Website Fingerprinting in Onion Routing Based Anonymization Networks

Andriy Panchenko
Interdisciplinary Center for Security, Reliability and Trust
University of Luxembourg
http://lorre.uni.lu/~andriy/
{firstname.lastname}@uni.lu

Lukas Niessen
Computer Science dept.
RWTH Aachen University
lukas.niessen@rwth-aachen.de

Andreas Zinnen, Thomas Engel
Interdisciplinary Center for Security, Reliability and Trust
University of Luxembourg
{firstname.lastname}@uni.lu

ABSTRACT

Low-latency anonymization networks such as Tor and JAP claim to hide the recipient and the content of communications from a *local observer*, i.e., an entity that can eavesdrop the traffic between the user and the first anonymization node. Especially users in totalitarian regimes strongly depend on such networks to freely communicate. For these people, anonymity is particularly important and an analysis of the anonymization methods against various attacks is necessary to ensure adequate protection. In this paper we show that anonymity in Tor and JAP is not as strong as expected so far and cannot resist *website fingerprinting* attacks under certain circumstances. We first define features for website fingerprinting solely based on volume, time, and direction of the traffic. As a result, the subsequent classification becomes much easier. We apply support vector machines with the introduced features. We are able to improve recognition results of existing works on a given state-of-the-art dataset in Tor from 3% to 55% and in JAP from 20% to 80%. The datasets assume a closed-world with 775 websites only. In a next step, we transfer our findings to a more complex and realistic open-world scenario, i.e., recognition of several websites in a set of thousands of random unknown websites. To the best of our knowledge, this work is the first successful attack in the open-world scenario. We achieve a surprisingly high true positive rate of up to 73% for a false positive rate of 0.05%. Finally, we show preliminary results of a proof-of-concept implementation that applies camouflage as a countermeasure to hamper the fingerprinting attack. For JAP, the detection rate decreases from 80% to 4% and for Tor it drops from 55% to about 3%.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*

WPES'11, October 17, 2011, Chicago, Illinois, USA.

General Terms

Security

Keywords

Anonymous Communication, Website Fingerprinting, Traffic Analysis, Pattern Recognition, Privacy

1. INTRODUCTION

Anonymous communication aims at hiding the relationship between communicating parties on the Internet. Thereby, anonymization is the technical basis for a significant number of users living in oppressive regimes [15] giving users the opportunity to communicate freely and, under certain circumstances, to evade censorship. For these people, anonymity is particularly important, and an analysis of anonymization methods against various attacks is necessary to ensure adequate protection. Users must be able to trust the methods and know their limits. If the governmental bodies in totalitarian regimes could find out the content of communication and evasion of censorship, this might lead to severe consequences for the users such as imprisoning or even life-threatening ones. For that reason, reliable and secure anonymization methods are of high importance.

Anonymization networks such as Tor [7] and Jap [1] promise anonymity by routing data through several overlay nodes and using layered encryption of the content. They promise to strengthen a user's civil rights, to protect the privacy, or even to give a user the opportunity to evade the censorship. Clearly it is vital to know the networks' level of protection and to enforce their promised protection.

Several attacks against anonymization networks have been discovered, e.g., [6, 17, 19, 18], most notable the traffic confirmation attack. All of them require a particular power, i.e., access to particular data. Often, an attacker must observe both ends of the communication at the same time [6, 7]. Totalitarian regimes such as China or Iran usually do not have control over the communication party located in western countries precluding a traffic confirmation attack. Still, they are able to observe the connection between the user and the first anonymization node. A so-called local eavesdropper can be for example a local system administrator, an ISP, or anybody who is able to monitor the link between the original message sender and the first node of the anonymization network (e.g., everyone in the sending range of a victim communicating via a wireless link). Since local eavesdropping is one of the weakest attacks one can imag-

ine, this attack is very realistic and anonymization networks must by all means be secure with respect to local attacks.

Website fingerprinting (WFP) [12] is an attack that can be performed by a local attacker. WFP is a special form of *traffic analysis* where a local attacker observes the encrypted data and tries to draw conclusions from certain features of the traffic, such as the volume of transferred data, the timing, or the packet sizes. The method does not aim at breaking the cryptography, but even if an attacker does not understand a message's semantic, he can try to match observed patterns to known patterns of, e.g., web pages. If successful, website fingerprinting destroys the whole protection and anonymity promised by, e.g., Tor and JAP as it can be carried out by local attackers.

Previous approaches [11] applying WFP only achieve recognition rates of less than 3% in a closed-world setting for a limited set of 775 web pages. Closed-world means that all web pages are known in advance. At 3%, one can assume that many websites are not sufficiently recognizable, and therefore the protection still seems ensured. In this work, we consider closed-world to ensure the comparability of our results to related works. We show the superiority of our methods and that anonymization networks do not provide sufficient protection against website fingerprinting in closed-world.

Much more interesting though is the open-world scenario where a real attacker does not know in advance which URLs are ordinarily visited by the victim. One of the most prominent examples for the open-world scenario of WFP is the authorities that monitor a small set of *censored* web pages and try to find out whether a victim visits one of them, whereas they are not interested in other (*uncensored*) web pages visited by the victim. Hence, the classifier should be able to first detect access to censored web pages (this problem is hard due to the large background class) and second to classify the corresponding censored web pages. In order to achieve this, the training data needs to contain both censored and uncensored instances. Despite the attack's simplicity and the problem's complexity in open-world, we show that website fingerprinting is stronger than expected so far and is a major risk for users that rely on the protection of anonymization networks.

The contributions of this paper are as follows: (*i*) we show that Tor and JAP, the largest anonymization networks used today, do not offer sufficient security against WFP. We first define and analyze features for WFP solely based on volume, time, and direction of the traffic. We apply support vector machines with the introduced features. We are able to improve recognition results in closed world on a given state-of-the-art dataset of 775 websites in Tor from 3% to 55% and in JAP from 20% to 80%. (ii) We extend the website fingerprinting attack under the closed-world assumption to the much harder problem in open-world. We achieve a surprisingly high true positive rate of up to 73% for a false positive rate as low as 0.05%. To the best of our knowledge, we are the first who show the feasibility of website fingerprinting in the more complex open-world setting; (*iii*) we study the influence of the proposed features on the recognition results. This analysis is the basis for camouflage as a practical countermeasure. We provide preliminary results on how camouflage effectively hampers the attack along the introduced features without modifications in the underlying anonymization protocols. Therefore, we provide a proof-of-concept implementation showing how camouflage can enhance the anonymity.

The paper is structured as follows. Section 2 summarizes related work in the paper's context. In Section 3 we describe the experimental setup and the data set of our analysis. Section 4 introduces our method including features for WFP and standard machine learning algorithms. Section 5 summarizes the experimental results. Finally Section 7 concludes and discusses the contributions of this paper.

2. RELATED WORKS

The term *website fingerprinting* was coined by Hintz in [12], even though there are earlier works that pursued the same goal. Back in 1996, Wagner and Schneier pointed out that traffic analysis could be used to draw conclusions about content of encrypted SSL packets [23]. Two years later, the attack was successfully mounted against a single server offering different URLs [16, 4].

Several related works are based on a very strong assumption that web objects can be differentiated by examining the TCP connection between the user and the proxy, which only holds true for HTTP/1.0. For the first time, Sun et al. [20] presented a fingerprinting attack that particularly targeted the deanonymization of users of an SSL encrypted connection to a proxy server. The detection rate of about 75% was achieved while recognizing 2,000 pages in the set of 100,000 pages. As similarity metric the authors used the *Jaccard coefficient* [21]. By applying padding, the authors achieved a depression of the detection rate to 4% at the cost of tripled data volume. Hintz [12] presented a similar attack on the *SafeWeb*[1] web proxy. In his proof-of-concept experiment, the author distinguished between 5 websites based on similarities of the object sizes and achieved detection rates of 45 up to 75%.

As protocols applied nowadays make use of persistent connections and pipelining (e.g., HTTP/1.1 [8], Tor circuits), no new TCP connection is opened for each object and, hence, it is no longer possible to trivially distinguish between single objects in HTTP requests. Therefore, the above mentioned techniques based on object sizes are no longer feasible. Bissias et al. [3] were the first ones to perform the attack based on IP packet sizes and packet inter-arrival times instead of object sizes. The investigated setup consisted of an *OpenSSH*[2] tunnel to a proxy server. The authors computed correlation coefficients for both the temporal and packet size data sets of 100 web pages and achieved detection rates of about 20%.

Liberatore and Levine [14] carried out a more extensive investigation. They extended the attack on OpenSSH tunnels to a large scale and used a sample of 2,000 web pages. Instead of a simple metric based on the correlation coefficient, the authors applied more sophisticated methods from data mining, namely the Jaccard's coefficient and a *naïve Bayes classifier* [13]. The only considered feature is the packet size of the transmitted data whereas timing and order information is neglected. The success rate of correct classification in both methods is roughly 70%.

Recently, Wright et al. [24] analyzed the influence of *morphing* as a countermeasure against statistical traffic analysis.

[1] *SafeWeb* was a free web-based proxy service with SSL support. The service was discontinued in late 2001.
[2] `http://www.openssh.org/`

Morphing software transforms one class of traffic so that it looks like another class. The technique was evaluated for protection of VoIP and web traffic. While morphing against the naïve Bayes classifier in a web traffic data set of 50 URLs, the recognition rate fell from 98% to 64% while having only 39% overhead. Instead of transforming websites, we obfuscate the page by loading another page in parallel.

Besides recognition of websites, several works deal with detection of other characteristics, e.g., the language of a VoIP call [25], or even of phrases in encrypted VoIP calls [26] using hidden Markov models. Gong et al. [10] applied remote traffic analysis to show the feasibility of website fingerprinting in a small scale (distinguishing between 12 and 24 URLs).

The most recent publication in the area of website fingerprinting is by Herrmann et al. [11]. The authors investigated recognition of 775 websites by many different anonymization techniques: OpenSSH, OpenVPN, Stunnel, Cisco IPsec-VPN as well as the onion routing based systems Tor [7] and JAP [2]. Besides the classifiers used by Liberatore and Levine, the authors also apply the *multinomial naïve Bayes classifier*. The latter achieves detection rates of more than 90% in all single hop systems. The results for the multi hop systems, however, differ considerably. Whereas for JAP only 20% of the pages could be classified correctly, the detection rate for Tor is only 2.95%.

Given the results of Herrmann, it might seem as if anonymization networks are still secure due to low recognition rate in closed-world. In this paper, we increase the corresponding detection rate to an alarming degree. We show that padding is not a sufficient countermeasure in the way it is already included in Tor and JAP. In addition to existing approaches, we provide a detailed analysis of features serving as a basis for the definition and the design of additional countermeasures. We provide preliminary work on camouflage as a promising countermeasure in anonymization networks. To the best of our knowledge, we finally provide the first approach for successful website fingerprinting in the open-world setting.

3. DATA SETS

In practice an attacker first retrieves a certain amount of relevant web pages by himself as training data for fingerprinting, using the anonymization network that he assumes his victim uses as well. He records the transferred packets with a traffic analyzer tool such as *tcpdump*[3] which provides information about IP layer packets, e.g., the length of the packet, the time the packet was sent or received, the order in which the packets were sent and received, etc. The attacker can make use of various information contained in the dumps to create a profile of each web page, the so-called *fingerprint*. Later, wiretapping on the victim's traffic, the attacker similarly collects data which we call *test data*. The test data resembles the fingerprints, but it usually differs from them due to a number of reasons, e.g., indeterministic packet fragmentation, updates in web pages, etc. Hence, the attacker needs to apply statistical methods to compare the recorded information to the fingerprints and to probabilistically match it to a certain web page.

In this paper, we simulate an attacker's practical procedure. In doing so, we evaluate our methods for fingerprinting (Section 4) in Section 5 and our countermeasures in Section 6 by collecting three different data sets. For fetching the websites, we use lab computers running under *Debian GNU/Linux*. As browser, we use *Firefox* in version 3. Active contents (e.g., *Flash*, *Java*, *JavaScript*) as well as the browser cache are disabled[4]. If such active content is enabled, detection rates probably go up because more unique and therefore distinguishing data is transmitted, hence, pages become more distinguishable. To automate the retrieval of web pages, we use the Firefox plugin *Chickenfoot*[5]. Chickenfoot features a script language and is primarily written to enable the automated execution of typical user actions such as typing a URL into the browser's address bar. The experiments were performed in the beginning of 2010 using Tor in version 0.2.0.35 and JAP in version 00.11.012.

The Closed-World Dataset is taken from the recent related work of Herrmann et al. [11] in order to compare our results to previous best achieved results. Based on this data set, we show superiority of our features and methods compared to the state-of-the-art work (see Section 5.2.1). Next, we extend this proof-of-concept scenario – which uses a closed world assumption – to an open world one. To make it realistic we use 1,000,000 most popular Internet pages from a well-known web statistics service to collect our Open-World Dataset (see Section 5.2.2 for the results in open-world). Finally, we collect a Countermeasures Dataset in order to evaluate camouflage as a suitable countermeasure (see Section 6 for results).

3.1 Closed-World Dataset

To ensure comparability of our detection rates with most recent works, we take the list of 775 URLs used by Herrmann et al. [11] and first improve the detection under the closed-world assumption: the victim retrieves only web pages from the predefined set and the attacker knows the set of possible web pages. This set of URLs is built from the most popular URLs according to a proxy server used for 50 schools in Germany [11].

To collect training and test data, we retrieve 20 instances per website from our list of 775 sites. For each fetch of a URL, we store the sizes of packets in the observed order while recording incoming packets as positive, outgoing ones as negative numbers. This representation contains the sequence in which the packets were sent or received.

We collect datasets for two popular anonymization networks. For Tor we collect only one data set with the URLs mentioned above. JAP offers two different cascade types, namely free cascades and premium (paid) cascades. The service operators promise a higher degree of anonymity for the premium cascades, which shall be accomplished by having at least three mix servers in one cascade, and these being spread across several countries. Moreover, the premium cascades offer guaranteed uptime and higher performance than the free cascades. To evaluate protection of both types of cascades in JAP, we use two different configurations: a free cascade which consists of only one mix (with more than 2,000 users) and a commercial cascade consisting of three mixes (with few dozens of users). This number of users is typical for each configuration of JAP.

[3] `http://www.tcpdump.org/`

[4] For privacy reasons it is recommended to disable active contents while using anonymization techniques such as Tor [9].

[5] `http://groups.csail.mit.edu/uid/chickenfoot/`

Herrmann et al. [11], using their best classifier – MNB – achieved a 2.96% detection rate on the Closed-World Dataset in the Tor network. We achieved similar results. In addition, we found out that in the real Tor network a significant amount (36.4%) of pages was not completely loaded. The default browser timeout of 90s as proposed by Herrmann was not sufficient because of a poor performance in Tor. In the experiments of this paper, we try to include only fully loaded pages. This behavior is rather natural since in reality, most users would initiate a reload if a load fails or takes too long. To achieve the effect of completely loaded pages in our script, we simply increase the timeout to 600s for all data sets. This does not mean that in practice a user would wait 600s for loading a page. This modification already leads to an increase of the detection rate to 7.08% for MNB on the Closed-World Dataset. Applying SVMs, we obtain an increase of detection rate from 11.09% to 30.98%.

3.2 Open-World Dataset

Besides the comparison with existing methods in closed-world, the impact of website fingerprinting in an open-world setting is especially important. Contrary to 775 URLs only in the closed-world, the user can now access any possible URL. The attacker such as a totalitarian regime, however, is interested in finding out whether the victim accesses one of the censored URLs. The attacker is not interested which of the uncensored URLs are visited. Since in this scenario an attacker does not have a possibility to find out which URLs are ordinarily visited by the victim, the website fingerprinting becomes much more difficult. We need a large and representative data set in order to evaluate our methods and features. Ideally, such a data set includes realistic uncensored and censored pages of arbitrary formats.

For uncensored web pages, we use the list of the 1,000,000 most popular Internet pages from the web statistics service *Alexa*[6]. Out of this list, we randomly choose 4,000 URLs as uncensored and include 1 instance each to the training data. Furthermore, we include 1,000 random URLs (disjunct from training) as uncensored with one instance to the test data.

In practice, the censored web pages contain real illegal websites. Clearly, we had to choose an alternate and legal approach to not break the law in order to evaluate our method. Therefore, we decided to use three different lists of URLs for open-world to testify to the universality of our results: The first set of interesting pages is chosen to contain sexually explicit content (we call this set *Sexually Explicit*), which is legal in EU and US, but illegal in many other countries, e.g., in many countries in Asia and in the Middle East. The second dataset contains the most popular pages from the Alexa list (called *Alexa Top Ranked*). The third set contains pages that are randomly chosen from the Alexa URLs (called *Alexa Random*). We expand the training set by 5 censored URLs with 35 instances each (for Sexually Explicit, Alexa Top Ranked and Alexa Random). Finally, we add new 25 instances of the same censored URLs to the test data.

3.3 Countermeasures Dataset

Section 6 summarizes the effect of camouflage in anonymization networks. To demonstrate the strengths of camouflage, we apply the method to the more difficult case which is the closed-world. In this case the protection is harder than in the open-world. If we manage to hamper the attack in this setting, it is intuitively hampered even more in the open-world scenario. We collect the Countermeasures Dataset similar to the closed-world dataset based on 775 sample URLs. Unlike in the closed-world, web pages are not loaded separately. Instead, we at the same time load a random website. This leads to confusion of data. Please refer to Section 6 for a detailed description of camouflage and the effects on the recognition results.

In summary, we collect three representative data sets, namely the Closed-World, the Open-World and the Countermeasures Dataset. We deliberately did not select the corresponding URLs in the datasets on our own, but used state-of-the-art representative lists including a great number of diverse possible web page types with a high probability. In open-world, we even consider the worst case scenario where the attacker has never seen (i.e., does not train on) the ordinarily URLs visited by a victim. Note that we have a disjoint dataset for tuning the features and optimizing the SVM parameters. This is done in order to obtain representative results. The following sections will introduce our algorithms for website fingerprinting, camouflage as countermeasure and the corresponding results.

4. A NEW APPROACH

In this section we describe our new approach for website fingerprinting. Our contributions are twofold. First, we define general and powerful features (Section 4.1) that facilitate the subsequent classification. Second, we apply state-of-the-art machine learning technique for pattern recognition (Section 4.2). We compare our approach with Herrmann et al. [11] using Naïve Bayes. Later, we show that the results can significantly be improved by applying the more powerful support vector machines on the defined features.

4.1 Features

In data mining, general and powerful features are defined in order to facilitate the subsequent classification. In doing so, certain characteristics that are already implicitly in the data are made explicit to increase the accuracy of classification. The subsequent machine learning algorithm practically disregards features with little or no predictive information while considering features with great predictive information.

Previous works on website fingerprinting, e.g., [11, 14], only used the packet size and whether packets are incoming or outgoing as features. In the following, we define additional features that help to heavily improve the detection rates both in closed- and open-world. Clearly, feature engineering is far from trivial. In this paper, we define features by exploiting additional characteristics of the data. Since the power of features cannot easily be predicted, we empirically tested a large set of features. For the paper's sake, we restrict ourselves solely on the most powerful features that are described in the following:

- **Without Packets Sized 52**: In data mining, feature values often cannot be used to typify a classification problem. For the given problem, packets of length 52 occur for all possible classes (web pages). Usually, these packets are used to send acknowledgments between sender and receiver (TCP ACK packets with no

[6] `http://www.alexa.com/`

payload). The feature Without Packets Sized 52 filters the corresponding packets as noise.

- **Size Markers**: We introduce *markers* (special text labels) at each point in the traffic flow where the direction changes, i.e., where a package with positive size follows one with negative or vice versa. At each direction change, a marker is inserted reflecting how much data was previously sent into the respective direction. We sum up the size of all packets of a similar direction to obtain the Size Markers. Subsequently, the values are rounded. Several rounding increments were tested, and an increment of 600 yielded the best detection rates. The results show that both size markers and number markers have an impact on the classification results. Note that this feature improved the results only in combination with feature Without Packets Sized 52 as otherwise the direction of traffic flow changes almost after each packet.

- **HTML Markers**: When a browser retrieves a web page, the HTML document is requested, received and parsed. Subsequently, the browser requests embedded objects such as pictures. The order of object retrieval is indeterministic, but the HTML document certainly has to be accessed first because it contains the links to the embedded objects. Counting the size of incoming packets between the first outgoing packet (request for the HTML document) and follow-up outgoing packets (requests for embedded objects), we can extract the HTML document's size and use it as a feature which obviously contributes to the discrimination between pages. To this end we use a special marker while using the same rounding increment of 600 as in feature Size Markers.

- **Total Transmitted Bytes**: Even though the amount of transmitted data is already implicitly represented by the raw data and feature Size Markers, we additionally include the explicit markup of the total transmitted bytes. Practically, we add the size of all packets separately for incoming and outgoing packets. These numbers are rounded in increments of 10,000 and then appended to each instance, prefixed by a certain letter. We also examined the use of other rounding increments, but the best results were achieved using the above mentioned increment.

- **Number Markers**: As mentioned before, markers are introduced in order to indicate direction changes in the traffic flow. For each direction change, a marker is inserted reflecting how many packets were previously sent into the respective direction. The feature Number Markers performed best when grouping the packet size as follows: 1, 2, 3-5, 6-8, 9-13, 14. Once more, this feature improved the results only in combination with feature Without Packets Sized 52 as otherwise the direction of traffic flow changes almost after each packet.

- **Occurring Packet Sizes**: For each instance of a website, we counted the number of occurring packet sizes. Subsequently, the number was rounded in increments of 2 and added explicitly as an additional feature. Incoming and outgoing packets are considered separately.

- **Percentage Incoming Packets**: This feature adds the percentage of incoming/outgoing packets rounded in steps of 5.

- **Number Of Packets**: This feature represents the total number of packets. In a similar way to the total size of transmitted data, the Number Of Packets is calculated separately for incoming and outgoing packets. The best results were achieved when rounding the result in increments of 15.

As mentioned before, we restricted ourselves solely on the most powerful features in the preceding itemization. A number of features did not improve the results. Among others, we considered incoming/outgoing packets only, leaving out frequent/rare packet sizes, including TLS/SSL record sizes (which can be up to 2^{14} bytes long) or leaving out empty TLS records[7], preserving the packet order with n-grams (subsequence of n packets from a given sequence is joined together as a single attribute), rounding packet sizes, and rounding packet frequencies. A detailed discussion of these features is out of scope of this paper.

4.2 Support Vector Classification

Unlike Herrmann et al. using the Naïve Bayes classifier, we apply the more powerful support vector machines (SVM). SVMs are state-of-the-art supervised learning methods used in data mining which are well-known for their high performance in terms of the classification accuracy. The technique dates back to the work of Vapnik and Chervonenkis [22] in 1974. The key idea is the interpretation of instances as vectors in a vector space. In the case of website fingerprinting, the features and raw data are derived for one page retrieval and represented as a vector. Based on training data, the classifier tries to fit a hyperplane into the vector space which separates the instances that belong to different classes. The plane is fitted such that the accumulated distance between the closest instances (support vectors) and the plane is as high as possible to ensure a clear distinction between the classes. In cases where the vectors are not linearly separable, the vector space is transformed into a higher dimensional space by the so-called *kernel trick*. In this higher dimension, a hyperplane can be fitted again, while this was not possible in the lower dimension. An interested reader is pointed to [5] for thorough information about SVMs. We applied the SVM implementation in the data mining software collection *Weka*[8] in version 3.5.8. It is highly modular and provides a wide range of possibilities for transforming, evaluating, and visualizing data.

To successfully apply the SVM, some parameters have to be optimized first. The choice of these parameters strongly influences the quality of the results. Using a script, we have evaluated and optimized following parameters. We obtained the best results using a radial basis function (RBF) kernel with parameters $C = 2^{17}$ (cost of errors if no perfectly separating hyperplane can be found: the vectors which are on the wrong side of the plane are multiplied by this factor) and $\gamma = 2^{-19}$ (kernel parameter that determines the smoothness

[7] `http://archives.seul.org/or/dev/Dec-2008/msg00005.html`

[8] `http://www.cs.waikato.ac.nz/ml/weka/`

of the hyperplane, i.e., size of the region influenced by a support vector). Note that the chosen parameters led to improved results on all datasets including open- and closed-world as well as Tor and JAP. The largest computational effort (which takes several days on AMD Athlon 64 3000+ CPU) was in the tuning of the features and optimization of the SVM parameters. Due to large and representative number of websites in our datasets we assume that the found parameters are good in general. Therefore, for new webpages we only need to train the SVM only once for the found parameters (which takes about two hours). The final classification (recognition of webpages) takes only few milliseconds.

5. EXPERIMENTS AND RESULTS

In this section, we evaluate the introduced method for WFP (see Section 4). The section starts with a short description of the experiments' setup in Section 5.1 followed by a presentation of the results. Section 5.2.1 will summarize the algorithm's superior performance to state-of-the-art works [11] in the closed-world setting. This part includes a detailed analysis of the features. Section 5.2.2 will apply the algorithm to the harder open-world scenario. We once more achieve alarming detection rates motivating the need for additional countermeasures for anonymization networks as discussed in Section 6.

5.1 Experiments

Section 3 introduces the Open-World and Closed-World Dataset that will be used in the following in order to evaluate the proposed algorithm and features. Before the classifier can be applied and evaluated, the given data sets have to be split into training and test data. For the 775 different web pages of the Closed-World Dataset, we use *cross-validation* in order to obtain representative and comprehensive test and training sets based on data set in Section 3. Cross-validation is a common method employed in data mining. It is often used for small data sets and works as follows: the data is split into n evenly large parts, the *folds*. Then, the entire process of training and testing is repeated n times, using one of the n folds as test data and the remaining $n-1$ folds as training data in turn. The results are averaged and therefore more solid and meaningful. In this paper, n is set to 10 with 2 instances per fold and per class. Additionally, the so-called *stratification* is used, which ensures that the instances within a class are represented as balanced as possible in each fold. The entire procedure is called *ten-fold stratified cross-validation*. If not stated otherwise, we used 18 instances for each censored page as training and 2 for testing applying our algorithm with support vector classification on the proposed features.

For the Open-World Dataset, an application of cross-validation was not necessary due to a sufficient amount of data. Instead, we collect enough instances for training and test data in advance and give those data separately to the classifier. For each of the five censored web pages and for Sexually Explicit, Alexa Top Ranked and Alexa Random (see Section 3), we use 35 instances as training and 25 disjunct instances as test. For uncensored web pages, we use the list of the 1,000,000 most popular Internet pages from the web statistics service *Alexa*. Out of this list, we randomly choose 4,000 uncensored URLs and include 1 instance each to the training data. Furthermore, we include 1,000 random uncensored URLs (disjunct from training) with one instance to the test data.

We additionally increase the classification's complexity by a disjoint definition of the training and test sets for uncensored web pages. Note that the recognition task is much harder since instances of the web pages in the test phase have never been seen by the system in advance. This is more realistic because the attacker usually does not know what other common web pages the victim usually accesses (see Section 3 for more details). We repeated the measurements on the Open-World Dataset 20 times. In each of these runs, the uncensored pages are randomly chosen, while the censored pages remain the same (we only replace the instances of censored pages in every run). We performed the recognition using the SVM classifier with enabled additional features and plotted averaged results including 95% confidence intervals.

As mentioned before, both the feature tuning as well as optimizing the SVM parameters is done on a disjoint data set. For neither the open- nor the closed-world setting, test data is included during training. This avoids that learned features may perfectly fit a specific test data set by accident (over-fitting), but generally be not representative for the considered classification problem. The next step applies the support vector machine with our designed features as introduced in Section 4. The SVM determines a score for each test instance with respect to all classes and assigns the test instance to the class with the highest score. To compare ourselves with Herrmann et al. [11] on the Closed-World Dataset, we determine the detection rate as the percentage of correctly classified instances divided by all instances. On the Open-World Dataset, it is not only important what percentage of censored web pages can be classified, but also how many uncensored web pages will be classified as censored. In our motivating example (see Section 1), innocents would be wrongly accused. Therefore, we calculate the true positive rate as the percentage of correctly classified censored instances divided by all censored instances. Additionally, we provide the false positive rate as the percentage of wrongly classified uncensored instances divided by all uncensored instances. The following section summarizes the results for closed- and open-world scenario.

5.2 Results

This section first evaluates the proposed algorithm on the Closed-World Dataset. Particularly we show that we achieve the recognition rates of 54.61% on Tor and 80% on JAP which are much higher than previously thought [11]. We then consider the more difficult Open-World Dataset. The algorithm is robust enough that even in this case, we can achieve alarming results of a true positive rate of up to 73%.

5.2.1 Results on Closed-World Dataset

Using Bayes Nets, Herrmann et. al. [11] achieve a recognition rate of 3% on the Tor Closed-World Dataset (see Section 3.1). We have implemented the algorithm of Herrmann as described in [11] and achieve similar results on the dataset. A recognition rate of less than 3% might appear non-threatening for many users. Figure 1 provides a visual impression of our recognition results and gives an overview of the features' impact on the detection rate. In this paper, we propose to use Support Vector Machines (SVM) instead of Bayes Nets as the first major expansion. The more power-

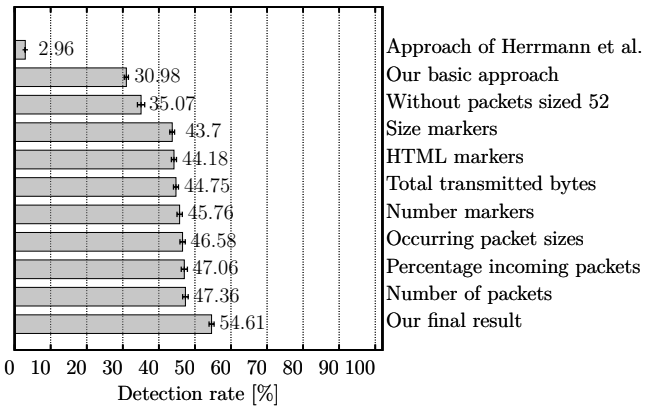

Figure 1: Influence of additional features on the detection rates for the Tor Closed-World Dataset

ful SVMs directly lead to an improved recognition of almost 31% (annotated as *Our basic approach*). By considering the proposed features (see Section 4.1), we increase the recognition rate to 47.36%, an increase by 17% (annotated as *Number of packets*, named after the feature that leads to this result). Note that the results for single features are shown in a cumulative way, i.e., the follow up features include all the prior features. The itemized list of the results for each feature indicates that each feature is important and contributes to the final recognition rate.

During the analysis of the results, we discovered that the Closed-World Dataset used by Herrmann contained several web pages that had a redirect to another URL. The practical problem with redirects is that between the loading processes of the original and the redirected page there is a small break. During this break, the browser displays a page with the information about the pending redirection and considers the web page as loaded. If the Chickenfoot script checks the value of the loading property at this point in time, the web page is regarded as loaded and the next URL is retrieved. Clearly, this leads to a corrupt data set, because the target web page has not been loaded yet. Hence, we disregard web pages with a redirect statement and add the URLs of the final web pages instead. It also turned out that there were a few cases of two different URLs redirecting to the same page. Clearly, this only disturbs the detection accuracy as practically there is no difference between the sites. Hence, double entries are removed from the list.

Additionally, we found instances of URLs that were not completely loaded even though Firefox regarded the web page as completely loaded. Triggering successful loading of web pages is vital as incomplete pages and errors in page loading lead to a deterioration of the detection rates. Automated detection of this property, however, is not trivial: sometimes pages have the status *loaded*, but remain empty. Hence, only pages with adequate size are considered. Practically, we initiat the reload of a page if its size changed by more than 20% from its mean size. Using these improvements we obtain our final result of 54.61%.

The proposed algorithm improves the previously known detection rate of Herrmann by more than 51%. A detection rate of almost 55 % on average is alarming. The anonymization using Tor is less robust as previously assumed. This indicates the need to take appropriate countermeasures.

The analysis of the detection results reveals that there are certain pages which are correctly classified in the majority of cases, whereas there are others which are hardly ever recognized by the classifier. Figure 2 shows this ratio in form of two complementary cumulative distribution functions (CCDF) for the experiments in Tor.

The blue squared curve shows the distribution of correctly classified instances (true positives). For 91% of all web pages, at least five instances were correctly classified. Instances of four pages only – about 0.5% – were never correctly classified. The results suggest that for almost all pages, at least some instances were correctly classified. This fact further illustrates a user's risk who relies on the anonymity when using Tor. The brown crossed curve shows the number of instances that were wrongly assigned to each class as the respective share of web pages (false positives). At least ten wrong instances were erroneously assigned to 47% of all web pages. 20 or more wrong instances were assigned to only a few pages (less than 5%), whereas only one page stands out with more than 60 wrongly assigned instances. Those numbers prove that the chosen set of URLs is rather representative and realistic. We do not expect much differences for an alternate set of URLs.

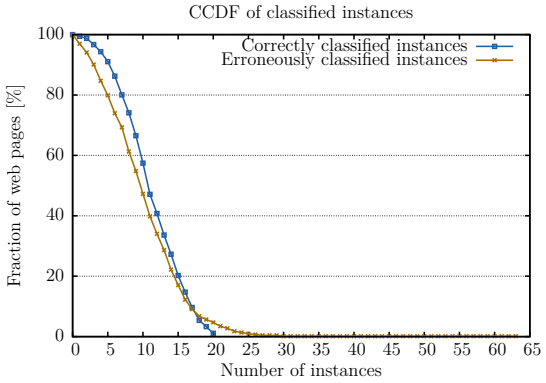

Figure 2: Complementary cumulative distribution functions (CCDF) of correctly and erroneously classified instances for the Tor Closed-World Dataset

We also applied our improved methods for website fingerprinting on the JAP Closed-World Dataset (see Section 3.1). The results are visualized in Figure 3. Herrmann et al. [11] achieved a detection rate of 20% in the JAP network. Their study is limited to a premium cascade only (see Section 3). Using Bayes Nets, we achieve similar results of 26% in premium cascade. With our algorithm, we boost the detection rate to almost 80%. Our results confirm that the JAP framework also does not provide the anonymity as previously thought. In addition, we investigated the accuracy of recognition in free cascades. Using Bayes Nets, we achieve 3.5% only. The application of our approach leads to a recognition rate of 60%. The overall result is surprising because JAP operators claim that premium cascades (paid) offer more anonymity to their users than free cascades. The surprising effect can be explained by less users in premium cascades due to higher costs. This fact leads to a lower workload. As further investigations experiments confirm, less workload directly leads to an increase in recognition rate of about 30%.

For both cascade types in JAP, we achieve higher detec-

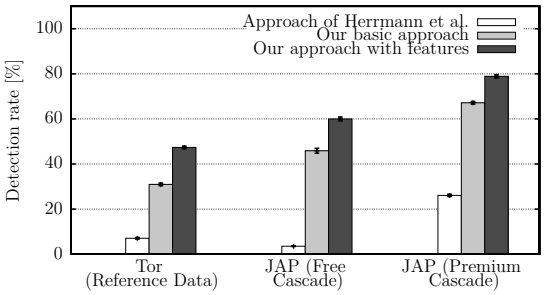

Figure 3: Detection rates for the JAP Closed-World Dataset

tion rates compared to Tor. Therefore JAP seems to be more vulnerable against fingerprinting in the considered scenario. Similar to Tor, the selected features play an important role for the high detection rates. Both for premium and free cascades, the features boost the results by at least 11%. The high recognition rates on the Closed-World Dataset confirm the generality and importance of the selected features and our method for the whole domain of website fingerprinting. In future work we plan to further explore features and alternative rounding increments to increase the recognition rates. This exploration provides the basis for the development of additional countermeasures as a prerequisite for an increased anonymity of Tor and JAP.

5.2.2 Results on Open-World Dataset

This section summarizes alarming results of our algorithm on the three introduced Open-World Datasets, namely *Sexually Explicit*, *Alexa Top Ranked*, and *Alexa Random* (see Section 3.2). In a first experiment (*Experiment 1*), we fix the number of censored pages to 5 and use 35 training instances each (in future work we will extend our experiment to include more URLs in the set of censored web pages). For the uncensored web pages, we use one instance for each of 4,000 random URLs from the Alexa statistics for training (Experiments 2–4 at the end of this section justify the selection of these numbers). For the testing phase, we consider 1,000 URLs that differ from the 4,000 URLs used for training. Hence, the classifier is not trained on the URLs that are used in the test phase (worst-case and challenging scenario for the attacker).

Page Set	True Positives	False Positives
Sexually explicit	56.0%	0.89%
Alexa top ranked	73.0%	0.05%
Alexa random	56.5%	0.23%

Table 1: True and false positive rate for Sexually explicit, Alexa top ranked and Alexa random of the Open-World Dataset

Table 1 shows the average of 20 runs for the three introduced data sets. For each, the true positive rate is higher than 56% for a false positive rate of less then 1%. For Alexa Top Ranked, we even achieve a true positive rate of 73% for a false positive rate of only 0.05%. These results are alarming because anonymization networks such as Tor do not provide sufficient anonymity in the open-world setting with respect to a local attacker – one of the weakest attacker in the threat model of low-latency anonymization techniques.

Apparently, the top ranked pages can be distinguished more easily than the pages from the two other sets. One indicator could be the fact that `google.com` is contained in the set of top pages, which is an exceptionally small page and therefore easy distinguishable.

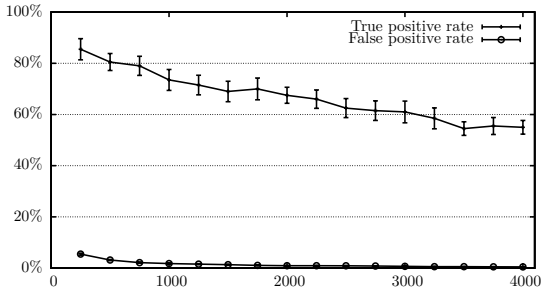

Figure 4: Influence of growing number of uncensored web pages on the true and false positive rates in the Open-World Setting

A second experiment (*Experiment 2*) shows how the number of uncensored URLs used for training influences true positive and false positive rates. Here, the number of censored pages is fixed to 5 URLs with 20 training and 2 testing instances each. Figure 4 suggests that at least 2,000 uncensored instances have to be used in the training phase to achieve a false positive rate of less than 1%. For a training on 4,000 instances, the false positive rate even drops below 0.5%. However, this low rates are achieved at the expense of falling detection (true positive) rates. Whereas for 2,000 instances, 67.5% of the censored instances are classified correctly, this number decreases to 55.5% for 4,000 instances.

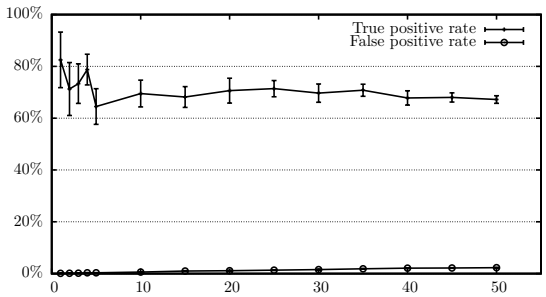

Figure 5: Influence of growing number of censored web pages on the true and false positive rates in the Open-World Setting

A third experiment (*Experiment 3*) analyzes how the number of censored URLs influences the true and false positive rates. To this end, we randomly selected censored URLs with 35 training instances each from the whole set of Alexa URLs. For training of uncensored pages, we randomly used 4,000 URLs with one instance each. As Figure 5 shows, a higher number of censored pages entails increasing false positive rates. This is because more censored pages increase the probability of confusion. The false positive rate does significantly increase from 0.08% for one censored page to 2.27% for 50 of them. Therefore, a high number of censored pages can lead to more confusion of the classifier thereby

ensuring a greater anonymity. Interestingly, the number of censored pages has only a minor impact on the true positive rate. The confidence interval for only a few censored web pages is rather high because in this case the result is easily influenced by outliers. For higher number of web pages, the mean and covariance values become more representative.

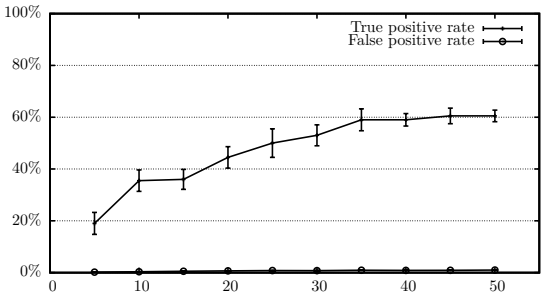

Figure 6: Influence of growing number of instances of censored web pages on the true and false positive rates in the Open-World Setting

A fourth experiment (*Experiment 4*) evaluates how the number of censored instances used for training influences the result (see Figure 6). In this experiment, the number of censored URLs is fixed to 5. The number of uncensored URLs used for training is fixed to 4,000 with one instance each. The results clearly prove that a higher number of training instances leads to a better true positive rate. If more than 35 instances are used, the true positive rate converges, and hence 35 is a reasonable choice of censored training instances. The false positive rate slightly increases for a higher number of censored training instances. In this setting, the false positive rate amounts to less than 1% for 35 censored training instances.

The presented results for the Closed-World and Open-World Datasets show that an attacker can spoil the anonymity of Tor and JAP through careful selection of parameters and training data. Using the proposed algorithm, website fingerprinting can be mounted by a local observer (e.g., an ISP of a victim) – one of the weakest attackers low-latency anonymization networks deal with. Since many users on anonymization networks rely and strongly depend on the provided anonymization, researchers in the security domain must identify adequate countermeasures. The next section summarizes preliminary results of camouflage as a successful countermeasure against fingerprinting attacks in anonymization networks.

6. COUNTERMEASURES

The previous Section 5 summarizes the results of the proposed algorithm on the Closed-World Dataset and Open-World Dataset. In both cases, the given anonymity is much lower than expected. Obviously, the results reinforce the necessity for adequate countermeasures.

Padding is the most common technique which is employed to impede traffic analysis. The sender appends a certain amount of dummy data to the actual application data before performing the encryption. Typically, padding is performed either to achieve a fixed packet size or a random one. Even though padding entails a slight decrease of detection rates (compare our results on Tor and JAP that use padding to ssh tunnels without padding [11]), the given rate in Tor and JAP is still too high to provide sufficient anonymity rising the necessity for more sophisticated countermeasures. In this section, we describe preliminary work on using camouflage as a more successful countermeasure. Camouflage is a smarter way to intentionally change the patterns of the data traffic. In this work, we load a randomly chosen web page (background page) simultaneously with the actually requested one. In doing so, we obfuscate the traffic by loading several pages simultaneously. This extension is used in both training and testing (no background page during training leads to worse detection rate). The suggested approach is easy to implement and can be used without any modification to the anonymization services Tor and JAP. Practically, the protection can be realized by, e.g., a web browser plug-in that, whenever the user requests a page, automatically loads an additional randomly chosen page in the background. Another great advantage is that this protection can be employed solely on the user side without any modifications to the anonymization network.

We tested camouflage for real Tor and JAP networks on the Countermeasures Dataset as introduced in Section 3.3. Keep in mind that due to an easier classification, a successful application of camouflage on the Closed-World Dataset will also lead to a better anonymity on the Open-World Dataset. The results are illustrated in Figure 7. In both networks, the overhead between anonymization as it is and with camouflage is about 85%. The effect of this simple countermeasure is very notable. For JAP, in which we used a free cascade for the tests, the detection rate decreases from 60% to 4% and for Tor it drops from 54% to about 3%. It should be clear that camouflage greatly reduces the features' significance because descriptive statistics of simultaneously loaded pages are mixed.

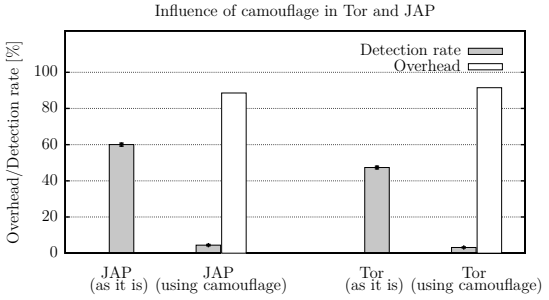

Figure 7: Effect of camouflage as countermeasure in the Tor and JAP Closed-World Datasets

Although the detection rate is on average about 3%, the classification of some web pages might still be easy. Successful countermeasures should decline the detection rates of all web pages to a level that is almost similar to random guess and at the same time cause only little performance losses. Clearly, the additional traffic generated by already one background page leads to a serious confusion of the classifier. We expect even better obfuscation for additional background pages as it will be more challenging for the attacker to extract the original statistics from the merged packets. Still, it has to be explored whether more sophisticated statistical measures can achieve this extraction. In future, we plan to analyze this specific effect of camouflage on diverse web

pages in detail. In the end, a user can decide how much overhead he will accept for the sake of a higher anonymity. However, the detection rate heavily depends on the loaded web page. In future we plan to provide tools to facilitate an assessment of anonymity per web page.

7. CONCLUSION AND FUTURE WORK

In this paper we showed that anonymity in Tor and JAP is not as strong as expected so far and cannot resist *website fingerprinting* attacks under certain circumstances. We first defined features for website fingerprinting solely based on volume, time, and direction of the traffic. As a result, the subsequent classification became much easier. We applied support vector machines with the introduced features. We were able to improve recognition results of existing works on a given state-of-the-art dataset in Tor from 3% to 55% and in JAP from 20% to 80%.

In a next step, we transferred our findings to the more complex and realistic open-world scenario, i.e., recognition of several websites in a set of thousands of random unknown (not previously seen) websites. To the best of our knowledge, this work is the first successful attack in the open-world scenario. We achieved a surprisingly high true positive rate of up to 73% for a false positive rate of 0.05%.

Finally, we showed preliminary results of a proof-of-concept implementation that applies camouflage as a countermeasure to hamper the fingerprinting attack in the open- and closed-world settings. For JAP, the detection rate decreased to 4% and for Tor it dropped to about 3%. Our camouflage strategy can be adapted without any changes in the underlying anonymization protocol. This countermeasure can be simply implemented in the form of a browser plug-in.

Future work will include additional feature selection to further boost the quality of recognition, study the influence of enabled active content, as well as consider clicking on embedded links. Moreover, we will provide an analysis of recognition results for specific and single web pages in addition to the given average results. Even if standard countermeasures are applied, it is possible that some web pages can easily be recognized. We aim to empower users with tools that would provide the feedback about the level of anonymity per web page. The countermeasures could then be dynamically adjusted based on this information. Therewith it would be possible to provide adequate anonymity for every possible web page.

Acknowledgements

Parts of this work have been funded by the National Research Found (FNR) of Luxembourg within the CORE grant project MOVE and by the EU FP7 IP OUTSMART.

8. REFERENCES

[1] Jap anonymity and privacy. `http://anon.inf.tu-dresden.de`.

[2] O. Berthold, H. Federrath, and S. Köpsell. Web MIXes: A System for Anonymous and Unobservable Internet Access. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*, pages 115–129. Springer-Verlag, July 2000.

[3] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine. Privacy Vulnerabilities in Encrypted HTTP Streams. In G. Danezis and D. Martin, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2005)*, volume 3856 of *Lecture Notes in Computer Science*, pages 1–11. Springer-Verlag, June 2005.

[4] H. Cheng and R. Avnur. Traffic analysis of SSL encrypted Web browsing. Project paper, University of Berkeley. Available at `http://www.cs.berkeley.edu/~daw/teaching/cs261-f98/projects/final-reports/ronathan-heyning.ps`, Dec. 1998.

[5] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Mar. 2000.

[6] G. Danezis. The traffic analysis of continuous-time mixes. In *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*, volume 3424 of *LNCS*, pages 35–50, May 2004.

[7] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, pages 303–320. USENIX Association, Aug. 2004.

[8] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol: HTTP/1.1. Internet Engineering Task Force: RFC 2616, June 1999.

[9] T. G, Abbott, K. J. Lai, M. R. Lieberman, and E. C. Price. Browser-based attacks on Tor. In N. Borisov and P. Golle, editors, *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*, Ottawa, Canada, June 2007. Springer.

[10] X. Gong, N. Kiyavash, and N. Borisov. Fingerprinting websites using remote traffic analysis. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 684–686, New York, NY, USA, 2010. ACM.

[11] D. Herrmann, R. Wendolsky, and H. Federrath. Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier. In *CCSW '09: Proceedings of the 2009 ACM CCS Workshop on Cloud Computing Security*, pages 31–42, Chicago, Illinois, USA, Nov. 2009. ACM Press.

[12] A. Hintz. Fingerprinting websites using traffic analysis. In R. Dingledine and P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*, volume 2482 of *Lecture Notes in Computer Science*, pages 171–178. Springer-Verlag, Apr. 2002.

[13] G. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345. Morgan Kaufmann, Aug. 1995.

[14] M. Liberatore and B. N. Levine. Inferring the Source of Encrypted HTTP Connections. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006)*, pages 255–263, Alexandria, Virginia, USA, Oct. 2006. ACM Press.

[15] D. McCoy, K. Bauer, D. Grunwald, T. Kohno, and

D. Sicker. Shining Light in Dark Places: Understanding the Tor Network. In N. Borisov and I. Goldberg, editors, *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)*, pages 63–76, Leuven, Belgium, July 2008. Springer-Verlag.

[16] S. Mistry and B. Raman. Quantifying Traffic Analysis of Encrypted Web-Browsing. Project paper, University of Berkeley. Available at `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.5823&rep=rep1&type=pdf`, Dec. 1998.

[17] S. J. Murdoch and G. Danezis. Low-Cost Traffic Analysis of Tor. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 183–195. IEEE Computer Society Press, May 2005.

[18] S. J. Murdoch and P. Zielinski. Sampled Traffic Analysis by Internet-Exchange-Level Adversaries. In N. Borisov and P. Golle, editors, *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*, volume 4776 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, June 2007.

[19] L. Øverlier and P. Syverson. Locating hidden servers. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*. IEEE CS, May 2006.

[20] 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 2002 IEEE Symposium on Security and Privacy (IEEE S&P)*, pages 19–30. IEEE Computer Society Press, May 2002.

[21] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 2nd edition, 1979.

[22] V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition [in Russian]*. Nauka, 1974. (German Translation: Vapnik, Vladimir and Chervonenkis, Alexey: Theorie der Zeichenerkennung, Akademie-Verlag, 1979).

[23] D. Wagner and B. Schneier. Analysis of the SSL 3.0 protocol. In *Proceedings of the 2nd USENIX Workshop on Electronic Commerce (EC-96)*, pages 29–40. USENIX Association, Nov. 1996.

[24] C. Wright, S. Coull, and F. Monrose. Traffic Morphing: An Efficient Defense Against Statistical Traffic Analysis. In *Proceedings of the 14th Annual Network and Distributed Systems Symposium - NDSS '09*, San Diego, California, USA, Feb. 2009. The Internet Society.

[25] C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson. Language Identification of Encrypted VoIP Traffic: Alejandra y Roberto or Alice and Bob? In *Proceedings of the 16th Annual USENIX Security Symposium*, pages 43–54, Boston, MA, USA, Aug. 2007.

[26] Wright, C.V. and Ballard, L. and Coull, S.E. and Monrose, F. and Masson, G.M. Spot Me if You Can: Uncovering Spoken Phrases in Encrypted VoIP Conversations. In *In Proceedings of the 29th IEEE Symposium on Security and Privacy (IEEE S&P 2008)*, pages 35–49, Oakland, California, USA, May 2008.