numerical examples with measured traffic data. Finally, in Section VI we give our conclusions.

II. RELATED WORK

In [3], Karagiannis et al. develop a heuristic that uses social, functional and application level behaviours of a host to identify traffic flows originating from it. This approach, although really innovative, is tailored onto a specific source host. Most approaches aim at identifying application layer traffic from IP or transport (flow) level traffic measurements. Crotti et al. [4] used only size and inter-arrival time of first n packets to create a statistical descriptor (a Fingerprint) of an application layer protocol: this fingerprint is then used to measure the similarity of a certain flow to the corresponding protocol. Moore et al. [5] use a supervised machine learning algorithm called Naive Bayes (and its generalization, Kernel Estimation) on a wide set of characteristics (tens or hundreds), as flow duration, packets inter-arrival time and payload size and their statistics (mean, variance...). Moreover, they use a filtering technique to identify the best characteristics to be used with the mentioned methods. A number of works [6][7][8] rely on unsupervised learning techniques as K-means. McGregor et al. [6] explore the possibility to use cluster analysis to group flows using transport layer attributes, but they do not evaluate the accuracy of the classification. Zander et al. [7] extend this work using another Expectation Maximization (EM) algorithm named Autoclass. They also analyze the best set of attributes to use. Both these works only test Bayesian clustering technique trained by an EM algorithm, which has a slow learning time. Bernaille et al. [8] use faster clustering algorithms representing data in different spaces: K-means and Gaussian Mixture Models (GMM) for Euclidean space and Spectral clustering in Hidden Markov Models (HMM)

based space. The only features they use are packet size and packet direction: they demonstrate the effectiveness of these algorithms even using a small number of packets (e.g. the first four of a TCP connection). The HMM theory is used in [9]: packets size and inter-arrival time are used to build a model describing a certain protocol. The results of the training phase is a HMM model describing the behaviour of each protocol. Even though this approach can classify distinct encrypted applications, its performance on SSH is (76% detection rate and 8% false negative) is not as good as well known application traffic such as WWW and instant messaging. Other works are focalized on this topic, in particular on SSH. Alshammari et Al [10], work attempted to classify/identify applications services running over SSH. Results indicate that a supervised learning algorithm, RIPPER, can recognize applications inside SSH flows such as SCP and SFTP with accuracy up to 99.8% by running off-line analysis on complete traces. Concerning SSH encrypted application Dusi et al. [11] exploit GMM and SVM based techniques. They achieve accuracy up to 99.2% for POP3S analyzing four encoded packet after SSH handshake. So statistical classifiers based on packet length observations can reap easily more than 90% success in application flow classification.

Besides mere application protocol classification, there are other privacy breaking attacks considered in the literature based on analysis of traffic flows features, primarily on packet lengths. In [12] effective classification tools are demonstrated to identify encrypted web pages on the basis of similarities to features in a library of known profiles. Counter-measures based on padding techniques are shown to be effective though at the cost of a large overhead, i.e. up to 50% overhead to defeat most classifiers, but not all of them, up to 150% overhead to defeat all classifiers considered in that work. An earlier work that exploits object count and size to identify encrypted web pages is [13]. In [14] Authors exploit the length of encrypted VoIP packets to show that the language of the conversation can be successfully identified when the underlying audio is encoded using Variable Bit Rate (VBR) coders. Accuracy over 21 different languages is 66%, it is greater than 90% for 14 out of the 21 languages. This privacy breaking is based on analysis of the encoded and encrypted audio frame lengths. Related issues are investigated in [16], where packet length knowledge plays a minor role in assisting guessing of passwords typed in SSH session based on inter-packet times.

Finally, in a closely related work [15] Authors suggest morphing one class of traffic to look like another class. Through the use of convex optimization techniques, they show how to optimally modify packets in real-time to reduce the accuracy of a variety of traffic classifiers and prove their technique on real traffic data. Results indicate that classifiers can be made significantly less effective at reasonable costs in terms of packet length overhead (some tens per cent). Major differences between [15] and the approach we propose is that [15] introduces *morphing*, i.e. they aim at transforming flows generated by one application so that they look like flows generated by another application. They do not provide an optimal solution to do this, rather a numerical procedure

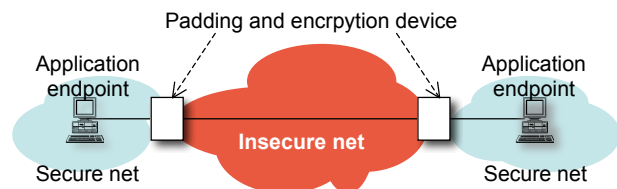


Fig. 1. Scenario for the privacy attack on the packet traffic.

to approximate optimum, i.e. minimum overhead, *given the initial and the target applications*. Moreover, they assume it is possible to cut packets if the morphing process requires output packets shorter than the corresponding input ones. In this work we propose to mask flows from any number of input applications with the constraint that only padding be provided, we give an explicit optimum solution (minimum average overhead) and we account for the different probability distributions of the packet lengths according to the position of the packet inside the flow and for the correlations between packet lengths of subsequent packets inside a flow. The scope of this work is in some sense more limited than [15], since we confine ourselves to padding only, but the results are more general (masking instead of morphing) and easier to use (explicit solution, no computational complexity due to heavy numerical convex optimization). The value of the result is to provide a theoretical limit on the overhead required to mask application layer traffic by resorting to padding only.

III. SCENARIO OF THE ATTACK ON PRIVACY BASED ON PACKET LENGTHS

As discussed in previous Section, packet length statistics leaks information about what application originates packets, even if flows are encrypted. To define the security problem we have to outline the attack scenario and the attacker model.

We assume a quite general setting, where we can identify origin and destination secure networks (each possibly reducing to a single device), where application endpoints are located (see Figure 1). In between there is an insecure network, where the attacker can observe all flowing packets and attempt to break the privacy of the information flows carried in the insecure network. Confidentiality of packet payload is protected by encryption, but we concede the attacker can identify boundaries of application layer flows, i.e. the attacker can select packets making up a single session of an (unknown) application protocol out of the aggregated packet traffic observed on a network link¹. Then, the attacker can apply statistical classification to identify the specific application that has originated the flow, even though she can not read into the packet payloads. We want to prevent this attack, specifically the information leakage given by the *packet lengths*. We consider packet padding to mask this information.

Figure 2 illustrates a block diagram of the padder. Packet belonging to different application flows enter the edge device

¹As a matter of example, for TCP based applications, a *flow* is defined as the ordered list of packets belonging to a same TCP connection, starting from the first packet after the three-way handshake for TCP connection establishment.

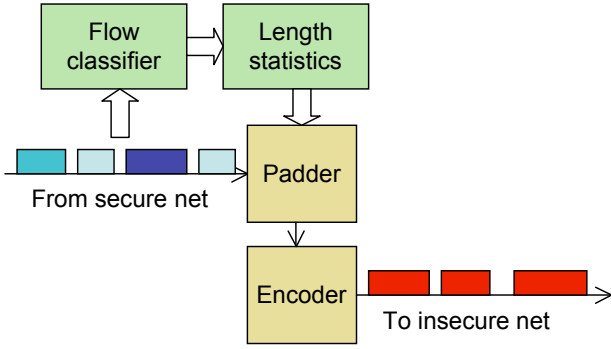


Fig. 2. Block scheme of padder operation in case of trusted/insecure network edge.

connecting the secure network to the public, insecure network. The padder contains tables that give the amount of overhead to be added as a function of the incoming packet length and the application type it belongs to. Such tables can be computed once the probability distributions of the packet lengths of the applications to be mixed have been estimated (see next Section). Then, they are filled and periodically refreshed by a background process that observes incoming traffic (the two upper blocks in the figure, connected with the large white arrow). From analysis of this packet stream, classification of application flows is possible (traffic is assumed to be uncoded in the secure network) and packet length statistics can be estimated.

To read the proper table and apply padding, the *padder box* must know the application the incoming packet belongs to. This information cannot be obtained by classification in real time, since the initial packets of a flow must be released outside to let the application progress. So, when classification is possible in a reliable way, a number of packets have already been released with no padding or a padding compute without knowledge of the correct table to be used. This difficulty can be overcome by means of a cooperating secure network, where some tag is added to application to be mixed, so that they can be recognized by the padder since the very first packet of each flow. A more practical situation can be envisaged in the common case where the secure network reduces to a single host device. Then, padding and background length statistics collection can be carried out by an internal process, e.g. embedded in the operating system. Such a process can obviously know the exact application/service each packet flow belongs to, since they are generated within the same device under the control of a same operating system.

The obtained padded packet is enciphered, so as to protect confidentiality of payload and prevent the attacker from removing padding (*encoder box*). At the far end, packets are deciphered, padding is stripped off and clean packets are forwarded to the appropriate application layer endpoint.

In the following, we focus on the padder box. The aim of the padder is to alter packet length so as to confuse a given set of pre-defined application protocols. The key idea is to add a random amount of padding so that lengths of output packets appear as drawn from a *same* Probability Mass

Function (PMF) *independently* of the application that has actually generated them.

IV. OPTIMUM PADDING ALGORITHM

Let us consider M application layer protocols \mathcal{A}_k for $k = 1, \dots, M$. As for the packet lengths, we assume application layer entity of each protocol can be characterized by a probability measure. Let $X_i^{(k)}$ be the random variable representing the length of the i -th packet of a flow generated by application protocol k , $i \geq 1$ and $k = 1, \dots, M$. For any random variable X we let $F_X(n) = \mathcal{P}(X \leq n)$ be the cumulative probability distribution function for $n \geq 0$.

We consider only packet length padding, so that lengths of packets of the anonymized flow are given by $Y_i^{(k)} = X_i^{(k)} + U_i^{(k)}$, where the $U_i^{(k)}$'s are non negative random variables in general. The value of $U_i^{(k)}$ can be a function of $X_j^{(h)}$ for $j \leq i$ and $h = 1, \dots, M$, which guarantees that the padding algorithm can be run in real time, with minimum delay of padded packets (just processing time delay, no need to wait for following packets). This condition also enables the padding device to be different from the source of packet flow.

Let us focus on two application protocols ($M = 2$) and on a specific packet within their respective flows, say the i -th one. We drop the subscript i for the sake of simple notation. Let $a_n = \mathcal{P}(X^{(1)} = n)$ and $b_n = \mathcal{P}(X^{(2)} = n)$ for $n = L_{min}, \dots, L_{max}$. Practical values of minimum and maximum packet lengths are e.g. $40 \text{ bytes} \leq L_{min} \leq 56 \text{ bytes}$, depending on options on IP or TCP headers, and $L_{max} = 1500 \text{ bytes}$ for most widespread access networking technology. To keep notation simple, without losing generality, we set $L_{min} = 1$, i.e. the minimum length quantum (e.g. one byte), and $L_{max} \equiv \ell$. We let also $F_a(n) = \mathcal{P}(X^{(1)} \leq n)$ and $F_b(n) = \mathcal{P}(X^{(2)} \leq n)$.

We aim to make packet length series belonging to the two protocols indistinguishable once packets are padded. So, it must be $Y^{(1)} \sim Y^{(2)} \sim Y$. Let $c_n = \mathcal{P}(Y = n)$. We search for a Probability Mass Function (PMF) $\{c_n\}_{1 \leq n \leq \ell}$ among all those satisfying the constraint that only padding be non negative, i.e. $Y^{(i)} = X^{(i)} + U^{(i)}$ with $U^{(i)} \geq 0$, $i = 1, 2$. We show first the following:

Theorem: Let $\{a_n\}_{1 \leq n \leq \ell}$ and $\{b_n\}_{1 \leq n \leq \ell}$ be the PMFs of lengths of packets of two applications. Let $\{c_n\}_{1 \leq n \leq \ell}$ any PMF describing the padded lengths of both applications. Then.

$$F_c(n) \equiv \sum_{j=1}^n c_j \leq \min \{F_a(n), F_b(n)\}, \quad n = 1, \dots, \ell. \quad (1)$$

Proof: Since only padding is allowed, the length of output packets is $Y_1 = X^{(1)} + U_1$ or $Y_2 = X^{(2)} + U_2$, where $Y_1 \sim Y_2 \sim Y$. Therefore, $\mathcal{P}(Y > k) \geq \mathcal{P}(X^{(1)} > k)$ whence $\mathcal{P}(Y \leq k) \leq \mathcal{P}(X^{(1)} \leq k)$. Similarly for the other random variable, $X^{(2)}$. It follows that any output packet length PMF in case of padding must satisfy $\mathcal{P}(Y \leq k) \leq \min\{\mathcal{P}(X^{(1)} \leq k), \mathcal{P}(X^{(2)} \leq k)\}$ or

$$\sum_{j=1}^k c_j \leq \min \left\{ \sum_{j=1}^k a_j, \sum_{j=1}^k b_j \right\}, \quad k = 1, \dots, \ell \quad (2)$$

q.e.d. ■

We aim at minimizing the amount of overhead due to padding. Given the PMFs of the padder input packet lengths, this is the same as minimizing $E[Y]$. We can prove the following:

Theorem: Let $\{a_n\}_{1 \leq n \leq \ell}$ and $\{b_n\}_{1 \leq n \leq \ell}$ be the PMFs of lengths of packets of two applications. Then $E[Y^*] \leq E[Y]$ for any PMF $\{c_n\}_{1 \leq n \leq \ell}$ of the r.v. Y under the non negative padding constraint, with the PMF $\{c_n^*\}_{1 \leq n \leq \ell}$ of the r.v. Y^* given by

$$F_{c^*}(n) \equiv \sum_{j=1}^n c_j^* = \min \{F_a(n), F_b(n)\}, \quad n = 1, \dots, \ell. \quad (3)$$

Proof: First, we argue $F_{c^*}(n)$ is a proper Cumulative Distribution Function (CDF), if $F_a(n)$ and $F_b(n)$ are. It is non negative, monotonous non decreasing and it attains 1 for $n = \ell$, since both $F_a(n)$ and $F_b(n)$ do so.

Further, we have

$$\begin{aligned} E[Y^*] &= \sum_{j=1}^{\ell} j c_j^* = \sum_{j=1}^{\ell} [1 - F_{c^*}(j)] \\ &= \sum_{j=1}^{\ell} [1 - \min\{F_a(j), F_b(j)\}] \\ &\leq \sum_{j=1}^{\ell} [1 - F_c(j)] = E[Y] \end{aligned}$$

where last inequality derives from eq. (1). q.e.d. ■

The PMF $\{c_n\}_{1 \leq n \leq \ell}$ is just the target common PMF of the packet length at the output of the padder device to the insecure network. It is the optimum one, i.e. the output packet length PMF with minimum mean value (hence minimum average overhead, given the mean length of input packets) under the constraint that only padding is applied (i.e. no packet fragmentation).

Once $\{c_n\}_{n=1, \dots, \ell}$ is given, it is possible to compute the PMF of the random overhead U , conditional on the input packet length of the m -th application, namely $x_{h,k}^{(m)} = \mathcal{P}(U = h | X^{(m)} = k)$ for $h = 0, 1, \dots, \ell - k; k = 1, \dots, \ell$ and for $m = 1, \dots, M$. The values $x_{h,k}^{(m)}$ depend on the marginal PMF $\mathcal{P}(X^{(m)} = k)$. As a matter of fact, for $m = 1$ we have

$$\begin{aligned} c_n &= \mathcal{P}(Y = n) \\ &= \sum_{k=1}^n \mathcal{P}(X^{(1)} = k) \mathcal{P}(X^{(1)} + U = n | X^{(1)} = k) \\ &= \sum_{k=1}^n a_k \mathcal{P}(U = n - k | X^{(1)} = k) \\ &= \sum_{k=1}^n a_k x_{n-k,k}^{(1)} \quad n = 1, \dots, \ell. \end{aligned} \quad (4)$$

The values of $x_{h,k}^{(m)}$ can be computed by Algorithm 1. At step k , we consider the fraction of input packets of length k , i.e. a_k : at the output we have a packet with length n

with probability $[\sum_{j=1}^n c_j - \sum_{j=1}^{k-1} a_j] / a_k$ (provided this is positive and less than 1). This is simply the probability of the output length be not greater than n minus the probability mass of the output length PMF already ‘‘assigned’’ to input packet of length less than k . Then, the conditional probability that overhead U is no greater than $n - k$ is

$$\begin{cases} \mathcal{P}(U \leq n - k | X^{(1)} = k) = \min \{1, \max \{0, z_{k,n}\}\} \\ z_{k,n} = \frac{1}{a_k} \left(\sum_{j=1}^n c_j - \sum_{j=1}^{k-1} a_j \right) \end{cases} \quad (5)$$

for $k = 1, \dots, n$ and $n = 1, \dots, \ell$ (as usual it is intended that $\sum_{j=j_1}^{j_2} \equiv 0$ for $j_1 > j_2$).

Let us assume that, for a fixed n , the smallest value of k such that $F_c(n) < F_a(k)$ be k^* ; then it is $F_c(n) \geq F_a(k)$ for $k = 1, \dots, k^* - 1$. Note that it is $1 \leq k^* \leq \ell$ and this is well defined since $F_a(\ell) = 1 \geq F_c(n) \forall n$. Then, it is

$$\begin{aligned} z_{k,n} &\geq 1, \quad k = 1, \dots, k^* - 1 \\ z_{k^*,n} &= \frac{1}{a_{k^*}} \left(\sum_{j=1}^n c_j - \sum_{j=1}^{k^*-1} a_j \right) \in [0, 1) \\ z_{k,n} &\leq 0, \quad k = k^* + 1, \dots, n - 1. \end{aligned}$$

Then, we have

$$\begin{aligned} \mathcal{P}(Y \leq n) &= \sum_{k=1}^n a_k \mathcal{P}(U \leq n - k | X = k) \\ &= \sum_{k=1}^{k^*-1} a_k + a_{k^*} z_{k^*,n} = \sum_{j=1}^n c_j = F_c(n) \end{aligned}$$

Algorithm 1 Computation of the PMF of the padding overhead conditional on the input packet length

```

1: for  $n \leftarrow 1$  to  $\ell$  do
2:   for  $k \leftarrow 1$  to  $n$  do
3:      $z = 0$ 
4:     if  $a_k > 0$  then
5:        $z = \frac{\sum_{j=1}^n c_j - \sum_{j=1}^{k-1} a_j}{a_k}$ 
6:     end if
7:      $\gamma_{n-k,k} = \min \{1, \max \{0, z\}\}$ 
8:   end for
9: end for
10:  $x_{0,k} = \gamma_{0,k}$ 
11: for  $k \leftarrow 1$  to  $\ell$  do
12:   for  $h \leftarrow 1$  to  $\ell - k$  do
13:      $x_{h,k} = \gamma_{h,k} - \gamma_{h-1,k}$ 
14:   end for
15: end for

```

The arguments of the proofs as well as the algorithms can easily be generalized to the case of M input PMFs of packet lengths that are to be confused into a single target PMF. The key characteristic of this common PMF, which is equivalent to Algorithm ??, is

$$\sum_{j=1}^k c_j = \min \left\{ \sum_{j=1}^k a_j^{(1)}, \dots, \sum_{j=1}^k a_j^{(M)} \right\} \quad (6)$$

for $k = 1, \dots, \ell$.

A. Generalization to conditional packet length PMFs

The Algorithm 1 aims at computing a overhead length PMF used to pad packets from M different application protocols, so that the *marginal* PMF of the i -th packet of each application flow has a resulting length that is drawn from a same PMF, irrespective of the specific application that generated that packet. What we need to compute the target PMF and hence the conditional pad overhead PMFs is knowledge of the PMF of the i -th packet emitted by each application, i.e. $a_k^{(m)}(i) = \mathcal{P}(X_i^{(m)} = k)$, $m = 1, \dots, M; k = 1, \dots, \ell; i \geq 1$, where the subscript i of $X_i^{(m)}$ refers to the order of occurrence of the packet inside the flow it belongs to. According to the algorithms defined above, we can compute a padded packet length PMF for each value of i , $\{c_n(i)\}_{n=1, \dots, \ell}$

This way we neglect correlation information. While marginal distribution of packet length is completely masked, we could expect some information leakage still take place since subsequent packets belonging to a same flow have correlated packet lengths. We can tackle this issue, at least for one-step dependencies, by considering *conditional* PMFs instead of just marginal ones. For the sake of notation, we consider two applications only, the generalization to M being straightforward as done in eq. (6). Let $a_k(1) = \mathcal{P}(X_1^{(1)} = k)$ and $b_k(1) = \mathcal{P}(X_1^{(2)} = k)$; let also

$$\begin{aligned}\tilde{a}_{k|h}(i) &= \mathcal{P}(X_i^{(1)} = k | X_{i-1}^{(1)} = h) \\ \tilde{b}_{k|h}(i) &= \mathcal{P}(X_i^{(2)} = k | X_{i-1}^{(2)} = h)\end{aligned}$$

with $k, h = 1, \dots, \ell$ and $i \geq 2$. The target padded packet length PMF $\{c_n(i)\}$ is computed by exactly the same expression as eq. (3), except that $\{\tilde{a}_{k|h}(i)\}_k$ and $\{\tilde{b}_{k|h}(i)\}_k$ are fed as input for each given value of h instead of $\{a_k(i)\}_k$ and $\{b_k(i)\}_k$. Analogously, the PMF of the random padding to be applied to a packet of length k belonging to e.g. application 1 is computed from $\{\tilde{a}_{k|h}(i)\}_k$ and $\{c_n(i)\}_n$ as $\{\tilde{x}_{j|k,h}(i)\}_{j=1, \dots, \ell-k}$ for each given value of h and k . Computational burden is strongly reduced by the typically high correlation found in packet length sequences², that imply $\{\tilde{a}_{k|h}(i)\}_k$ is non null only for few values of h .

B. Partial masking

We have considered optimum masking to destroy any information leakage of the application flow packet lengths that could enable successful classification. Optimality refers to padding overhead minimization. We could consider partial information leakage removal versus overhead amount, still under the constraint that only padding be allowed.

Let us assume that flows of application i are a fraction ρ_i of all the considered flows. Assume a hypothetical classifier based on the observation of a single packet length, L . The

²This is just another face of the good capability of statistical classifiers found in the literature, as discussed in Section I

probability of correct classification, P_{cc} , is

$$\begin{aligned}P_{cc} &= \sum_{k=1}^{\ell} \mathcal{P}(L = k) \mathcal{P}(\text{correct classification} | L = k) \\ &= \sum_{k=1}^{\ell} \sum_{i=1}^M \mathcal{P}(\mathcal{A}_i, L = k) \mathcal{P}(D = \mathcal{A}_i | \mathcal{A}_i, L = k) \\ &= \sum_{k=1}^{\ell} \sum_{i=1}^M \rho_i a_k^{(i)} \delta_{i,k}\end{aligned}\quad (7)$$

where \mathcal{A}_i denotes the i -th application and D is the classification decision output. In the following we assume that $\rho_i = 1/M$. The decision variables $\delta_{i,k}$ must satisfy the condition $\sum_{i=1}^M \delta_{i,k} = 1$. Then, it is easy to see that

$$P_{cc} \leq \sum_{k=1}^{\ell} \max_{1 \leq i \leq M} \{\rho_i a_k^{(i)}\} = \frac{1}{M} \sum_{k=1}^{\ell} \max_{1 \leq i \leq M} \{a_k^{(i)}\} \equiv \frac{S_M}{M}\quad (8)$$

The equality sign is achievable by letting $\delta_{i,k} = 1/\nu_k$ for each k , where $a_k^{(j)} = \max_{1 \leq i \leq M} \{a_k^{(i)}\}$ for $j = j_1, \dots, j_{\nu_k}$ and $a_k^{(j)} < \max_{1 \leq i \leq M} \{a_k^{(i)}\}$ for $j \neq j_1, \dots, j_{\nu_k}$ ($1 \leq \nu_k \leq M$). Note that $P_{cc} \geq 1/M$; this minimum is attained when the best possible classification is just guessing at random.

Full masking of application flows is shown to consist in reducing PMFs of packet lengths belonging to different application to a same (optimum) common PMF, which we denoted with $\{c_k\}_{1 \leq k \leq \ell}$. Modification of packet length PMF is obtained by padding of original packets. Partial masking can be defined in general as statistical padding that turns the PMF $\{a_k^{(i)}\}_{1 \leq k \leq M}$ to a new PMF $\{c_k^{(i)}\}_{1 \leq k \leq M}$ for $i = 1, \dots, M$. The non negative padding constraint implies that $\sum_{k=1}^n c_k^{(i)} \leq \sum_{k=1}^n a_k^{(i)}$, for $n = 1, \dots, \ell$ and $i = 1, \dots, M$.

The average amount of overhead after this partial masking padding is

$$H \equiv \frac{\sum_{i=1}^M \rho_i (\mathbb{E}[Y^{(i)}] - \mathbb{E}[X^{(i)}])}{\sum_{i=1}^M \rho_i \mathbb{E}[X^{(i)}]} = \frac{\sum_{i=1}^M \sum_{k=1}^{\ell} k c_k^{(i)}}{\sum_{i=1}^M \sum_{k=1}^{\ell} k a_k^{(i)}} - 1\quad (9)$$

An optimization problem can be stated, to minimize H under non negative padding constraints and a constraint on the maximum allowed correct classification probability, i.e. requiring that

$$P_{cc} = \frac{1}{M} \sum_{k=1}^{\ell} \max_{1 \leq i \leq M} \{c_k^{(i)}\} \leq \epsilon\quad (10)$$

where $\epsilon \in [1/M, S_M/M]$. Note that it is $1 \leq S_M \leq M$. Also the non negative padding constraint must be considered.

This general problem is here stated though its solution is outside the scope of this contribution and it is left for further work.

V. NUMERICAL RESULTS

According to the scenario defined in Figures 1 and 2, we consider a collection of traces (ground truth), made up of the ordered sequence of packet lengths of flows belonging to different applications. We consider the following application

layer protocols: HTTP, FTP-c (control session), SSH, POP3. Then $M = 4$ in the numerical results presented in this Section. All of these application layer protocols are supported by TCP so that a flow is identified by all packets belonging to the same TCP connection. We count packets starting from the first data packet, i.e. the first one immediately after the three way handshake of the TCP to establish the connection. So, each flow is defined by the sequence $[x_1, \dots, x_Q]$ of the lengths (in bytes) of the first Q packet of that flow; a metadata is tied to this descriptor to label the application it belongs to.

We use the same dataset collected and commented in [17]. Traffic is generated by considering clients and servers placed in University lab and industrial lab LANs, as well as private domestic locations. All connections are through public Internet. Servers and clients use different operating systems, so as to collect a representative traffic sample. For each of the four considered application protocols 2000 flow are collected. The classifier algorithm is based on K-means clustering technique with an optimized number of clusters³: 1000 flows for each application are used for training, performance tests are run on the other 1000 flows per application. Packet lengths for the used traffic dataset range from 52 bytes up to 1500 bytes (sizes refer to IP packets).

The cumulative probability distribution functions (cpdfs) of the flow first packet for the four considered applications is potted in Figure 3 along with the cpdf of the padded packets, $\{c_n\}$. We are mixing applications with typically short packets (few hundred bytes) such as POP3, SSH and FTP-c, with HTTP, whose packet lengths easily saturate to the maximum 1500 bytes. As a consequence, it is apparent that the probability mass be concentrated around length 100-150 and about 1400-1500. In the light of this, padding overhead is expected to be large.

The classifier performance is qualified by the confusion matrix $\mathbf{K} = [\kappa_{i,j}]$. Let D be the classifier output decision, i.e. the application that has supposedly generated the observed packet flow, and let A be the actual application that has generated the observed flow. We can represent both D and A as integer ranging from 1 to M . Then $\kappa_{i,j} = \mathcal{P}(D = j|A = i)$ for $i, j = 1, \dots, M$. Diagonal elements represent success probabilities of the classifier, while off-diagonal elements on each row are mis-classification decision probabilities. A flow classifier corresponds to an information channel that maps input flows (D “symbols”) into classification decisions (A “symbols”) and is therefore described by the matrix \mathbf{K} .

We can model the application flow classifier as an information channel that takes padded flows as input and outputs a label which is the supposed application each flow belongs to. A perfect classifiers has a confusion matrix \mathbf{K} equal to the identity matrix, hence an average mutual information from input to output equal to $\log_2 M$ bits of information per input flow. For a given confusion matrix the average mutual information of the “information channel” from A to D is

³It has been chosen via cross-validation so as to maximize the success rate of the classifier.

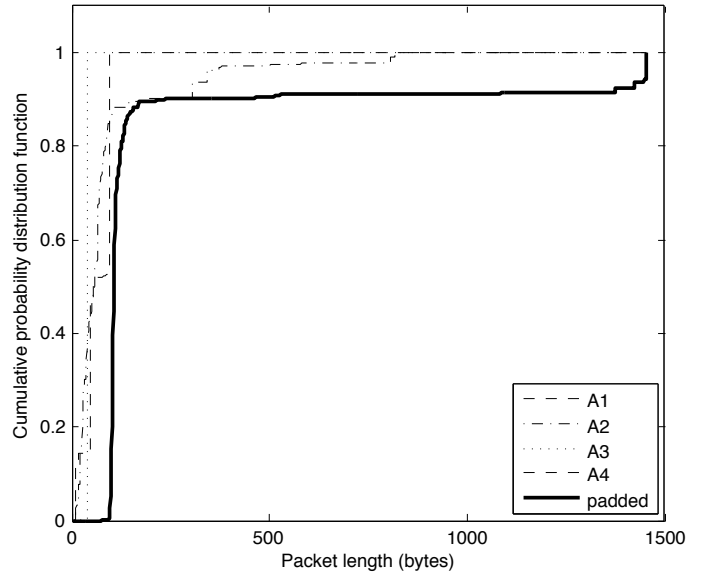


Fig. 3. Cumulative probability distribution function of the packet lengths for the four considered applications (A1=HTTP; A2=FTP-c; A3=SSH; A4=POP3) and for the packets padded according to PMF $\{c_n\}$ of eq. (6).

obtained from its very definition as

$$I(A; D) = \log_2 M + \frac{1}{M} \sum_{i=1}^M \sum_{j=1}^M \kappa_{i,j} \log_2 \left(\frac{\kappa_{i,j}}{\kappa_j} \right) \quad (11)$$

where $\kappa_j = \sum_{r=1}^M \kappa_{r,j}$. In the following we consider the normalized mean mutual information $\hat{I}(A; D) = I(A; D) / \log_2 M$, which is just the fraction of the mean mutual information of the ideal classifier that a real classifier attains. We prefer to use this performance parameter since it makes clearer comparison than matrices; also, the aim of masking is making flow classification based on packet length fail, which does not mean it should necessarily become same as random guessing. Perfect masking, leading to failure of the flow classifiers, can be defined as follows: $\mathcal{P}(D = j) = \mathcal{P}(D = j|A = i)$ for all $j = 1, \dots, M$. This definition is reminiscent of perfect secrecy by Shannon [18] and indeed it implies that the classifier decision is *independent* of the analyzed flow, i.e. information provided by the analyzed flow (ordered sequence of packet lengths in our case) is totally irrelevant as to flow classification. With this definition, $\hat{I}(A; D) = 0$ in case of perfect masking, while $\hat{I}(A; D) = 1$ for an ideal classifier.

It is to be stressed that the K-means classifier is trained by feeding it with *padded* flows, so that it can learn to recognize any feature that possibly leaks from the ordered sequence of padded packet lengths, that is to say *after* the application of the masking algorithm. This is consistent with the usual security approach where the attacker is granted knowledge of the security algorithm, i.e. the padding anonymization in our case. Results are shown in Tables I, II and III: the average mutual information associated to the flow classifier $\hat{I}(A; D)$ and the average fraction of output bytes that are padding overhead are listed as a function of the number Q of packets of each flow examined by the classifier. The average overhead

fraction is defined as

$$E[\text{overhead}] = \frac{\sum_{m=1}^M \sum_{i=1}^Q E[U_i^{(m)}]}{\sum_{m=1}^M \sum_{i=1}^Q (E[U_i^{(m)}] + E[X_i^{(m)}])} \quad (12)$$

The padding algorithm is effective in cancelling most of the

Q	No padding	With padding			
	$\hat{I}(A; D)$	marginal PMFs $\hat{I}(A; D)$	$E[\text{overhead}]$	conditional PMFs $\hat{I}(A; D)$	$E[\text{overhead}]$
1	0.6316	0.0020	0.1275	0.0011	0.1261
2	0.6795	0.0453	0.1034	0.0024	0.0870
3	0.9919	0.0464	0.1234	0.0027	0.1049
4	0.8698	0.2692	0.1762	0.0474	0.0854
5	0.9971	0.1481	0.3717	0.0457	0.3399

TABLE I

AVERAGE MUTUAL INFORMATION OF THE CLASSIFIER BASED ON THE FIRST Q PACKETS OF THE APPLICATION FLOWS: TWO APPLICATIONS (SSH, POP3).

Q	No padding	With padding			
	$\hat{I}(A; D)$	marginal PMFs $\hat{I}(A; D)$	$E[\text{overhead}]$	conditional PMFs $\hat{I}(A; D)$	$E[\text{overhead}]$
1	0.6096	0.0005	0.3148	0.0005	0.3148
2	0.7938	0.0066	0.5100	0.0027	0.2399
3	0.8267	0.0727	0.6325	0.0710	0.4649
4	0.9093	0.1046	0.6315	0.0818	0.4466
5	0.9115	0.1255	0.6112	0.1131	0.4426

TABLE II

AVERAGE MUTUAL INFORMATION OF THE CLASSIFIER BASED ON THE FIRST Q PACKETS OF THE APPLICATION FLOWS: FOUR APPLICATIONS (HTTP, FTP-c, SSH, POP3).

Q	No padding	With padding			
	$\hat{I}(A; D)$	marginal PMFs $\hat{I}(A; D)$	$E[\text{overhead}]$	conditional PMFs $\hat{I}(A; D)$	$E[\text{overhead}]$
1	0.8697	0.0001	0.1295	0.0001	0.1292
2	0.9441	0.0034	0.1570	0.0021	0.0757
3	0.9478	0.0013	0.1183	0.0010	0.0606
4	0.9943	0.0030	0.1463	0.0053	0.1062
5	0.9033	0.0254	0.2256	0.0003	0.1165

TABLE III

AVERAGE MUTUAL INFORMATION OF THE CLASSIFIER BASED ON THE FIRST Q PACKETS OF THE APPLICATION FLOWS: TWO APPLICATIONS TUNNELED INSIDE SSH CONNECTIONS (HTTP-OVER-SSH, SFTP).

information provided by the flow classifier, which is otherwise quite successful in detecting origin application, at least when a sufficient number of packets is considered (e.g. $Q = 5$). As Q increases, a growing amount of information leaks through the padder device, since correlation of the flow packet length sequence are not masked in case of marginal padded packet length PMF or only partially masked in case of one-step conditional padded packet length PMF.

With two application protocols to be mixed up (SSH and POP3, Table I), an almost perfect classifier ($\hat{I}(A; D) = 0.99$ for $Q = 5$) is turned into a poor or even an extremely poor tool with random padding based on marginal PMFs (about 0.15 residual average mutual information) or on conditional PMFs (less than 0.05 average mutual information left). Overhead

increases as the scope Q of the classifier grows, reaching between 34% and 37% of the output traffic. Similar results are found in terms of effectiveness reduction of the classifier in case four applications are considered (HTTP, FTP-c, SSH and POP3, Table II). Overhead is much larger due to the remarkable difference of typical message lengths in HTTP and the other application: the first one tends to exhibit packets close to the maximum 1500 bytes size, the other three protocols typically send packets between few tens and some hundreds of bytes.

A third different numerical example gives more striking results (Table III). In this case we consider application services tunneled inside SSH connections (so that every packet is entirely encrypted). Even a simple K-means based classifier can be very effective in spotting which application service is being tunneled within each observed SSH connection by just looking at the sequence of packet lengths. Details on the dataset collection and pre-processing and on the classifier optimization are given in [17]: we just mention that in this case 8000 flows have been used, 4000 to train the classifier and 4000 to test it and obtained results in Table III. Random padding as defined in this work is definitely effective in killing classifier capability, e.g. an information leakage that makes the K-means classifier almost perfect for $Q = 4$ is largely obfuscated with only about 11% overhead traffic at the output of the padding device in case of conditional PMFs and 15% overhead with marginal PMFs. In general, conditional PMF approach has superior performance both in terms of anonymization effectiveness and amount of required overhead.

VI. CONCLUSIONS

This work focuses on application layer traffic anonymization by means of packet padding, to defeat application flow classifiers based on ordered sequences of flow packet lengths. Several works have shown that these classifier can be very successful in distinguishing between different applications just looking at packet lengths. Besides enciphering packet payloads, traffic anonymization requires avoiding information leakage on application by modifying packet lengths. We address packet padding, which leads to simpler devices that should scale to high speed link gracefully. The aim of the work is to explore limits and trade-offs of packet padding. Given probability distributions of the packet lengths on input application flows, we identify the optimal padding probability distribution. Optimality consists in minimizing the padded mean packet length.

From numerical findings based on real traffic datasets, we can draw some remarks. Given only padding is allowed, then

- even if optimal padding is determined, overhead can be quite large (efficiency);
- quite a large amount of statistical data collection is required to provide solid basis for computation of optimal padding PMF;
- traffic anonymization through padding does destroy a large amount of the information potentially attained by a classifier, but still a fraction of this information leaks.

These drawbacks could be overcome by allowing also fragmentation to come into play, though at the cost of adding a

significant complexity to the devices in charge of carrying out traffic masking.

REFERENCES

- [1] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, D. Sadok, A Survey on Internet Traffic Identification, *IEEE Communications Surveys & Tutorials*, Volume 11, Issue 3, 3rd Quarter 2009, pages: 37-52.
- [2] T. Ylonen and C. Lonvick, The Secure Shell (SSH) Protocol Architecture, RFC 4251, IETF, Jan. 2006.
- [3] T. Karagiannis, K. Papagiannaki, M. Faloutsos, BLINC: Multilevel traffic classification in the dark, *Proceedings of ACM SIGCOMM 2005*, Philadelphia, PA, USA, August 2005.
- [4] M. Crotti, M. Dusi, F. Gringoli, L. Salgarelli, Traffic Classification through Simple Statistical Fingerprinting, *ACM SIGCOMM Computer Communication Review*, Vol. 37, No. 1, pp. 5-16, Jan. 2007.
- [5] A.W. Moore, D. Zuev, Internet traffic classification using Bayesian analysis techniques, *ACM SIGMETRICS 2005*, Banff, Alberta, Canada, June 2005.
- [6] A. McGregor, M. Hall, P. Lorier, J. Brunskill, Flow clustering using machine learning techniques, *PAM 2004*, Antibes Juan-les-Pins, France, April 2004.
- [7] S. Zander, T. Nguyen, G. Armitage, Automated traffic classification and application identification using machine learning, *IEEE Conference on Local Computer Networks (LCN 2005)*, Sydney, Australia, November 2005.
- [8] L. Bernaille, R. Teixeira, and K. Salamatian, Early Application Identification, in *Proceedings of CoNEXT*, 4-7 December 2006, Lisboa, Portugal, 2006.
- [9] C. Wright, F. Monrose, G. Masson, On Inferring Application Protocol Behaviors in Encrypted Network Traffic, *Journal of Machine Learning Research (JMLR): Special issue on Machine Learning for Computer Security*, volume 7, pp. 2745-2769, January 2006.
- [10] R. Alshammari and A. Nur Zincir-Heywood, A Flow Based Approach For SSH Traffic Detection, *IEEE International Conference on Systems, Man and Cybernetics*, Montral, Canada, 7-10 October 2007.
- [11] M. Dusi, A. Este, F. Gringoli, L. Salgarelli, Using GMM and SVM-based Techniques for the Classification of SSH-Encrypted Traffic, *Proceedings of the 44th IEEE International Conference on Communication (ICC'09)*, Dresden, Germany, June 14-18, 2009.
- [12] Marc Liberatore and Brian Neil Levine, Inferring the Source of Encrypted HTTP Connections, *13th ACM conference on Computer and Communications Security (CCS'06)*, October 30 - November 3, 2006, Alexandria, Virginia, USA.
- [13] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu. Statistical identification of encrypted web browsing traffic. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 19-30, May 2002.
- [14] Charles V. Wright Lucas Ballard Fabian Monrose Gerald M. Masson, Language Identification of Encrypted VoIP Traffic: Alejandra y Roberto or Alice and Bob?, In *Proceedings of the 16th Annual USENIX Security Symposium*, pages 43-54, Boston, MA, August 2007.
- [15] Charles V. Wright, Scott E. Coull, Fabian Monrose, "Traffic Morphing: An Efcient Defense Against StatisticalTraffic Analysis", *16th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, 8 - 11 February 2009.
- [16] D. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and SSH timing attacks. In *Proceedings of the 10th USENIX Security Symposium*, August 2001.
- [17] G. Maiolini, G. Molina, A. Baiocchi, A. Rizzi, On the fly Application Flows Identification by exploiting K-Means based classifiers, *Journal of Information Assurance and Security*, issue no. 2 (2009), pp. 142-150.
- [18] C. E. Shannon, Communication theory of secrecy systems, *Bell Syst. Tech. J.*, Vol. 28, pp. 656-715, October 1949.