Cluster Comput (2016) 19:427–438
DOI 10.1007/s10586-015-0502-4

Techniques and countermeasures of website/wireless traffic analysis and fingerprinting

Taher Ahmed Ghaleb[1]

Received: 3 August 2015 / Revised: 9 October 2015 / Accepted: 10 October 2015 / Published online: 24 October 2015

Abstract The behavior of a communication traffic may reveal some patterns (such as, packet size, packet direction, and inter-packet time, etc.) that can expose users' identities and their private interactions. Such information may not be concealed even if encrypting protocols have been employed, which gives traffic analysis attacks an opportunity to infer the identities of the visited websites by Internet users, or the applications being running in wireless networks. In response, defense schemes and anonymity networks endeavor to disguise traffic features in order to preserve user privacy. This paper reviews existing traffic analysis techniques along with their countermeasures, and categorizes them into two main domains: websites and wireless. In addition, we propose a unified traffic analysis process model compound of a set of layers that demonstrate the stages of traffic analysis techniques. Then, factors that can impact the fingerprinting accuracy are elaborated to show how can the change of such factors affect the success results of fingerprinting. Finally, we present various potential challenges that need to be considered when implementing and deploying real-world traffic analysis systems. A recommendation of a future research direction regarding the enhancement of fingerprinting success rates and fair evaluation of them is also introduced.

Keywords Anonymity · Defense scheme · Traffic analysis · Website fingerprinting · Wireless fingerprinting

✉ Taher Ahmed Ghaleb
g201106210@kfupm.edu.sa

[1] Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Main Campus, P.O. Box 8119, Dhahran 31261, Saudi Arabia

1 Introduction

Usually, Internet users are not aware of how their privacy is being protected from other parties. Instead, they transfer such issues to the concerned authorities (e.g., browsers, encrypting protocols, etc.) that can take the responsibility of protecting user identities and provide a safe web browsing. Such security providers do their best to conceal users' private information being transmitted through their machines. One solution that could be used to this end is to encrypt all the communication packets into hard-to-reveal ones [20]. However, there might be some other useful information to adversaries that can expose the traffic behavior and user identities. This kind of information may actually not be encrypted, even if the traffic content is encrypted. For instance, packet lengths, timings, directions, and sequence can be utilized by traffic analyzers (or attackers) to infer the identity of the corresponding packets [15].

Traffic analysis can be defined as the process of monitoring the behavior of a communication traffic for the sake of discovering useful patterns inside the transmitted packets [18]. Series of such patterns can later be matched by traffic analyzers in order to reveal useful and private information about packet sources and destinations. Although traffic analysis is considered as a negative behavior, it may sometimes be advantageous and be used positively. Adversaries or attackers usually analyze a traffic for malicious purposes, such as recognizing user identities to violate their private communications. On the other hand, organizations and institutions may need to conduct a traffic analysis for safety purposes, such as tracking employees and students interactions. Also, governments can do so for investigating cybercriminal activities [10].

In the literature, various studies have been conducted to evaluate traffic analysis techniques [2,6,14,16,25]. One of

the main limitations of such studies is that they were not comprehensive enough as they reviewed only particular fingerprinting techniques, especially those that are related to websites. On the other hand, the positive aspect of such studies is their experimental evaluations. Several experiments were conducted by these studies to evaluate multiple attacks in terms of cost, time, feasibility and accuracy under different parameters. However, results obtained from their experiments were based on variations of a limited number of parameters, such as packet features, or target protocols, etc. We believe that this kind of evaluation is not adequate for a fair evaluation of traffic analysis techniques as there exist other factors that can significantly impact the accuracies of fingerprinting classifiers.

This has motivated us to establish a broad study of techniques and countermeasures of traffic analysis, which was conducted in [11]. The main purpose of our previous work was to give a state-of-the-art overview of the traffic analysis techniques and countermeasures and to categorize t into two main domains: websites and wireless. In this paper, we enhance our previous study by introducing the following list of extensions:

- First, we introduce a unified traffic analysis process model that can be followed in the design and development of new fingerprinting techniques. Implementing a certain traffic analysis technique based on the proposed model would actually differ from one technique to another. Therefore, we show how this model can be implemented as a comprehensive framework that permits researchers to construct their own fingerprinting techniques based on their preferences, or even to investigate existing techniques on various design choices. Alternatives, such as packet sniffers, browsers, data, and machine learning classifiers can participate in assembling newer fingerprinting techniques, which in turn may lead to producing better results. Moreover, the framework would support conducting fingerprinting operations using different computing technologies (e.g., cluster computing).
- Secondly, we extensively elaborate the factors that can directly impact the accuracy of traffic analysis techniques. It is important to take these factors into consideration when a new traffic analysis technique is intended to be implemented, as every particular factor can totally change the obtained results.
- Finally, we provide a detailed presentation of the potential challenges that can stand against the real-world implementation and deployment of traffic analysis techniques. Some of the state-of-the-art techniques attempted to address some of these challenges while some others have been overlooked. This is because that some issues could ordinarily be addressed with the use of simulation-based experiments. Whereas, in real-world implementations, they should seriously be considered as they can lead to unaffordable problems. Therefore, this paper discusses these challenges and suggests some recommendations about how they can be addressed in the future.

The remainder of the paper is structured as follows. In Sect. 2, we present the state-of-the-art traffic analysis and fingerprinting techniques for both domains: websites and wireless. Then, countermeasures against traffic analysis are presented in Sect. 3. Section 4 introduces a unified model that describes the overall process of traffic analysis. After that, a list of factors that can impact the accuracy of any fingerprinting technique are proposed in Sect. 5. Section 6 discusses all our observations of the results of the reviewed fingerprinting techniques and countermeasures. Challenges of applying real-world fingerprinting systems are illustrated in Sect. 7. Finally, Sect. 8 concludes the paper and suggests the possible future work.

2 Fingerprinting techniques

The literature is full of techniques that endeavor to identify users' online activities by analyzing the traffic, obtaining useful patterns, and then fingerprinting user identities. Most of wireless and website fingerprinting techniques targeted certain protocols and used particular features of traffic packets. In addition, all fingerprinting techniques have dealt with this issue from a machine learning perspective. This means that statistical classifiers have been employed to train the machine recognizing users' identities from the collected traces.

2.1 Techniques for website fingerprinting

Diverse experiments have been conducted in the literature to demonstrate that existing traffic analysis countermeasures still need to be enhanced. One of the first attempts in attacking website traffic to recognize users' identities was introduced by Wagner and Schneier [23]. The SSL protocol uses the random padding method that works on the block cipher modes only, which allows ciphertexts reveal the lengths of plaintexts. This shortcoming has effectively been availed by Wagner and Schneier to infer the identities of the visited websites. As a result, they suggested that random-length padding should be applied to all cipher modes to resolve the SSL problem.

Sun et al. [20] also showed that encrypted communications still reveal a considerable amount of information about the content. Their attacking algorithm could identify the user visited websites from the length and number of HTTP web page objects downloaded while browsing. They collected around

100,000 web pages from various websites and then categorized them in order to build a signature database. After that, the scalar closeness metric was used to measure the similarities between every two signatures. Finally, the Jaccard coefficient similarity metric was utilized to identify the lengths of page objects. Their experiments were repeated with the change of the similarity threshold in order to minimize the false positive rates.

Another work was done by Bissias et al. [4] who kept collecting their data for around 1-year using Firefox linked with OpenSSH tunnel. Then, they selected two features of each HTTP trace, namely the size and inter-arrival time of the packets. The cross-correlation metric was used in their technique to recognize the similarity of two traces of different time gaps. In spite of their low success rates, their obtained results were reasonably good.

Liberatore and Levine [15] introduced two new techniques for fingerprinting websites' traffic using the Naïve Bayes (NB) classifier along with a density estimation. This was done by identifying two unencrypted features of the packet traces, namely the direction and size. Their experiments showed that the OpenSSH protocol needs to be more secure to block attackers from analyzing traffics passing through which. While they were studying certain characteristics and limitations of Tor, such as low latency, they believed that these defense mechanisms may suffer from their attacks as well. Similarly, Herrmann et al. [13] did the same experiments (with the same packet's features) as Liberatore and Levine, but with the use of the Multinomial Naïve Bayes (MNB) classifier. The key feature of Herrmann's approach was the utilization of the total frequency of packet features across all training vectors.

Panchenko et al. [17] improved the work conducted by Herrmann [13] with the employment of the Support Vector Machine (SVM) classifier. They applied SVM to Tor network and, after experimenting their technique, they observed that securing websites against fingerprinting attacks was not totally sufficient. Their classifier was also applied to close- and open-world websites with utilizing the volume, time and direction features of the packets for fingerprinting. Their work was one of the first successful attacks under open-world websites and resulted in a higher accuracy than others. Open-world fingerprinting attacks allow fingerprinting websites that are not listed in the traffic analyzer's database.

Shi et al. [19] proposed a technique for website fingerprinting that is able to analyze communication traffic over Tor. In their attack, incoming and outgoing packets were divided into several intervals and then converted into vectors. Similarities between observed vectors and well-known fingerprints were then be calculated using the *Cosine Similarity* formula. Their experimental results were theoretically and practically evaluated to show the effectiveness of their technique in degrading users' anonymity over Tor.

Cai et al. [7] proposed new fingerprinting methods for attacking websites. Data in their experiments were collected by capturing packets generated from visiting 100 pre-listed websites through Firefox associated with Tor using *Tshark*. Each of the 100 websites was visited 40 times to end up with 4000 packet traces. The Damerau–Levenshtein Distance (DLD) algorithm was used for identifying web pages. This classifier was able to recognize the sizes, ordering, directions and other useful information of the transmitted packets. In addition, they used Hidden Markov Models (HMMs) to identify which group of web pages relate to the same website. Their experiments demonstrated that current defense schemes against traffic analysis over Tor were still weak. As a result, they introduced an improved defense scheme called congestion–sensitive BuFLO. As demonstrated in their work, congestion–sensitive BuFLO could provide better security properties than previous ones (more details about this countermeasure can be found in Sect. 3.3).

Based on the fingerprinting attack proposed in [7], Wang and Goldberg [26] came up with an improved version of Cai's fingerprinting attack, called Combined OSAD. A new metric called Combined Optimal String Alignment Distance (Combined OSAD) was proposed to enable the identification of the similarities between two instances of traffic collected from Firefox with Tor. They could achieve higher accuracies by removing the SENDME packets from Tor cells. In addition, they introduced a new data collection methodology that can also be used by other techniques. Furthermore, they carried out experiments over open-world websites, like Panchenko [17], and could result in higher success rates.

In [24], a new fingerprinting attack was introduced by combining the attacks proposed in [7] and [26]. The resulting attacking technique promisingly achieved higher results than previous ones in terms of accuracy and processing time. Their experiments were conducted over large amounts of Open-World data gathered through Tor under realistic scenarios, in addition to several Closed-World experiments. Some of the well-known defense schemes, like Traffic Morphing and HTTPOS, failed to impact the accuracy of the proposed attack.

2.2 Techniques for wireless fingerprinting

On account of wireless communication links and the ease of snooping of WLANs, much of user traffic over WiFi links are exposed and can simply be analyzed by adversaries, even when the communication is encrypted. Adversaries can infer user's local and online activities by using specific patterns about the behavior of the packet being transmitted (e.g., size, direction, sequence, etc.). However, traffic analysis is restricted to a certain online application or service users are running. One of the challenges in such a process is

that users may run different activities in the same application (e.g., browsing, downloading, uploading, etc.). Another challenge happens when a user is simultaneously running more than one application (such as, chatting, network gaming, etc.), which complicates the extraction of traffic features due to the interference of applications' packets between each other.

Wright et al. [27] introduced a new method for identifying encoded VoIP calls. Through the packet sizes of the encrypted VoIP signals, they were able to identify the phrases spoken during call conversations. Their method used Hidden Markov Models (HMMs), which was trained using the TIMIT training data for the sake of recognizing 122 target sentences. By simulation, they created five conversations for every individual speaker in the test dataset, and then encoded them using *Wideband Speex*. Eventually, they applied HMMs to identify the occurrences of their target phrases from the lengths and signal noises of the packets being transmitted. They achieved promising results of more than 90 % accuracy.

Tavallaee et al. [21] proposed a hybrid mechanism for classifying network traffic, which initially applies a signature-based method and then several machine learning techniques. Over 250,000 flows were collected through IRANET, a large-scale network, and seven classifiers were applied to produce different results. Despite the high success rate obtained from the J48 Decision Tree classifier in a reasonable learning time, their approach failed to identify unknown running applications.

Another hierarchical classification-based system (using SVM and RBFN) was conducted by Zhang et al. [29] to infer the user's online activities. Their experiments were conducted on various wireless networks (home, university, and public) with diverse activity scenarios. Data were collected from an encrypted MAC-layer online traffic. Then, a set of classification operations were carried out using the data rate, packet count, and mean packet size features. With different snooping durations, they could obtain different accuracies. For example, in case of 5 s snooping duration, the accuracy was 80 %, while when lasted for 1 min, it was 90 %.

A recent technique was proposed by Atkinson et al. to infer the identity of wireless users whose communications are being transmitted through encrypted WiFi [1]. Their experiments involved running user applications (in particular, Skype and BitTorrent), and then collecting data by simulating user actions. The gathered packets were then filtered by the identification of a set of metrics, namely inter-arrival time and packet sizes. At last, aggregate normalized distributions were applied over each metric to extract the Skype voice as well as the BitTorrent and web traffic. Multiple accuracy scores were gained on a basis of the source of the collected packets (i.e., Skype, non-Skype, or web applications).

3 Protecting the communication traffic

Different defense schemes, or traffic analysis countermeasures, have been proposed in the literature in order to conceal, encrypt, and protect the private information inside traffic. These countermeasures have demonstrated an excellent success at the time they had been proposed. Nevertheless, researchers later on have proven that these schemes are breakable, and could easily analyze the packets coated through which.

3.1 Encrypting protocols

Encrypting protocols (such as, SSL, TLS, HTTPOS, IPsec, etc.) and tunnels (such as, SSH and OpenSSH) have been designed to protect internet communications by encrypting the entire traffic being passed through which. This kind of protection is considered as vulnerable since some information in the ciphertext itself are unconcealed and can reveal the traffic source or content. Therefore, the traffic could be analyzed by recognizing some useful information inside its behavior, such as packet size, count, direction, timing or sequencing. As a result, enhanced defense schemes have been proposed generally based on the previous schemes, in order to provide a better protection for the internet and wireless traffic with less overhead.

3.2 Anonymity networks

The *anonymity* terminology refers to anything without a name. In computer networks, it refers to the untraceable users' identities while they are browsing the web or running applications through wireless networks. Anonymity networks impede all attempts of traffic analysis and monitoring, or, at least, make them more complicated.

The Onion Router (Tor) [22] and Java Anonymous Proxy (JAP) [3] are well-known anonymity networks. They employ thousands of relay routers and proxies that can disguise the identity of their clients. The main objective of Tor and JAP is to provide privacy to network users, giving them a chance to escape inspection. Attackers, on the other hand, try to analyze Internet traffic using statistical techniques for the sake of recognizing the identities of victims by making use of certain patterns inside the traffic. Tor and JAP have shown outstanding effectiveness in reducing this kind of threats by spreading users' tracks over several cooperative relays, which in turn cover users' interactions so that no one can recognize packet sources.

3.3 Countermeasures against website fingerprinting

Dyer et al. [9] have conducted a comprehensive study of the countermeasures proposed in the literature against website

fingerprinting. Different experiments were accomplished on those countermeasures to investigate their ability to stand against several fingerprinting attacks. They showed that attackers can still recognize the identity of communication packets, and all previously proposed defenses failed to protect website traffic effectively. They also noticed that hiding useful information of the traffic, such as the bandwidth, total time, and direction, would enhance the effectiveness of such defense schemes. They ended up with a new defense scheme called Buffered Fixed-Length Obfuscator (BuFLO). However, since BuFLO sends all traffic packets in fixed sizes and inter-packet times, it resulted in a high bandwidth overhead.

Cai et al. [7] claimed that in addition to the bandwidth overhead problem, attackers are still able to identify which web page is visited by a victim in some configurations of BuFLO [9]. Therefore, they introduced the congestion–sensitive BuFLO defense scheme, which could resolve several issues of the original BuFLO (e.g., performance and security). The congestion–sensitive BuFLO operates by sending the communication packets in N cells. This is accomplished by monitoring the output queue every T milliseconds to either suspend sending packet cells when the output queue becomes full, or resume the transmission process when it is capable of transmitting more cells.

Wright et al. [28] introduced another defense scheme called Traffic Morphing (TM). TM aims to prevent traffic analysis algorithms based on optimization methods that can alter packet features to make them appear like they came from another web source. Both privacy and efficiency were supported by TM due to its capability of reducing time and bandwidth overheads that may be caused by other conventional strategies, which used to pad extra bytes to the packets. Their experiments showed the accuracy of some selected classifiers (like the one proposed in [15]) was effectively reduced after morphing the traffic.

A recent systematic analysis of existing attacks and defense schemes was conducted by Cai et al. [6]. Along with this, the study suggested a mathematical framework in which current fingerprinting attacks and defenses can be evaluated. In addition, they proposed a stronger, mathematically-designed defense scheme called Tamaraw. This scheme outperformed previously proposed schemes as it could reduce the time and bandwidth overheads, and provided a better protection than BuFLO by hiding all traffic features.

Recently, Cai et al. came up with another enhanced defense scheme called CS-BuFLO as a real-world defense solution for website fingerprinting [5]. Compared with HTTPOS, SSH, and Tor, CS-BuFLO conceals more information and could reduce the accuracies of well-known fingerprinting attacks, but with a higher bandwidth cost. In terms of bandwidth, CS-BuFLO seems to be similar to the original BuFLO in only two different experimental manners. This means that CS-BuFLO was empirically analyzed, while the original BuFLO was evaluated through simulation-based experiments.

3.4 Countermeasures against wireless fingerprinting

Greenstein et al. proposed SlyFi, "a wireless identifier-free link layer protocol" [12]. Their proposed protocol could preserve user's privacy by concealing their MAC addresses. This prevented attackers from recognizing traffic features in WiFi communications. A symmetric-key encryption is used in this protocol to encrypt the packets at the senders, while, at the reception end, a lookup-table with a symmetric-key decryption is required. The limitation of this approach is the overhead caused by the key management and encryption operations.

Zhang et al. [30] introduced a new technique called Traffic Reshaping. This technique reshapes the packets transmitted over wireless cards by transporting them through multiple virtually created MAC interfaces. Reshaping includes the change of the length of the packets, I/P time, and other features. This enhanced the privacy protection by preventing leaks of information without sacrificing bandwidth overhead. The performance of their technique was evaluated over traces collected from a WiFi network card and resulted in a better accuracy. Traffic reshaping proved its ability to defend against traffic analysis when it degraded the accuracy of an attack from 91.86 % down to 44.49 %.

4 Traffic analysis process model

Generally, traffic analysis can be realized as a process of intercepting network traffic. The main objective of such a process is to build a database of fingerprints that symbolize users' identities and activities for the sake of recognizing them later and, subsequently, break user's privacy. Therefore, traffic analysis has also been referred to by 'fingerprinting' to reflect its actual and main goal. Traffic analysis comprises a set of dependent operations. Some of these operations are usually used for preprocessing, while the others are for the actual processing. All such operations have the same ultimate goal, which is producing fingerprints that represent user interactions based on certain patterns in the traced packets. Packet length, direction, order, and timing are examples of the features that can be monitored in order to identify patterns among them and, eventually, construct corresponding fingerprints.

The following unified process model is introduced to direct future researchers about the required stages, operations, and tools for developing and implementing traffic analysis techniques. This model encompasses a set of layers

representing the key operations any fingerprinting technique should utilize. These layers also represent all necessary stages for carrying out a traffic analysis for any given communication traffic (i.e., packet traces collected from a website or wireless traffic). The sequence of these layers is important as each layer accepts the outcome of its preceding layer, processes them, and then feeds its following layer with the appropriate data.

4.1 Preparation

Packet sniffing tools (also known as packet sniffers) facilitate intercepting, capturing, visualizing, and logging network traffic (e.g., TCPDump, Tshark, Wireshark, etc.). With respect to traffic analysis, packet sniffers play a vital role in the preparation stage as they can trace the traffic of various types of networks. By default, packet sniffers capture all packets sent and received during web browsing or wireless connections. However, they provide several facilities that allow users configure the tool to work on a specific connection, protocol, and port. In addition, a particular or multiple network channel(s) can be selected with particular adapters.

Another action that should be taken into consideration is the termination of all applications that are using the target connection, except the ones prepared to capture traffic from which. This includes the deactivation of all the applications' plugins that may use that connection to allow producing clean traffic traces.

4.2 Packet filtering

A traffic analysis technique should employ a list of filtering rules that relief the entire process. Collected packets from a communication traffic should be filtered and grouped based on the features that are desired to be used by the traffic analyzer. In other words, packet information can be separated into different log files, each with a different feature of the packet. For example, one file may involve packet sizes while inter-packet times may be stored in another file. This isolation of packet features is very beneficial as it can group packets based on source addresses to build a strong background about the activities of each user. It is also favorable to keep information about each website in a separate file. This operation can be automated to allow the synchronous supply of the updated data to the next operations.

4.3 Database construction

Once the listening packet-sniffer accepts a newly incoming or outgoing packet received from/sent to different data sources, a database is triggered to categorize and store data sources (e.g., websites, web-pages, or applications) and to match them with the existent records in the database. If, for example, a visited website matches an existing record in the database, its signature is added to the existing record to build a lot of signatures that represent it. If not, the database will be instructed to create a new record storing the information about that data source, including the signature of the current visit. The more the signatures about a certain data source in the database, the better the possibility of fingerprinting it.

4.4 Collection strategy

While requests keep incoming from various data sources, the sniffer keeps collecting them a trace after another. Indeed, the amount of the collected traces would extremely be huge. Therefore, traffic analysis techniques should have a smart strategy for handling them thoroughly, but with a minimal storage overhead. To minimize the amount of the data collected, traces should properly be filtered according to the predefined features that would assist in the identification of certain behaviors. In other words, not every datum gathered from the sniffing tool is required to be collected and to be used for fingerprinting. This issue is crucial and should carefully be taken into account as excluding some information during the sniffing may negatively affect the final accuracy of fingerprinting techniques.

4.5 Preservation strategy

Since not all information in the raw traces collected by the sniffing tool are important, they may not be preserved in the database as preserving them would be a waste. Therefore, the traffic analysis technique should be aware of this issue and should only store the most important characteristics of the packet, which will later on be needed by the classifier. Notice that we might sometimes need to store raw traces in external storage devices for future investigation purposes. However, some data sources, such as news or social websites, periodically update their contents (e.g., per minutes). Repeating the entire process of collecting, filtering, storing and detecting at every change would consume too much CPU time as well as memory space. Therefore, techniques should be smart enough to deal with such sites by setting the clock to work at reasonable periods, hourly for example.

4.6 Building the classification model

It is necessary to properly configure traffic analysis techniques to have a maximum number of packets needed per each website/application. This number represents the adequate information required to train and build the clustering

model, which will be utilized for packet classification. Once the predetermined limit of traces is reached per a particular website/application, all packets incoming/outgoing from/to which should be discarded. For a better performance, a feedback channel between the fingerprinting model and sniffing tool has be established to allow refining the filtering rules of the sniffer to exclude the data sources that have reached their limits of the number of traces. This would also liberate the operations in-between these two stages from unwanted work.

4.7 Training and testing the fingerprinting classifier

In the literature, traffic analysis has been addressed as a classification problem using Machine Learning classifiers. Fingerprinting techniques need to repeatedly rebuild the classification model with the newly collected packet traces. Each class in the clustering model can either represent a particular website/application or a category of them (e.g., social, news, emails, etc.) or applications (e.g., chatting, games, downloaders, etc.). For each set of traffic traces, the classifier chooses the desired features (e.g., size, I/P time, direction, etc.) from them and then starts training its model using a considerable subset of the traces. After training, the rest subset of the traces can be used as test cases to measure the ability of the classifier in recognizing packets' identities. For example, among 40 visits of 'www.google.com', the classifier utilizes 36 of them for training the model and the rest 4 for testing. It is common in machine learning classifications to accomplish the training and testing in several folds, where the accuracy of each fold is independently calculated and, eventually, the ultimate accuracy is calculated by aggregating the accuracy values of the different folds.

5 Fingerprinting accuracy factors

5.1 Domains of communication traffic

Implementing a traffic analysis technique over website traffic is, for sure, different from doing so over a wireless traffic. The distinction is related to the different functionalities provided within each. While websites' traffic can have packets representing interactions being done within a numerous number of websites and web pages, wireless traffic can involve packets representing interactions done within a wide range of applications.

Possible domains Websites, Wireless, or both.

5.2 Network protocols and defenses

Selecting a target protocol for mounting a fingerprinting attack is important as it determines how secure the traffic is. If a traffic analysis attack targeted a non-secured protocol, such as HTTP, it would lead to promising results. However, mounting the same attack against more secure protocols, such as Tor, would significantly reduce its accuracy. Therefore, choosing more robust protocols directs to constructing robust techniques that never fails when applied to less secure ones.

Possible protocols HTTP, HTTPOS, SSL, TLS, SSH, OpenSSH, IPsec, Tor, JAP, Encrypted WiFi, etc. Plus, all defense schemes shown in Table 1.

5.3 Packet features

Traffic analysis techniques can recognize to which website or application a packet relates based on certain features and patterns inside them. Working on a single feature of

Table 1 Summary of traffic analysis countermeasures

Technique	Description
Traffic morphing [28]	Alters packet features to make them appear like they came from another web page
BuFLO [9]	Sends the whole communication packets in fixed sizes and fixed inter-packet times, which results in a high bandwidth overhead
Congestion–sensitive BuFLO [7]	Operates like the original BuFLO, but by sending communication packets in N cells (i.e., multiple patches) with less bandwidth overhead
Improved CS-BuFLO [5]	Randomizes the writing timings of a network with adding extra junk data. It transmits data in fixed-size chunks at semi-regular periods
Tamaraw [6]	Operates like the original BuFLO, but packets are sent in 750 bytes, and it deals with incoming and outgoing packets differently. Time and bandwidth overheads are significantly reduced
SlyFi [12]	Conceals the MAC address by encrypting the traffic, which is an overhead in addition to the overhead caused by the key management
Traffic reshaping [30]	Over each wireless card, a set of virtual MAC interfaces is created in order to schedule packets over the created interfaces dynamically, and features of each individual interface are then reshaped

the traffic does not always produce outstanding results. Therefore, a combination of several features can assist in building smart models that can match any trace of a particular website/application traffic to its corresponding class.

Possible features Packet size, count, direction, sequence, inter-packet time, noise.

5.4 Machine learning classifiers

The major part of any traffic analysis technique is the Machine Learning classifier. Fingerprinting accuracy can definitely be enhanced if an excellent classifier is employed. Selecting a classifier may depend on several criteria. For instance, if we are aware of speed, we should not go for slow classifiers like decision trees and k-Nearest Neighbors. Moreover, some classifiers are useful for huge amounts of data; but if the data are not that enormous, other classifiers, like SVM, would be favorable.

Possible Classifiers Support Vector Machine (SVM), Artificial Neural Networks (ANNs), Naïve Bayes (NB), Multinomial Naïve Bayes (MNB), Hidden Markov Model (HMM), k-Nearest Neighbors (kNN), decision trees, logistic regression, etc.

5.5 Ways of data collection

Most of traffic analysis techniques collect a big deal of data, filter them, and then process them offline. On the other hand, some other techniques analyze the traffic instantly (online). Certainly, a fingerprinting technique will not produce the same results if conducted in these different ways.

Possible data collection ways Closed-World, Open-World, Both.

5.6 Web browsers

During our study, we have noticed that most fingerprinting techniques have been evaluated under data collected from the Firefox web browser. Based on [31], the accuracy of fingerprinting techniques might be changed if evaluated under data collected from different web browsers, given that browsers might contribute in preserving user's privacy.

Possible web browsers Internet Explorer, Google Chrome, Firefox, Safari, Opera, etc.

5.7 Filtering rules

Packet filtering may also directly affect the accuracy of a fingerprinting technique, either positively or negatively. The reason behind this is that protocols may inject flags or even produce extra control packets to manage the traffic during communication. The existence of such extra information in the packets may disturb the classifier. Therefore, eliminating such information would lead to clean traces that exactly reflect the data traffic.

Excludable packets and control flags SENDME, ACK, SYN, NULL, etc.

6 Observations, analysis, and discussion

A summary of existing defense schemes against traffic analysis is demonstrated in Table 1. Table 2 summarizes the various traffic analysis techniques reviewed in this paper. Notice that we have omitted results related to the *False Positive Accuracy* attribute from that table as we could not find adequate information about them in the reviewed papers. Authors of these papers just claim that the false positive rates were reduced.

It can be observed from Table 2 that every traffic analysis technique was implemented using almost a different machine learning classifier from the others. In addition, some of the techniques were implemented under more than one classifier, but the most accurate one among others is listed in the comparison table. Essentially, the accuracy of each technique depends on the set of factors described in the previous section.

Fingerprinting techniques that were applied to Tor traffic may result in higher accuracies if applied to encrypting protocols like SSL. Likewise, techniques applied to encrypting protocols may also produce better accuracy if implemented on non-encrypting protocols like HTTP. In case of the technique introduced by Tavallaee et al. [21], it resulted in high accuracies when implemented on the data captured from an encrypted wireless network using the different 6 classifiers. Therefore, implementing those accurate classifiers on Tor traffic, might produce better accuracies as well.

During our study of the existing fingerprinting techniques, we have noticed that they are semi-automated (i.e., no tool support). They used to collect data using packet sniffers, then manually run another tool (or script) to filter the packets collected and eventually mount their attacks using different tools that are responsible for building the classification models. This procedure is not practical, especially when applied in a real-world environment. Therefore, it is recommended to build an integrated system that can encompass all operations and tools needed for traffic analysis.

With respect to data collection, various aspects should be investigated. First of all, the collected traces should cover different web browsers, including Firefox, Chrome, Internet Explorer, Safari, Opera, etc. In addition, web browsers running on smart and mobile phones, including Android and iOS, should be involved as well to investigate their impact on the traffic generated. Second, traces should address all secure routes a traffic might go over, including TLS, SSL, SSH, Tor, etc. Last, real-time open-world experiments should further be

Table 2 Summary of traffic analysis techniques

Technique	Domain	Target protocol	Features (Packets)	Classifier	Data processed	Reported accuracy
Sun et al. [20]	Websites	Http	Count, length	Jaccard coefficient similarity	Closed-World	75 %
Bissias et al. [4]	Websites	SSL	Length, I/P time	Cross correlation	Closed-World	40 %
Liberatore et al. [15]	Websites	OpenSSH	Length, direction	Naïve Bayes	Closed-World	90 %
Herrmann et al. [13]	Websites	Tor Network	Length, direction	Multinomial Naïve Bayes	Closed-World	2.96 %
Yi Shi et al. [19]	Websites	Tor Network	Length, direction	Cosine Similarity	Closed-World	50 %
Panchenko et al. [17]	Websites	SSH, TLS, IPSec RFCs,	Length, ordering + Total transmission bytes	Ad hoc SVM	Closed-World Open-World	82 %
Cai et al. [7]	Websites	Tor	Length, direction and ordering	DL-Distance with SVM	Closed-World	87.30 %
Wang and Goldberg [26]	Websites	Tor	Length, direction and ordering	Combined OSAD	Closed-World Open-World	91 %
Wang et al. [24]	Websites	Tor + defenses	Length, Ordering + total transmission size, time, and packets	k-Nearest Neighbour	Closed-World + Open-World	Diverse
Wright et al. [27]	Wireless	Encrypted VoIP	Noise, length	HMMs	Closed-World	90 %
Tavallaee et al. [21]	Wireless	Encrypted WiFi IRANET Network	Protocol, period, Flow & Packet Length, I/P time	J48 decision tree, 6 Others	Open-World	99.27 %
Zhang et al. [29]	Wireless	Encrypted IEEE 802.11	Data rate, Count, Mean Length	SVM + ANN	Open-World	83 %
Atkinson et al. [1]	Wireless	Encrypted IEEE 802.11	Length, I/P time	Normal distribution	Closed-World	76.56 %

improved to cope with the rapid spread of wireless applications and real-world websites and web pages, such as the massive creation of *Facebook* groups and pages).

Hybridizing multiple Machine Learning classifiers (or countermeasures) would lead to developing robust techniques and to enhancing the success results. In addition, applying classifiers of a certain domain (e.g., wireless) to other domains (e.g., websites) may contribute to better success rates in the future. This is because that some classifiers that were applied to wireless traffics have not been validated on websites traffic, to the best of our knowledge. Furthermore, complementing fingerprinting techniques with other useful techniques, such as network forensics, may produce robust models that can usefully and positively be incorporated by ISPs [10].

There is no adequate information in the literature on how to keep up-to-date with the advancement of security protocols like Tor to automatically refine and improve traffic analysis strategies. The state-of-the-art techniques used to transact with current situations of anonymity protocols without mentioning how things are going to be in the future, and what parts of traffic analysis techniques should be enhanced in case of new countermeasures have been developed. In addition, fingerprinting techniques have been considered as classification techniques. Therefore, extensive studies should be conducted in the future to investigate whether it can be reduced to another kind of problems.

7 Challenges of applying real-world traffic analysis systems

Most of the traffic analysis techniques proposed in the literature were evaluated based on either simulation experiments or empirical analysis. This indicates that building a real-world fingerprinting system would be accompanied by many obstacles and challenges. These challenges have to carefully be considered and addressed in the development of any real-world fingerprinting technique. Although most of the state-of-the-art techniques attempted to address some of these challenges, they overlooked some other aspects that can vitally impact their obtained results. There also exist other kinds of challenges for deploying real-world traffic analyzers, whereas, in simulation-based experiments, they were ordinary. For instance, extensive filtering rules of capturing traffic packets help in reducing space and processing overhead in later stages. This is a normal operation for any fingerprinting technique that relies on simulations. However, having them in real-world systems would consume more than expected processing time, which, consequently, would delay all traffic analysis operations.

7.1 Data collection and storage

Collecting data from various sources is a major challenge for real-world traffic analysis systems. Existing fingerprinting techniques are restricted to a set of common websites and wireless applications, and also a limited number of visits per each. For a real-world traffic analysis system, all visited websites (with all their web pages) and applications running on a wireless network (with all their different functionalities) should be considered. This would require maintaining a huge database that can handle signatures of all pages or functionalities available. The widespread construction of websites and their countless web pages in the world wide web would indeed complicate analyzing the traffic. Similarly, the variety of applications that can run via wireless networks and the persistent increase of their functionalities would also make packet collection and preservation more complex. Definitely, securing a database that handles all such stuff is a severe challenge.

One can benefit from the data collection strategy introduced in [24,26], which could effectively gather information from packets in realistic scenarios.

7.2 Fingerprinting real-world websites

Almost all fingerprinting techniques were evaluated under pre-listed websites or applications. For a comprehensive traffic analysis, real-world traffic analysis systems can be up-to-date with Internet Service Providers, which can supply them with the newly created websites. This allows them run various visits to these websites to collect packet traces, which in turn assists in building a primary signature history about them. This procedure can easily be carried out for the default pages of those websites, or at least the fundamental web pages of them. Nevertheless, it is not the case for websites that can have massive web pages to be created over and over. For instance, social websites, such as `facebook.com`, enable users create their own web pages. Therefore, a real-world fingerprinting system should employ sophisticated technologies that can cope with the tremendous construction of websites and web pages as well as the numerous versions of wireless applications.

Techniques introduced in [13,24,26] have demonstrated effectiveness towards simulating how fingerprinting techniques can recognize unlisted websites, web pages, or wireless applications.

7.3 Browsers variety

Internet browsers are diversified these days with too many versions of each. Each browser can have different policies regarding packet transmission, which can participate in protecting users' privacy. Expressly, traces generated by vis-

iting a website using a certain browser may have a different signature if another browser was used. This issue needs a deep investigation of how protection mechanisms employed by browsers work. This would help in fingerprinting the browsers in the early stages of analysis, and thereafter, websites can be analyzed based on the resultant browser.

Detailed analysis of browser fingerprinting can be found in [8,31]. As we have mentioned earlier, their techniques can be utilized to identify browsers that were used to transmit the packets and, subsequently, website fingerprinting systems can continue analyzing the packets based on the initial identification of the browser.

7.4 Processing time overhead

Traffic analysis and fingerprinting techniques always have issues with the time overhead required for processing traces. The main causes for this kind of overhead are the multiple dependent stages of any traffic analysis process, namely the data collection, preservation, filtering, and analysis. In regards to the analysis stage, Machine Learning classifiers consume a considerable amount of time to train the classification model and to normalize the traces collected. Normalization is sometimes needed because traces are not identical in size, even if they were collected from the same website/application.

This issue could be resolved by implementing the traffic analysis system on cluster or grid environments. This would help to decompose tasks into several sub-tasks, distribute them to a number of different nodes of the cluster, get the sub-results, and eventually aggregate them to produce the final results (i.e., parallel processing).

7.5 Limited identification

As presented earlier in this paper, traffic analyzers can identify Internet websites, web pages, downloaded files, user voice, etc. However, identifying such sample identities does not completely mean that we could recognize all user activities. Users may chat, post, like, listen to a music, watch a video, or do many other activities in the same web page or application. Hence, preserving fingerprints for all such activities would be more difficult for a fingerprinting system. Another challenge is related to the execution of multiple applications or websites at the same time, which leads to interfere packets of such various objects with each other.

8 Conclusion

This paper reviewed techniques and countermeasures for traffic analysis and fingerprinting, and categorized them into two main domains: websites and wireless. In our study, we demonstrated how current defense schemes, encrypting protocols, and anonymity networks did their best to harbor user private information. We have also shown how adversaries utilize all possibilities to analyze users' traffic for the sake of revealing their identities. In addition, a proposal of a unified model for traffic analysis was introduced with a detailed illustration of how each stage works. We also listed all the factors that can impact the accuracy of any fingerprinting technique, and we conducted a thorough comparison of the reviewed techniques showing how their accuracies vary with the change of such factors. Finally, we discussed a set of challenges may face the application and deployment of real-world traffic analysis systems.

As a future work, we recommend constructing an extensible framework that can aid in developing new traffic analysis techniques. In other words, such a framework would provide a toolbox of various strategies for data collection, filtering, preservation, and classification. The framework would also facilitate the change of different parameters and factors in order to reach better success rates. For instance, users can select the desired web browser, protocol, features, filtering rules, etc. In addition, users in such a framework would have the ability to extend the framework with their own classification algorithms given that existing ones do not fulfill their goals. We believe that having a framework with such characteristics would assist in a fair evaluation of current and future techniques since the implementation environment is going to be the same.

Acknowledgments The author would like to sincerely thank his home institution, Taiz University - Yemen, which donors him a scholarship to continue his graduate studies abroad.

References

1. Atkinson, J., Adetoye, O., Rio, M., Mitchell, J., Matich, G.: Your wifi is leaking: inferring user behaviour, encryption irrelevant. In: Wireless Communications and Networking Conference, pp. 1097–1102. IEEE (2013)
2. Back, A., Möller, U., Stiglic, A.: Traffic analysis attacks and trade-offs in anonymity providing systems. In: Information Hiding, pp. 245–257. Springer (2001)
3. Berthold, O., Federrath, H., Köpsell, S.: Web mixes: a system for anonymous and unobservable internet access. In: Designing Privacy Enhancing Technologies, pp. 115–129. Springer (2001)
4. Bissias, G.D., Liberatore, M., Jensen, D., Levine, B.N.: Privacy vulnerabilities in encrypted http streams. In: Privacy Enhancing Technologies, pp. 1–11. Springer (2006)
5. Cai, X., Nithyanand, R., Johnson, R.: Cs-buflo: a congestion sensitive website fingerprinting defense. In: Proceedings of the 13th Workshop on Privacy in the Electronic Society, pp. 121–130. ACM (2014)
6. Cai, X., Nithyanand, R., Wang, T., Johnson, R., Goldberg, I.: A systematic approach to developing and evaluating website fingerprinting defenses. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 227–238. ACM (2014)

7. Cai, X., Zhang, X.C., Joshi, B., Johnson, R.: Touching from a distance: website fingerprinting attacks and defenses. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 605–616. ACM (2012)
8. Chan-Tin, E., et al.: Identifying webbrowsers in encrypted communications. In: Proceedings of the 13th Workshop on Privacy in the Electronic Society, pp. 135–138. ACM (2014)
9. Dyer, K.P., Coull, S.E., Ristenpart, T., Shrimpton, T.: Peek-a-boo, i still see you: why efficient traffic analysis countermeasures fail. In: IEEE Symposium on Security and Privacy (SP), pp. 332–346 (2012)
10. Ghaleb, T.A.: Website fingerprinting as a cybercrime investigation model: role and challenges. In: First International Conference on Anti-Cybercrime (ICACC-2015), pp. 1–5. IEEE, In press (2015)
11. Ghaleb, T.A.: Wireless/website traffic analysis & fingerprinting: a survey of attacking techniques and countermeasures. In: International Conference on Cloud Computing (ICCC), pp. 1–7. IEEE (2015)
12. Greenstein, B., McCoy, D., Pang, J., Kohno, T., Seshan, S., Wetherall, D.: Improving wireless privacy with an identifier-free link layer protocol. In: Proceedings of the 6th International Conference on Mobile systems, Applications, and Services, pp. 40–53. ACM (2008)
13. Herrmann, D., Wendolsky, R., Federrath, H.: Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In: Proceedings of the 2009 ACM Workshop on Cloud Computing Security, pp. 31–42. ACM (2009)
14. Juarez, M., Afroz, S., Acar, G., Diaz, C., Greenstadt, R.: A critical evaluation of website fingerprinting attacks. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 263–274. ACM (2014)
15. Liberatore, M., Levine, B.N.: Inferring the source of encrypted http connections. In: Proceedings of the 13th ACM conference on Computer and communications security, pp. 255–263. ACM (2006)
16. Murdoch, S.J., Danezis, G.: Low-cost traffic analysis of tor. In: IEEE Symposium on Security and Privacy, pp. 183–195. IEEE (2005)
17. Panchenko, A., Niessen, L., Zinnen, A., Engel, T.: Website fingerprinting in onion routing based anonymization networks. In: Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society, pp. 103–114. ACM (2011)
18. Raymond, J.F.: Traffic analysis: Protocols, attacks, design issues, and open problems. In: Designing Privacy Enhancing Technologies, pp. 10–29. Springer (2001)
19. Shi, Y., Matsuura, K.: Fingerprinting attack on the tor anonymity system. In: Information and Communications Security, pp. 425–438. Springer (2009)
20. Sun, Q., Simon, D.R., Wang, Y.M., Russell, W., Padmanabhan, V.N., Qiu, L.: Statistical identification of encrypted web browsing traffic. In: Proceedings of IEEE Symposium on Security and Privacy, pp. 19–30. IEEE (2002)
21. Tavallaee, M., Lu, W., Ghorbani, A.A.: Online classification of network flows. In: Seventh Annual Conference on Communication Networks and Services Research , CNSR'09, pp. 78–85. IEEE (2009)
22. Tor project: anonymity online. https://www.torproject.org (Last visited: Oct 2014)
23. Wagner, D., Schneier, B.: Analysis of the ssl 3.0 protocol. In: Proceedings of the Second USENIX Workshop on Electronic Commerce, pp. 29–40 (1996)
24. Wang, T., Cai, X., Nithyanand, R., Johnson, R., Goldberg, I.: Effective attacks and provable defenses for website fingerprinting. In: Proceedings of the 23th USENIX Security Symposium (USENIX) (2014)
25. Wang, T., Goldberg, I.: Comparing website fingerprinting attacks and defenses. Tech. rep., Technical Report 2013-30, CACR, 2013. http://cacr.uwaterloo.ca/techreports/2013/cacr2013-30.pdf (2014)
26. Wang, T., Goldberg, I.: Improved website fingerprinting on tor. In: Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society, pp. 201–212. ACM (2013)
27. Wright, C.V., Ballard, L., Coull, S.E., Monrose, F., Masson, G.M.: Spot me if you can: uncovering spoken phrases in encrypted voip conversations. In: IEEE Symposium on Security and Privacy, SP 2008, pp. 35–49. IEEE (2008)
28. Wright, C.V., Coull, S.E., Monrose, F.: Traffic morphing: an efficient defense against statistical traffic analysis. In: NDSS (2009)
29. Zhang, F., He, W., Liu, X., Bridges, P.G.: Inferring users' online activities through traffic analysis. In: Proceedings of the fourth ACM Conference on Wireless Network Security, pp. 59–70. ACM (2011)
30. Zhang, F., He, W., Liu, X.: Defending against traffic analysis in wireless networks through traffic reshaping. In: 31st International Conference on Distributed Computing Systems, pp. 593–602. IEEE (2011)
31. Zhioua, S., Langar, M.: Traffic analysis of web browsers. In: Proceedings of the Formal Methods for Security Workshop (FMS 2014), pp. 20–33. CEUR Workshop Proceedings (2014)

Taher Ahmed Ghaleb is currently working on his Master thesis (in a topic related to program analysis, reverse engineering, and program comprehension) for the sake of holding the Masters degree in Information and Computer Science at KFUPM. He did his Bachelors in Information Technology from Taiz University, Yemen in 2008. His BS graduation project was related to telephony applications and IVR systems. After his graduation, he worked at Taiz University as a Teaching Assistant for three years. His domain experience includes databases, reverse engineering, program analysis, program comprehension, program visualization, programming languages, aspect-oriented programming, and extensible compilers.