2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI) | 978-1-6654-3875-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/IRI51335.2021.00044

HiddenText: Cross-Trace Website Fingerprinting over Encrypted Traffic

Jimmy Dani
Dept. of Electrical Engineering and Computer Science
University of Cincinnati
Cincinnati, OH, USA
danijy@mail.uc.edu

Boyang Wang
Dept. of Electrical Engineering and Computer Science
University of Cincinnati
Cincinnati, OH, USA
boyang.wang@uc.edu

***Abstract*—Website fingerprinting can infer which website a user visits in Tor networks by eavesdropping and analyzing encrypted traffic patterns. Recent attacks built upon deep neural networks can achieve more than 98% accuracy. To mitigate the privacy leakage under website fingerprinting, effective defenses, such as Walkie-Talkie, have been proposed, where the attack accuracy can be mitigated to 50% at most. In this paper, we propose a cross-trace website fingerprinting, which leverages the semantic correlation of the content of webpages across traffic traces to improve attack accuracy when existing defenses are enabled. Our experimental results on real-world datasets demonstrate that our proposed cross-trace website fingerprinting can completely defeat Walkie-Talkie, in which an attacker can still achieve more than 70% accuracy over defended data.**

***Index Terms*—Encrypted Traffic Analysis, Machine Learning, Privacy**

I. Introduction

Website fingerprinting [1]–[3] can infer which website a user visits by analyzing encrypted traffic patterns without decryption. It can be formulated as a supervised learning problem in machine learning. Recent studies [4]–[10] leveraging deep learning can achieve very high accuracy (e.g., 98%). To mitigate the privacy leakage under website fingerprinting attacks, many defenses [11]–[18] have been proposed. For instance, with Walkie-Talkie [14], which matches the traffic pattern of a sensitive website with the traffic pattern of a decoy website, an attacker's attack accuracy is reduced to 50% in theory, i.e., cannot distinguish a sensitive website from a decoy website based on encrypted traffic.

However, the existing studies in website fingerprinting only focus on attacking *single traffic traces*, where the prediction of a website is determined based on the traffic pattern of one trace independently. The correlation across multiple traces of a user has not been well investigated. The correlation across multiple traffic traces can be found easily in the real world. For instance, a user often visits multiple consecutive web pages, where the content of these web pages are semantically correlated to a specific topic (e.g., "covid testing"). This correlation can be utilized in an attack to identify decoy webpages as the decoy webpages generated by existing defenses (e.g., Walkie-Talkie) do not include the corresponding correlation.

In this paper, we propose a *cross-trace website fingerprinting attack*, which leverages the potential semantic correlation across traffic traces to compromise existing defenses. More specifically, our attack first leverages the existing attacks to output confidences/scores of top-m webpages of each single defended trace. Next, we add a semantic similarity evaluation component, which adjusts the confidences of top-m webpages based on the potential semantic correlations across multiple traffic traces. Any semantic similarity evaluation method, such as TF-IDF [19] or Word Mover's Distance [20], can be integrated into this attack. As a result, the true label of a traffic trace can still be learned even if a traffic trace is protected by existing defenses. Our main findings are summarized below:

- We collect a real-world dataset (100 web pages with 250 traces per webpage) from Tor to examine the performance of cross-trace website fingerprinting. Unlike previous studies, which select webpages based on Alexa top websites, we choose the webpages based on popular keywords searched in Google to capture/simulate the semantic correlations across multiple traffic traces.
- Our experimental results show that our cross-trace website fingerprinting can completely defeat Walkie-Talkie. Specifically, single-trace website fingerprinting can achieve 71.0% accuracy (with no defense) and only 50.4% accuracy (with Walkie-Talkie). With our cross-trace website fingerprinting, the attack accuracy regains to 72.5% (with Walkie-Talkie). In other words, Walkie-Talkie does not offer any protection under our cross-trace website fingerprinting.
- We examine four well-known semantic similarity evaluation methods, including TF-IDF [19], BERT [21], GloVe [22], and Word Mover's Distance [20] respectively, in our experiments. Our experimental results suggest that three of them, including TF-IDF, GloVe, and Word Mover's Distance, can defeat Walkie-Talkie in cross-trace website fingerprinting, where TF-IDF is the most effective one among the three.
- Our experimental results also show that the latest attacks [15], [16] based on adversarial examples, although currently cannot be completely implemented/integrated with Tor in the real-world as Walkie-Talkie, are robust under our cross-trace website fingerprinting attack.

Reproducibility. The source code and datasets of this study

978-1-6654-3875-9/21/$31.00 ©2021 IEEE
DOI 10.1109/IRI51335.2021.00044

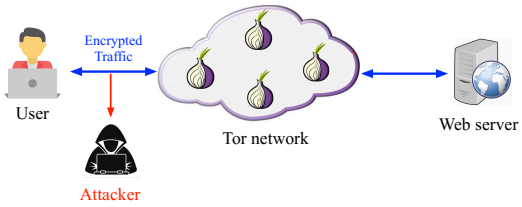

Fig. 1. The system and threat model.

are made publicly available [23] for others to reproduce our results.

II. Background

System and Threat Model. As illustrated in Fig. 1, we consider a system model including a user, an attacker, and a web server. This user visits the webserver through Tor networks. The network traffic between the user and a web server is encrypted. We assume there is an attacker eavesdropping on encrypted traffic between a user and the first Tor relay. This attacker aims to infer which website this user visits by analyzing encrypted traffic patterns. As in previous studies [2], [5]–[8], we assume that an attacker cannot decrypt packets. In addition, we assume that a user visits one website each time.

A *traffic trace* contains information of a sequence of incoming and outgoing network packets associated with one website visit. Given a traffic trace, we only keep the direction of each packet. We use +1 to represent an outgoing packet (to a website) and -1 to indicate an incoming packet (from a website). Each traffic trace is a vector of +1s and -1s. We also keep the same length for all the vectors by trimming or padding 0s at the end of each vector to feed them as inputs for deep neural networks.

Closed-World Setting. Website fingerprinting can be evaluated in two settings, including the closed-world setting and the open-world setting. We focus on a closed-world setting only in this paper. We defer the discussions on the challenges of examining the open-world setting in the cross-trace scenario in Sec. VI. In a closed-world setting, we assume that a user only visits a set of monitored websites and the attacker knows this set of monitored websites. Given an unlabeled traffic trace, an attacker infers which specific website it belongs to. As a result, the closed-world evaluation is formulated as multi-class classification. Accuracy is used as a metric to measure the attack performance in the closed-world evaluation.

III. Cross-Trace Website Fingerprinting

A. Main Idea

In this paper, we propose a cross-trace website fingerprinting attack, which can defeat existing defenses (especially the ones pairing a decoy webpage with a sensitive webpage) against single-trace website fingerprinting. The main idea of our proposed cross-trace website fingerprinting is to first output top-m predictions of each single (defended) trace with a classifier and then integrate semantic similarity evaluation

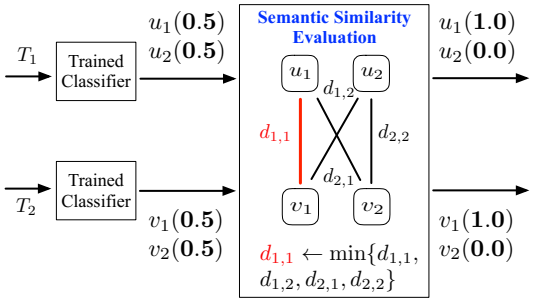

Fig. 2. The semantic similarity of two websites is denoted as $d_{i,j} = \texttt{distance}(u_i, v_j)$. If websites u_1 and v_1 have the minimal distance, where $d_{1,1} \leftarrow \texttt{min}\{d_{1,1}, d_{1,2}, d_{2,1}, d_{2,2}\}$, our cross-trace fingerprinting infers website u_1 as the label of traffice trace T_1 and website v_1 as the label of traffic trace T_2.

as an additional component. By calculating the semantic similarity of the content/text of these top-m predictions/webpages over different single traces, this additional component adjusts the confidences of the top-m predictions and jointly infers the true class for each trace.

The intuition behind our idea is that a user often visits multiple webpages consecutively, where the content/text of these webpages are semantically correlated. However, on the other hand, their decoy webpages generated by existing defenses are not semantically correlated. For example, a user can visit two consecutive web pages that are related to a topic (e.g., "U.S. presidential election") while the decoy (`espn.com`) of the first webpage and the decoy (`ieee.com`) of the second webpage are completely unrelated.

B. Details of Proposed Cross-Trace Website Fingerprinting

A high-level description is illustrated in Fig. 2. For ease of presentation, we assume 2 traces are evaluated jointly each time and top-m ($m = 2$) predictions of every single trace are examined with the semantic similarity evaluation component. We assume a state-of-the-art defense, named Walkie-Talkie [14], has been applied, where an attacker cannot achieve more than 50% accuracy (in theory) over every single trace.

As shown in Fig. 2, given two defended traffic traces T_1, T_2, our proposed cross-trace website fingerprinting works as follows.

- ***Step 1***: it outputs top-2 predictions of each trace (websites u_1, u_2 for T_1 and websites v_1, v_2 for T_2). Due to the existing defense, the classifier's confidence on each website in the top-2 predictions is at most 0.5.
- ***Step 2***: it evaluates the semantic similarity between website u_i and website v_j, where $i \in \{1, 2\}$ and $j \in \{1, 2\}$, and finds the pair of (u_i, v_j) with the *minimal distance in text similarity*. (i.e., $\texttt{argmin}_{i,j}\{d_{i,j}\}$).
- ***Step 3***: it adjusts the confidences of top-2 predictions of each trace based on the pair and outputs the top-1 prediction of each trace.

Note that the above process can be easily expanded to support the cases where the number of single traces is greater than

2 and the value of m (i.e., the number of top candidates) is greater than 2.

C. Semantic Similarity Evaluation Methods

In this study, we examine multiple existing semantic similarity evaluation methods, including TF-IDF (Term Frequency-Inverse Document Frequency) [19], BERT (Bidirectional Encoder Representations from Transformers) [21], GLoVE (Global Vectors for Word Representation) [22], and Word Mover's Distance (WMD) [20]. We select these four methods as they often perform well on evaluating the similarity of (long) text. We briefly explain each method below. Note that other semantic similarity evaluation methods can also be used in our cross-trace website fingerprinting as well as long as they can effectively computing semantic similarity over text.

TF-IDF. TF-IDF [19] is a common technique in information retrieval to measure the similarity of two documents. Each word is quantified and the weight of each word in a document is calculated. The weight is an indicator of a word's significance in a document. Term Frequency (TF) and Inverse Document Frequency (IDF) are the two essential elements of TF-IDF. The number of times a word appears in a document is measured by TF, and the importance of a word is measured by IDF. Given a document, TF-IDF transforms it into a vector. The distance/similarity of two documents can be computed via the cosine similarity of two corresponding vectors.

BERT. BERT [21] is a language model that is built upon bidirectional transformers and encoders. It can be fine-tuned for different purposes and contexts. In our study, a pre-trained BERT model is utilized to convert a text/document into a vector. The distance/similarity of two documents can be computed via the cosine similarity over their vectors. The model generates word embedding dynamically based on the context in which a specific word was used, rather than a predetermined representation of the word, resulting in a more accurate feature representation of the word.

GLoVe. Glove [22] is an effective algorithm for producing word embedding that uses statistical knowledge rather than context windows. Specifically, GLoVe collects global statistics of the words in a corpus in addition to local statistics. A co-occurrence matrix is used to determine the semantic association between words. For instance, the entry at i-*th* row and j-*th* column in the matrix decides how many times word *i* occurs with word *j*. A probability $p(i|j)$ is computed, which indicates the probability of occurrence of word in i-*th* row with the word in j-*th* column in the corpus. In this paper, a pre-trained GloVe model provided by [22] was leveraged to generate the embedding of the words. After obtaining vectors, cosine similarity is calculated to measure the distance of two vectors.

Word Mover's Distance (WMD). Given two documents, WMD [20] first obtains the embedded vector of each word in each document using word embeddings (e.g., word2vec). Next, WMD computes the distance between the two documents, which is the minimum distance that embedded vectors of one document need to travel to the embedded vectors of the other document using classic Earth Mover's distance transportation problem.

IV. Performance Evaluation

A. Data Collection Setting

In this study, we collect encrypted traffic in Tor to evaluate the performance of our proposed attack. Specifically, we leverage five virtual machines (with identical configuration) on campus to collect encrypted Tor traffic. Each virtual machine runs Ubuntu 18.04 with a 2.7GHz CPU and 4GB RAM. We use `tcpdump` to capture traffic and we utilize a tool, named `tor-browser-crawler` [24] to automatically visit each webpage over Tor browser (version 9.0.5). This tool was used in several previous studies [5] to collect Tor traffic. For each traffic trace, we collect the traffic for 60 seconds. A screenshot of a webpage was also captured along with a traffic trace. The screenshot was utilized later to verify whether a corresponding traffic trace is invalid due to reasons such as to request timeout, CAPTCHA verification, and browser crash. Invalid traces were removed.

Different from previous studies, in which only need to collect encrypted traffic, we also need to collect the text of each webpage to examine the semantic correlation across different webpages. Specifically, given a URL of a webpage, we leverage a `Python` package named `Beautiful-Soup` to automatically parse text from HTML and XML tags. These HTML tags are filtered out and only the text of a webpage was saved to a text file locally. We also manually verified the parsed text of each webpage. If the text was not parsed properly compared to the content of a webpage, we revisited the webpage and obtained the text manually.

B. Datasets

With the data collection setting described above, we collected two datasets of encrypted Tor traffic and we denote them as Alexa-100, and Keyword-100. An overview of the scales of the datasets is described in Table I.

TABLE I
AN OVERVIEW OF THE DATASETS.

	No. of Webpages	No. of Traces per webpage
Alexa-100	100	250
Keyword-100	100	250

Alexa-100. This dataset includes encrypted traffic traces of 100 websites, where each website (or class) has 250 traces. The 100 websites were selected from Alexa top-130 websites. Most of the previous research leveraged Alexa top websites to select websites/classes in the evaluation of website fingerprinting. The reason that we select 100 websites from top-130 websites is because some websites from the top-130 websites do not offer good quality of traffic data (e.g., some websites block Tor traffic).

Keyword-100. Different from previous studies, we investigate the semantic correlation of two websites/webpages of two consecutive encrypted traffic traces. However, selecting

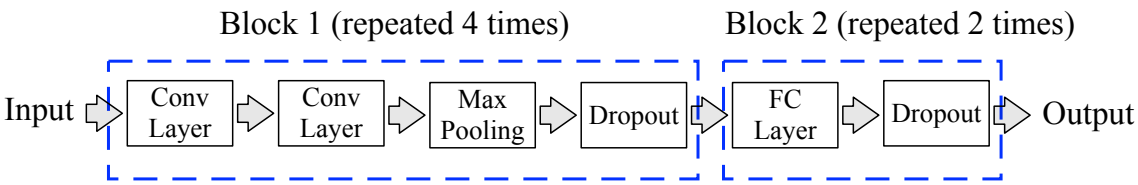

Fig. 3. The architecture of the DF model proposed in [5] (FC: fully-connect)

websites randomly from Alexa top websites does not result in semantic correlation among websites/classes. In other words, the way of selecting websites in previous studies is not suitable for our study in this paper. Therefore, we collect a Keyword-100 dataset to better simulate the semantic correlation across websites that could happen to real-world users.

In the Keyword-100 dataset, we select websites/classes related to popular keywords, which can help us simulate the case where a real-world user visits multiple different but content-correlated websites due to a keyword search. For example, in practice, when a user is interested in a specific subject/topic (e.g., "covid-19" or "presidential election"), it is very common that a user will first search the keyword by using a search engine online and then visit multiple websites from the search results to learn more detailed information about the subject/topic.

Specifically, we selected top-20 keywords in 2019 from Google Trends, which offers top keywords that people search using Google — the most popular search engine. Given each keyword from the top-20 keywords, we searched it with Google and then randomly selected five web pages from the top-50 search results provided by Google. This simulates the case that a real-world user visits web pages based on Google's search results but may not necessarily follow the order of these search results offered by Google. Overall, we obtained a list of 100 webpages (i.e., 5 webpage per keyword and 20 keywords) and we used these 100 webpages as the 100 classes in Keyword-100 dataset. For each webpage/class in Keyword-100, we collected 250 encrypted traffic traces in Tor. Traffic traces from the websites associated with the same keyword were randomly paired/grouped in our experiments to simulate the semantic correlation of two encrypted traffic traces. The list of 20 keywords that we used can be found in Appendix.

C. Experiment Setting

All the experiments were conducted on a Linux machine with Ubuntu 18.04, 2.8GHz CPU, 32GB RAM, and an NVIDIA Titan RTX GPU. Given a dataset, we use 80% for training, 10% for validation, and 10% for testing.

WF Classifier. We implemented a Convolutional Neural Network as a classifier to infer the label of each traffic trace. The architecture of this CNN is the DF model proposed in [5]. We choose the DF model as the classifier as it is currently one of the most effective models in website fingerprinting. The architecture of this DF model is highlighted in Fig. 3. It consists of two types of blocks, namely, Block 1 and Block 2. Block 1 is repeated four times. It consists of two convolutional layers and a pooling layer. Batch normalization is applied before the activation function at each convolutional layer. ReLU or ELU is selected as the activation function. Block 2 is repeated two times. It consists of a fully connected layer. Batch normalization is applied before the activation function at each fully connected layer. ReLU is selected as the activation function. We implemented the DF model with Tensorflow 2.0 and leveraged Microsoft NNI (Neural Network Intelligence) [25] to tune the hyperparameters.

Semantic Similarity Evaluation. We investigated four semantic similarity evaluation methods, including (1) TF-IDF with cosine similarity, (2) BERT embeddings with cosine similarity, (3) Word Movers Distance (WMD), and (4) GLoVe embeddings with cosine similarity.

We implemented each method in `Python`. The implementation of BERT embeddings [21] was based on `sentence-transformers` library and the implementation of WMD is based on `gensim` library. The GLoVe embeddings were generated using a pre-trained model available online [22]. The Cosine similarity between the two vectors generated using TF-IDF, BERT, and GLoVe was computed using `scikit-learn` library.

D. Closed-World Evaluation

Experiment A.1: Attack results on non-defended data over single traces. We first examine the attack results of website fingerprinting over single traces on non-defended data to demonstrate the results of website fingerprinting without defense. The attack accuracy was examined with the DF model as a classifier. The DF model was trained based on traces from each dataset, and the accuracy in this experiment was measured based on single traces as in previous studies without considering semantic correlation across traces. We examine the results over Alexa-100 and Keyword-100. In addition, we also validate the results over the DF-95 dataset from [5] to ensure we implement the DF model correctly.

TABLE II
ACCURACY ON NON-DEFENDED DATASETS (SINGLE TRACES).

	Training	Validation	Testing
Alexa-100	92.0%	86.1%	85.4%
Keyword-100	74.4%	71.3%	71.0%
DF-95 [5]	99.5%	97.7%	97.9%

As we can observe from Table II, the attack over the Alexa-100 dataset and DF-95 dataset achieve high accuracy with

85.4% and 97.9% in testing. These results are consistent with previous studies in website fingerprinting over single traces. This indicates that the CNN is implemented and tuned properly for website fingerprinting.

The attack accuracy over Keyword-100 achieves 71.0% in testing, which is not as high as the one from Alexa-100. We believe this is likely because some of the web pages in Keyword-100 are from the same websites, which derived similar traffic patterns. For instance, when we searched two different keywords, it is common to see that webpages from `youtube.com` or `wikipedia.com` are included in the top search results for both keywords.

Experiment A.2: Attack results on defended data over single traces. In this experiment, we investigated the attack accuracy of website fingerprinting on defended data over single traces. Specifically, one of the state-of-the-art defenses, Walkie-Talkie [14] was investigated. We implemented Walkie-Talkie by leveraging the source code released by the authors [14].

In Walkie-Talkie, the browser communicates in half-duplex mode instead of a full-duplex mode of communication. Further, burst molding is performed for two different web pages such that both the traffic pattern of the two web pages looks identical. As a result, Walkie-Talkie can reduce the attack accuracy in the closed-world setting to 50% in theory. The actual experimental results are normally close to 50% according to the results in previous studies. We generated the defended data for Alexa-100 and Keyword-100 using Walkie-Talkie and examined the attack accuracy over defended data using the DF model.

Results shown in Table III indicate that we implemented Walkie-Talkie properly. For example, the attack accuracy over defended data produced by Walkie-Talkie is 56% and 50.3% in testing for Alexa-100 and Keyword-100. These results are close to the theoretical result (i.e., 50% accuracy) offered by Walkie-Talkie.

TABLE III
ACCURACY ON DEFENDED DATASETS (SINGLE TRACES).

	Training	Validation	Testing
Alexa-100	66.1%	56.6%	56.0%
Keyword-100	54.2%	47.5%	50.3%

Experiment A.3: Attack results on defended data (generated by Walkie-Talkie) with cross-trace website fingerprinting. In this experiment, we demonstrated how our proposed cross-trace website fingerprinting can defeat Walkie-Talkie when there are semantic correlations across encrypted traffic traces. We evaluated the attack performance over the defended data of Keyword-100, where the defended data are generated with Walkie-Talkie. The DF model is still trained with training traces over single traces. During the test, the semantic similarity evaluation across candidate websites can adjust the score/confidence of labels, which improves the accuracy of test traces in the cross-trace scenario as explained in Sec. III.

For the test traces in Keyword-100, we randomly pair two traces as follows to simulate the case where a user visits two consecutive web pages with semantic correlation. Specifically, given a test trace f_1, we randomly select another test trace f_2, where f_1 and f_2 are associated with two different webpages but the two webpages are search results of the same keyword. We obtained an ordered pair (f_1, f_2) indicating trace f_1 happens right before trace f_2 on a user side. Each test trace is used only in one pair. We evaluate the performance of cross-trace website fingerprinting multiple times and randomly generate the pairs for each time.

As shown in Table IV, our proposed cross-trace website fingerprinting can defeat Walkie-Talkie. For example, when using TF-IDF, GLoVe, or WMD to measure the semantic correlation of candidate webpages inferred by the DF model, our method can achieve attack accuracy that is much higher than 50% over defended data generated by Walkie-Talkie. Particularly, when we use TF-IDF, our method can even achieve the same level of accuracy as the one over non-defended data, which indicates Walkie-Talkie fails to provide any privacy protection in the cross-trace evaluation.

In addition, we observe that different text similarity methods lead to different improvements in attack accuracy. For example, TF-IDF is the most effective one while BERT is not able to improve the attack accuracy over defended data. This is likely because BERT is not able to accurately measure the semantic correlation of different web pages in our evaluation.

Experiment A.4: The impact of the number of top predictions on single traces. In this experiment, we increase parameter m, the number of top predictions on single traces, and examine how it will affect the performance of cross-trace website fingerprinting. Specifically, rather than obtain the top-2 labels for each trace in the last experiment, we obtain top-3 or top-4 labels for each trace, and then compute the corresponding semantic correlation over these candidates across two traces in each pair formed from the last experiment. We focus on the results with TF-IDF and GLoVe.

As we can see from Table V, the attack accuracy of cross-trace website fingerprinting decreases when we increase the value of m. This is expected, when more candidate webpages are selected for each trace, webpages with similar content will likely be selected for a single trace, which leads to multiple similar semantic distances with candidate webpages of the other trace in the same ordered pair. These similar semantic distances make it difficult to distinguish the true labels/webpages of the two traces in the pair.

Experiment A.5: The impact of the number of semantically-correlated traces. We examine the impact of the number of semantically correlated traces in cross-trace website fingerprinting. Specifically, in previous examples, we assume that a user visits $\alpha = 2$ semantically-correlated webpages, which result in $\alpha = 2$ consecutive traffic traces.

In this experiment, we further consider the cases with $\alpha = \{2, 3, 4\}$, i.e., a user visits three, four, or five semantically correlated webpages consecutively. We evaluate with defended Keyword-100 dataset. The defended data are still generated by

TABLE IV
ACCURACY ON DEFENDED DATASETS (WALKIE-TALKIE [14]) USING OUR PROPOSED CROSS-TRACE WEBSITE FINGERPRINTING

	No Defense	With Defense				
		Without Our Method	TF-IDF + Cosine	GLoVe + Cosine	WMD	BERT + Cosine
Keyword-100	71.0%	50.4%	**72.5%**	**59.5%**	**63.0%**	51.5%

TABLE V
THE IMPACT OF THE NUMBER OF TOP CANDIDATES ON SINGLE TRACES WITH KEYWORD-100 DATASET

	TF-IDF + Cosine	GLoVe + Cosine
m = 2	72.50%	59.53%
m = 3	62.75%	46.92%
m = 4	53.76%	38.34%

Walkie-Talkie. We fix the number of top candidates per single trace as $m = 2$ in this experiment.

We use a similar approach as the case of $\alpha = 2$ in Experiment A.3 to select traces. For example, given $\alpha = 3$, we randomly choose traces f_1, f_2, and f_3 from test traces and form a 3-trace sequence (f_1, f_2, f_3), where the three traces belong to three different web pages but the three webpages are from the search results of the same keyword. Each trace is included in one 3-trace sequence. We assume that a user visits the three webpages sequentially and the three traces were captured by the attacker sequentially. We repeat the similar process for $\alpha = 4$.

TABLE VI
THE IMPACT OF THE NUMBER OF SEMANTICALLY-CORRELATED TRACES WITH THE KEYWORD-100 DATASET FOR $m = 2$.

	TF-IDF + Cosine	GLoVe + Cosine
α = 2	72.5%	59.5%
α = 3	73.2%	61.0%
α = 4	73.8%	61.3%

As we can observe from Table. VI, if a user visits more webpages that are correlated, then the attack accuracy can increase slightly. This indicates that cross-trace website fingerprinting can reveal more privacy when it monitors and evaluates more consecutive traces together from a user.

Experiment A.6: Attack results on defended data (generated by Adversarial Examples) with cross-trace website fingerprinting. In this experiment, we evaluate the performance of cross-trace website fingerprinting over defended data generated by the recent defenses [15], [16] based on adversarial examples. The main idea of those defenses is to modify the traffic pattern with minor changes such that it can force a known deep neural network (i.e., a WF classifier) to predict an incorrect class. Specifically, we leverage Mockingbird [15], which is one of these recent defenses, to generate defended data of the Keyword-100 dataset. We use TF-IDF and GLoVe as two semantic similarity evaluation methods and pair the test traces using the same way we did in Experiment A.3.

As shown in Table. VII, our proposed cross-trace website fingerprinting is not effective over the defended data generated by Mockingbird. This is because the attack accuracy over the defended data without our method is only 11.3%, which is very low and the true label is not even included in the top-m candidates ($m = 2$ in this experiment). As a result, the additional semantic similarity evaluation across those top-m candidates does not help to further identify the true label of a single trace. From the results we have in this experiment, we can conclude that, in general, the defenses (e.g., Mockingbird) based on adversarial examples are more robust than the defenses (e.g., Walkie-Talkie) based on super-sequences under cross-trace website fingerprinting.

On the other hand, we would like to point out that Walkie-Talkie can be implemented in practice while whether the defenses based on adversarial examples can be implemented to defend website fingerprinting in the real world remains unclear. More specifically, the current defenses based on adversarial examples need to assume the entire traffic trace is known in advance, which is a very strong assumption and difficult to overcome in practice (i.e., sending the current defended packet needs to the know the information of a future packet, which has not happened yet). Moreover, these defenses also assume that a defender knows the specific parameters and architectures of a deep neural network used by an attacker in website fingerprinting. This is another extremely strong assumption, as a real-world attack can always adapt and freely switch to different parameters and architectures. In other words, our proposed cross-trace website fingerprinting remains a real threat to the current practical defenses.

V. RELATED WORK

A. Website Fingerprinting Attacks

Website fingerprinting can be formulated as a supervised learning problem in machine learning. Initial studies [1]–[3] manually extract features from raw encrypted network traffic and leverage traditional machine learning algorithms to infer which website a user visits. Many recent studies [4]–[8] have shown that deep learning can achieve higher accuracy in website fingerprinting. In addition, with deep learning, an attack can automatically extract features. For instance, Sirinam et al [5] proposed Deep Fingerprinting, which is built upon Convolutional Neural Networks and achieves 98% accuracy in the closed-world setting.

A couple of recent studies [9], [10] leverage transfer learning to overcome the discrepancy between training data and test data, where training data and test data are collected with different network setups. The two studies also show that transfer learning can significantly reduce the amount of data needed for carrying out website fingerprinting. Studies in [26], [27] addressed website fingerprinting in the multi-tab scenario, where a user could open multiple websites at the

TABLE VII
ACCURACY ON DEFENDED DATASETS (MOCKINGBIRD [15]) USING OUR PROPOSED CROSS-TRACE WEBSITE FINGERPRINTING

	No Defense	With Defense		
		Without Our Method	TF-IDF + Cosine	GLoVe + Cosine
Keyword-100	71.0%	11.3%	11.1%	10.3%

same time. Juarez et. al. [28] investigated the base rate fallacy in the open-world evaluation of website fingerprinting. Wang [29] proposed to use precision to measure the performance in the open-world evaluation. Wang et. al. [30] examined voice command fingerprinting over encrypted traffic of smart speakers using deep learning to infer which voice command a user says to a smart speaker with high accuracy.

B. Defenses against Website Fingerprinting

Many defenses [11]–[18] have been proposed to protect user privacy against website fingerprinting. The general approach of the existing defenses is to somehow modify the pattern of encrypted traffic such that it is more difficult for an attacker to distinguish the traffic pattern of one website from others. Wang et al. [14] proposed a defense, named Walkie-Talkie, which combines traffic traces of two different websites into a super sequence. Gong and Wang [18] proposed two defenses, FRONT, and GLUE. FRONT introduces more random dummy packets to the front part of a traffic trace to hide more critical features against website fingerprinting. GLUE, on the other hand, inserts dummy packets between two traces, making it impossible for an adversary to determine the two traffic traces apart. Two recent studies [15], [16] produce adversarial examples of encrypted traffic to mitigate the attack accuracy against deep-learning-based attacks. Studies in [31], [32] investigated effective defenses against deep-learning-based stream fingerprinting over encrypted traffic in streaming services (e.g., YouTube).

VI. DISCUSSIONS AND LIMITATIONS

In this study, we focus on a closed-world setting but do not carry out evaluations in an open-world setting. The reason is that it is difficult to integrate semantic similarity evaluation in the current open-world evaluation. More specifically, to perform semantic similarity, we need to assume that the content/text of a webpage is known. This is reasonable for an attacker to obtain in the closed-world setting as an attacker can easily retrieve the text by visiting each webpage. However, in the open-world setting, a user can visit some webpages that an attacker does not know. Without knowing which webpages (more specifically, the content of those webpages), it is infeasible to measure semantic similarity. It would be interesting to see how to perform cross-trace website fingerprinting in the open-world setting. We leave it as future work.

VII. CONCLUSION

We propose a cross-trace website fingerprinting by measuring the semantic correlations among multiple traffic traces. Our results over large-scale real-world datasets show that our proposed attack can completely defeat current defenses based on super-sequences. On the other hand, our results suggest that the latest defenses based on adversarial examples are robust against our cross-trace website fingerprinting.

ACKNOWLEDGMENT

The authors also would like to thank Dr. Marc Juarez and Vera Rimmer for explaining details regrading Tor traffic collection and thank Ohio Cyber Range at UC for providing multiple virtual machines to facilitate Tor traffic collection. The authors were partially supported by National Science Foundation (CNS-1947913) and UC Office of the Vice President for Research - Pilot Program.

APPENDIX

Keywords. The list of keywords used in Keyword-100 Dataset can be found in Table VIII.

TABLE VIII
KEYWORDS USED IN KEYWORD-100 DATASET

21 Savage	Copa America
James Charles	Nipsey Hussle
Baby Shark	Disney Plus
Jordyn Woods	Noom Diet
Bryce Harper	Felicity Huffman
Keanu Reeves	Notre Dame Cathedral
Cameron Boyce	Game of Thrones
Lori Loughlin	R Kelly
Camp Style	Government Shutdown
Luke Perry	Rami Malek

REFERENCES

[1] T. Wang, X. Cui, R. Nithyannand, R. Johnson, and I. Goldberg, "Effective Attacks on Provable Denfenses for Website Fingerprinting," in *Proc. of 23rd USENIX Security Symposium*, 2014.

[2] A. Panchenko, F. Lanze, A. Zinnen, M. Henze, J. Penekamp, K. Wehrle, and T. Engel, "Website Fingerprinting at Internet Scale," in *Proc. of NDSS'16*, 2016.

[3] J. Hayes and G. Danezis, "K-Fingerprinting: A Robust Scalable Website Fingerprinting Technique," in *Proc. of USENIX Security'16*, 2016.

[4] K. Abe and S. Goto, "Fingerprinting Attack on Tor Anonymity Using Deep Learning," in *Proc. of Aisa Pacific Advanced Network (APAN)*, 2016.

[5] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep Fingerprinting: Understanding Website Fingerprinting Defenses with Deep Learning," in *Proc. of ACM CCS'18*, 2018.

[6] V. Rimmer, D. Preuveneers, M. Juarez, T. V. Goethem, and W. Joosen, "Automated Website Fingerprinting through Deep Learning," in *Proc. of NDSS'18*, 2018.

[7] S. Bhat, D. Lu, A. Kwon, and S. Devadas, "Var-CNN: A Data-Efficient Website Fingerprinting Attack Based on Deep Learning," in *Proc. of PETS'19*, 2019.

[8] S. E. Oh, S. Sunkam, and N. Hopper, "p-FP: Extraction, Classification, and Predication of Website Fingerprints," in *Proc. of PETS'19*, 2019.

[9] P. Sirinam, N. Mathews, M. S. Rahman, and M. Wright, "Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1131–1148.

[10] C. Wang, J. Dani, X. Li, X. Jia, and B. Wang, "Adaptive Fingerprinting: Website Fingerprinting over Few Encrypted Traffic," in *Proc. of ACM CODASPY'21*, 2021.
[11] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail," in *Proc. of IEEE S&P'12*, 2012.
[12] X. Cai, R. Nithyanand, and R. Johnson, "CS-BuFLO: A Congestion Sensitive Website Fingerprinting Defense," in *Proc. of 13th ACM Workshop on Privacy in Electronic Society*, 2014.
[13] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, "Toward an Efficient Website Fingerprinting Defense," in *Proc. of ESORICS'16*, 2016.
[14] T. Wang and I. Goldberg, "Walkie-Talkie: An Efficient Defense Against Passive Website Fingerprinting Attacks," in *Proc. of USENIX Security'17*, 2017.
[15] M. S. Rahman, M. Imani, N. Mathews, and M. Wright, "Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces," *IEEE Transactions on Information Forensics and Security*, 2021.
[16] M. Nasr, A. Bahramali, and A. Houmansadr, "Blind Adversarial Network Perturbations," in *Proc. of USENIX Security'21*, 2021.
[17] S. Henri, G. Garcia-Aviles, P. Serrano, A. Banchs, and P. Thiran, "Protecting against Website Fingerprinting with Multihoming," in *Proc. of PETS'20*, 2020.
[18] J. Gong and T. Wang, "Zero-delay Lightweight Defenses against Website Fingerprinting," in *Proc. of USENIX Security'20*, 2020.
[19] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing Management*, vol. 24, no. 5, pp. 513–523, 1988.
[20] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger, "From Word Embeddings To Document Distances," in *Proc. of 32nd International Conference on Machine Learning*, 2015.
[21] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training for deep bidirectional transformers for language understanding," https://arxiv.org/abs/1810.04805.
[22] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in *Proc. of 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.
[23] "Cross-trace website fingerprinting." [Online]. Available: https://github.com/SmartHomePrivacyProject/CrossTraceWF
[24] "tor-browser-selenium - tor browser automation with selenium." [Online]. Available: https://github.com/webfp/tor-browser-selenium
[25] Microsoft, "NNI: An open source AutoML toolkit for neural architecture search and hyper-parameter tuning," 2017. [Online]. Available: https://github.com/Microsoft/nni
[26] Y. Xu, T. Wang, Q. Li, Q. Gong, Y. Chen, and Y. Jiang, "A Multi-Tab Website Fingerprinting Attack," in *Proc. of ACSAC'18*, 2018.
[27] W. Cui, T. Chen, C. Fields, J. Chen, A. Sierra, and E. Chan-Tin, "Revisting Assumtions for Website Fingerprinting Attacks," in *Proc. of ACM ASIACCS'19*, 2019.
[28] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A Criticial Evaluation of Website Fingerprinting Attacks," in *Proc. of ACM CCS'14*, 2014.
[29] T. Wang, "High Precision Open-World Website Fingerprinting," in *Proc. of IEEE S&P'20*, 2020.
[30] C. Wang, S. Kennedy, H. Li, K. Hudson, G. Atluri, X. Wei, W. Sun, and B. Wang, "Fingerprinting Encrypted Voice Traffic on Smart Speakers with Deep Learning," in *Proc. of ACM WiSec'20*, 2020.
[31] X. Zhang, J. Hamm, M. K. Reiter, and Y. Zhang, "Statistical Privacy for Streaming Traffic," in *Proc. of NDSS'19*, 2019.
[32] H. Li, B. Niu, and B. Wang, "SmartSwitch: Efficient Traffic Obfuscation against Stream Fingerprinting," in *Proc. of SecureComm 2020*, 2020.